

R Packages

Justin Post

What do we want to be able to do?

Data Science!

- Read in raw data and manipulate it
- Combine data sources
- Summarize data to glean insights
- Apply common analysis methods
- Communicate Effectively

Where Do Our Objects & Functions Come From?

When you open R a few packages are loaded

- R package:
 - Collection of functions/objects/datasets/etc.

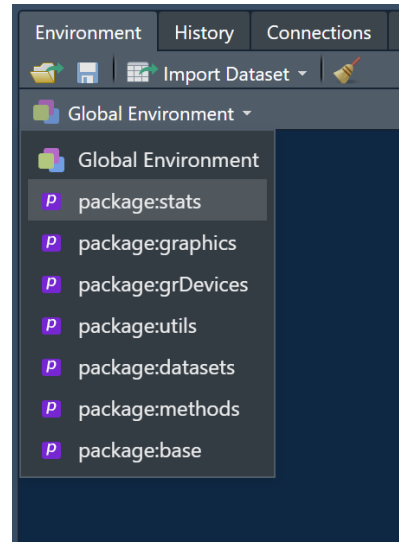
Where Do Our Objects & Functions Come From?

When you open R a few packages are loaded

- R package:
 - Collection of functions/objects/datasets/etc.
- Packages exist to do almost anything
 - **List of CRAN** approved packages
 - Plenty of other packages on places like GitHub!

Where Do Our Objects & Functions Come From?

- When you open R a few packages are loaded

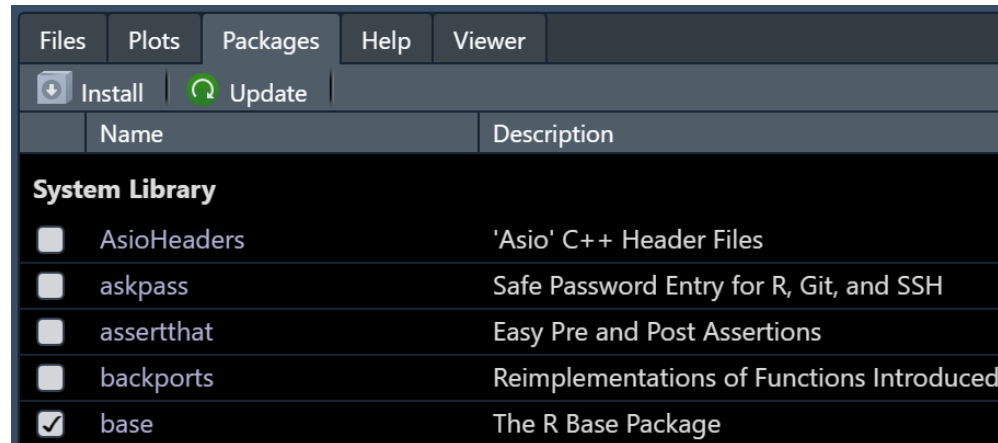


- base package has `c()`, `data.frame()`, `list()`, ...

Installing an R Package

- First time using a package
 - Must **install package**
 - Can use code, menus, or Packages tab

```
install.packages("dplyr")
```



Accessing a Package in Your R Session

- Only install once!
- **Each session**: read in package using `library()` or `require()`

```
library("dplyr")
```

Accessing a Package in Your R Session

- Difference - if no package
 - `library()` throws an error
 - `require()` returns FALSE

```
library("notAPackage")
```

```
## Error in library("notAPackage"): there is no package called 'notAPackage'
```

```
require("notAPackage")
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,  
## logical.return = TRUE, : there is no package called 'notAPackage'
```


Setting Packages to Automatically Load

To load packages by default, access your `.Rprofile` file

- Add in the code

```
options(defaultPackages=c(getOption("defaultPackages"),  
                           "mypackage1",  
                           "mypackage2"))
```

See [here](#) for info on finding your `.Rprofile` file

Accessing a Package in Your R Session

- See everything from a package:

```
ls("package:dplyr")
```

```
## [1] "%>%" "across" "add_count"
## [4] "add_count_" "add_row" "add_rownames"
## [7] "add_tally" "add_tally_" "all_equal"
## [10] "all_of" "all_vars" "anti_join"
## [13] "any_of" "any_vars" "arrange"
## [16] "arrange_" "arrange_all" "arrange_at"
## [19] "arrange_if" "as.tbl" "as_data_frame"
## [22] "as_label" "as_tibble" "auto_copy"
## [25] "band_instruments" "band_instruments2" "band_members"
## [28] "bench_tbls" "between" "bind_cols"
## [31] "bind_rows" "c_across" "case_match"
## [34] "case_when" "changes" "check_dbplyr"
## [37] "coalesce" "collapse" "collect"
## [40] "combine" "common_by" "compare_tbls"
## [43] "compare_tbls2" "compute" "consecutive_id"
## [46] "contains" "copy_to" "count"
## [49] "count_" "cross_join"
```

Calling From a Library

- Call functions without loading full library with `::`
- If not specified, most recently loaded package takes precedent

```
#stats::filter(...) calls time-series function from stats package  
dplyr::filter(iris, Species == "virginica")
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	6.3	3.3	6.0	2.5	virginica
## 2	5.8	2.7	5.1	1.9	virginica
## 3	7.1	3.0	5.9	2.1	virginica
## 4	6.3	2.9	5.6	1.8	virginica
## 5	6.5	3.0	5.8	2.2	virginica
## 6	7.6	3.0	6.6	2.1	virginica
## 7	4.9	2.5	4.5	1.7	virginica
## 8	7.3	2.9	6.3	1.8	virginica
## 9	6.7	2.5	5.8	1.8	virginica
## 10	7.2	3.6	6.1	2.5	virginica
## 11	6.5	3.2	5.1	2.0	virginica
## 12	6.4	2.7	5.3	1.9	virginica
## 13	6.8	3.0	5.5	2.1	virginica
## 14	5.7	2.5	5.0	2.0	virginica
## 15	5.8	2.8	5.1	2.4	virginica
## 16	6.7	3.1	5.3	2.3	virginica
## 17	6.9	3.1	5.5	1.8	virginica
## 18	6.7	3.1	6.7	2.2	virginica
## 19	7.7	2.6	6.9	2.3	virginica

Quick R Example

Let's jump into R and see how to download a package that will allow us to create a .pdf file from our .qmd!

R Packages Recap

- Install packages first (download it)
 - Can do more than one at a time
- Load package with `require()` or `library()`
- Call a function without loading using `::`