



Managing Multiple Versions of R in a Clinical Environment

twitter:@rinpharma github:philbowsher



It is important that R users understand the package ecosystem and that IT provides them an internal foundation to build on

Core Concepts

- Don't upgrade R - Install multiple versions
- A library is tied to a specific version of R
- Different versions of R will have different system libraries
- IT should create an internal server that houses the R packages
- Users will install packages from the frozen repositories states for the specific version of R
- Users will not install packages from the external CRAN
- The Production environment will house the artifacts

Step 1: Create a R Version & Package Plan

- In the open source ecosystem there is a combination of products (RStudio), core software (R), and add-ons (packages)
- It is a best practice not to upgrade R but rather to install multiple versions on the same server
- RStudio provides pre-compiled R binaries to help with the installation of different versions of R side by side in Linux environments (servers)



Files to customize R's behavior

- [Renviron.site](#)
- Allows users to specify environment variables that should be available to any R session
- [Rprofile.site](#)
- The site-wide R profile allows you to supply R code that will be executed prior to the R session starting for any user accessing a version of R



Environments

- Not all clinical environments are the same
- Stage, test, prod, data science labs, cloud images and desktop users environments
- The Environment drives the plan in step 1
- This Presentation is focused on Production



Step 2: Pick versions of R

- In both validated and research environments, it is important for IT to align on specific versions of R that they want to support and maintain
- It is recommended not to upgrade R, but rather maintain multiple versions of R side by side
- RStudio recommends organizations roll out new versions of R (and new package sets) every 4-6 months for most environments except for GxP, where new R versions are often only added every 1-2 years

The background of the slide is a photograph of a street scene at sunset. On the left, there are utility poles with many wires. A car with its headlights on is visible in the distance on the right. The sky is a mix of orange and white. The text is overlaid on the right side of the image.

If your old code relies on older packages, simply continue to use the old code with the older version of R. This is the beauty of having multiple R versions. If you want to bring old code up to date with a new version of R, expect that process to include upgrading packages and fixing your code



Step 3: Create an Internal CRAN and/or a curated repository

- Have a dedicated team or R admin that will create an internal repository of packages
- `options('repos')`
- If your old code relies on older packages, simply continue to use the old code with the older version of R



Don't install packages from external CRAN

- Usually users understand what R packages are, but lack an understanding of R versions, repositories, mirrors and package versions
- The goal of this presentation is to retrain users to install packages by using the IT specified internal repository

system libraries, user libraries, and project libraries (renv)

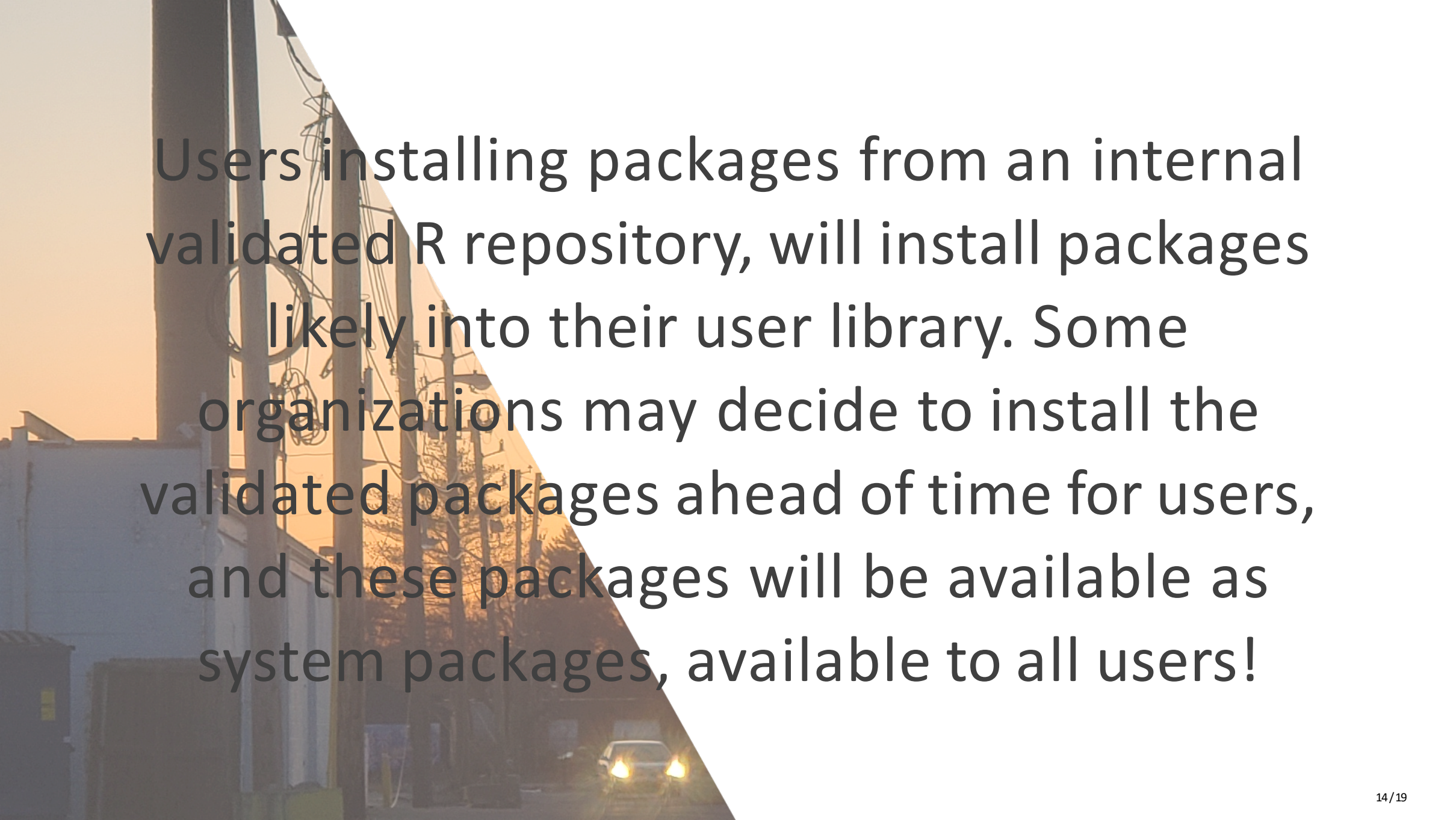
- R packages are installed into libraries, which are directories in the file system containing a subdirectory for each package installed there
- `.libPaths()`
- 1 `"/usr/home/phil/R/x86_64-pc-linux-gnu-library/3.6"`
- 2 `"/opt/R/3.6.2/lib/R/library"`



A library is tied to a specific version of R; and
different versions of R will have different
system libraries and require different user
libraries and package installations

Step 4: Freeze Repository & tie it to the R version

- How do you ensure the correct packages are installed in that library? This is where the admin management of repositories comes into play
- A. Allow users the freedom to install packages from a Frozen repository... OR
- B. Admins install packages via a Frozen Repository ahead of time for users

The background of the slide is a photograph of a street scene at sunset. On the left, there are several utility poles with a dense network of power lines. A car with its headlights on is visible in the distance on the road. The sky is a warm orange color from the setting sun. The text is overlaid on the right side of the image.

Users installing packages from an internal validated R repository, will install packages likely into their user library. Some organizations may decide to install the validated packages ahead of time for users, and these packages will be available as system packages, available to all users!

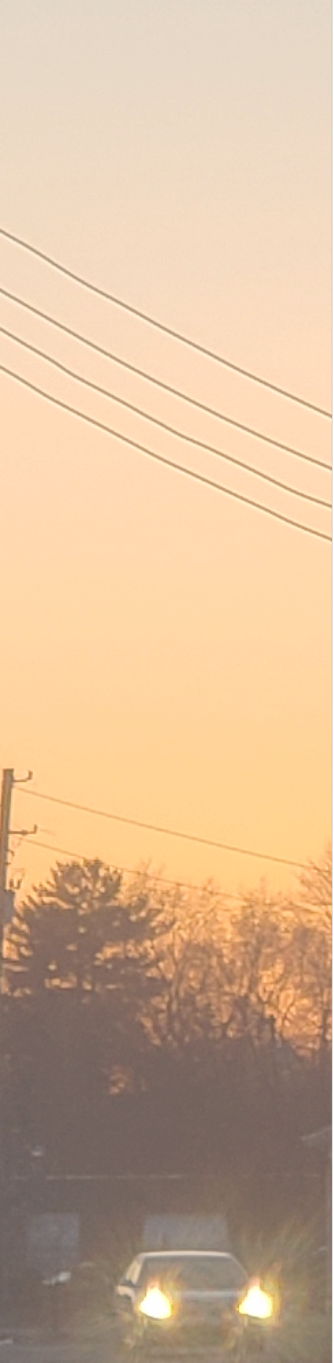
Docker

- Requires knowledge of Docker and other tools
- But the same lessons apply via...
- Include package installation in the Dockerfile which embeds the packages into the image
- add appropriate R packages when the container is run
- RUN R -e 'install.packages(..., repo = ["https://rpkgs.company.com/frozen/repo/123"](https://rpkgs.company.com/frozen/repo/123))'



Desktop Users

- It is a best practice to have users on a dedicated server
- Advice above still stands
- Standardize on a version of R, and have a repository of validated/clinical packages and have users only use those packages
- Often the challenge is the user knowing how to change the default location for installing packages so user training may be required or do it for them via the `Renviron.site`



Step 5: Production for Clinical Artifacts

- It is recommended that statistical programmers write the code; then harden the result
- Once code has completed development, it can be hardened and placed in a persistent environment
- If you want to have a qualified validated result, than your final code should not be in a "development" environment
- A production environment ensures that the code, results, and dependencies are isolated, easily accessible, and safely locked away

Validated Recap

- Step 1: Create a R Version and Package Plan
- Step 2: Install R versions (separately) that are supported
- (Don't upgrade R but rather add new versions of R)
- Step 3: Create an Internal CRAN and/or a curated repository
- Step 4: Freeze Repository and align to your specific version(s) of R to that internal repo
- Step 5: Production for Clinical Artifacts



rstudio conference & rinpharma