WEEK1

R BASICS

TOPICS

- Some R basics
- Dealing with R Packages

BASIC R: IMPORTANT CONCEPTS

- "Running code" involves giving commands to R that are executed in the Console pane.

- You *can* type each line of code in the Console, but it is a much better idea to write a *draft of your code as you go* by using an R script.

- A script is just a text file that you save R code in.

- Use Cmd + Enter (Mac) or Ctrl + Enter (Windows) to run selected lines from your R script.

BASIC R: IMPORTANT CONCEPTS

- Values are saved in *objects*, which have names.
  - Names are case-sensitive!
  - When creating objects, aim for short, meaningful names.
    - Style guides recommend names in all lowercase, with words separated by underscores (e.g., `my_name`).
    - Valid names in R cannot start with a number or include the following special symbols: `^ ! $ @ + - / *`
- Create new objects using the assignment operator (`<-` which will appear as ← hereafter)

BASIC R: IMPORTANT CONCEPTS

- You do things using operators and functions.
  - Example operators: `+`, `←`
  - Example functions: `c()`, `mean()`, `summary()`

BASIC R: OPERATORS

ASSIGNMENT OPERATORS

OPERATOR	DESCRIPTION
←	ASSIGN A VALUE TO A NAME
=	SAME BUT MORE LIMITED (DON'T USE FOR NOW)

```
my_name ← "Fernando Campos"
my_height ← 182
```

Tip: use RStudio shortcut (`⌥` + `-` in macOS, `Alt` + `-` in Windows)

BASIC R: OPERATORS

ARITHMETIC OPERATORS

OPERATOR	DESCRIPTION
+	ADDITION
-	SUBTRACTION
*	MULTIPLICATION
/	DIVISION
^	EXPONENTIATION

```
2 + 4
2 - 4
2 * 4
2 / 4
2 ^ 4
```

BASIC R: OPERATORS

COMPARISON OPERATORS

OPERATOR	DESCRIPTION
<	LESS THAN
<=	LESS THAN OR EQUAL TO
>	GREATER THAN
>=	GREATER THAN OR EQUAL TO
==	EXACTLY EQUAL TO
!=	NOT EQUAL TO

BASIC R: OPERATORS

COMPARISON OPERATORS

OPERATOR	DESCRIPTION
<	LESS THAN
⩽	LESS THAN OR EQUAL TO
>	GREATER THAN
⩾	GREATER THAN OR EQUAL TO
==	EXACTLY EQUAL TO
≠	NOT EQUAL TO

```
# What kind of values will be returned?
2 < 4
2 ⩾ 4
2 == 4
2 ≠ 4
```

BASIC R: OPERATORS

LOGICAL OPERATORS

OPERATOR	DESCRIPTION
!	NOT
|	OR
&	AND

```
!TRUE
TRUE | FALSE
TRUE & FALSE
(1 < 5) & (7 > 1)

my_name ← "Fernando Campos"
my_height ← 182

(my_height > 190) | (my_name == "Fernando Campos")
```

BASIC R: IMPORTANT CONCEPTS

- R comes with many built-in functions, and packages provide additional specialized functions.
- Data that you pass into a function is called the function's argument.
- Some arguments are optional, some are required. Optional arguments have a default value if none is specified.
- Arguments have names, which you *can* (but usually should not) omit.
- Nested functions are evaluated inside to outside.

```
round(3.1415)
round(x = 3.1415, digits = 2)
round(mean(1:6), digits = 0)
```

BASIC R: IMPORTANT CONCEPTS

- R uses *element-wise execution*
    - An operation on an object with multiple element is applied to each element in the set.
        - When both objects have multiple elements, they elements are matched up by position and the operations are applied separately to each pair.

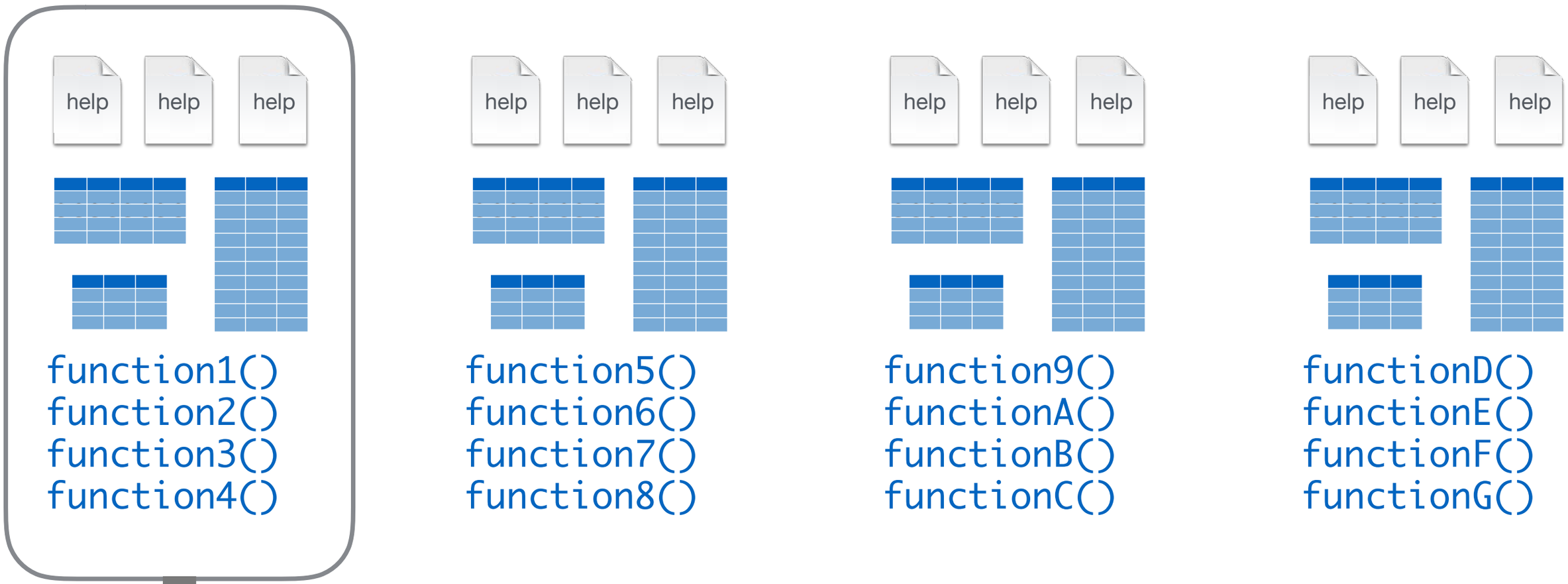
```
dice ← 1:6
dice * 10
dice * dice
```

BASIC R: IMPORTANT CONCEPTS

- Comments (lines that R does not try to evaluate) start with #
    - Good code includes lots of comments!
- Computers are stupid: R can't guess what you want to do. Attention to detail is important!

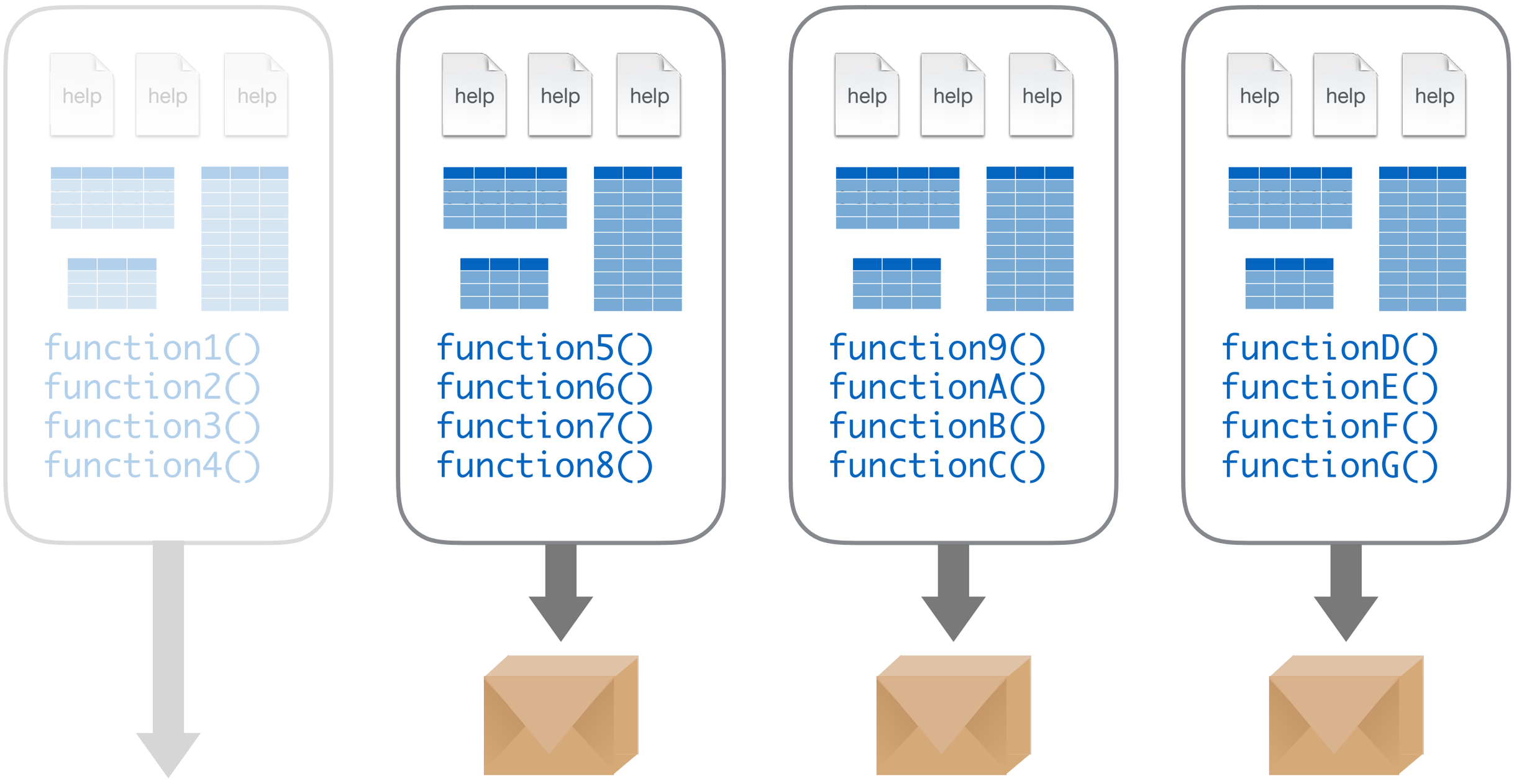
R PACKAGES

R PACKAGES

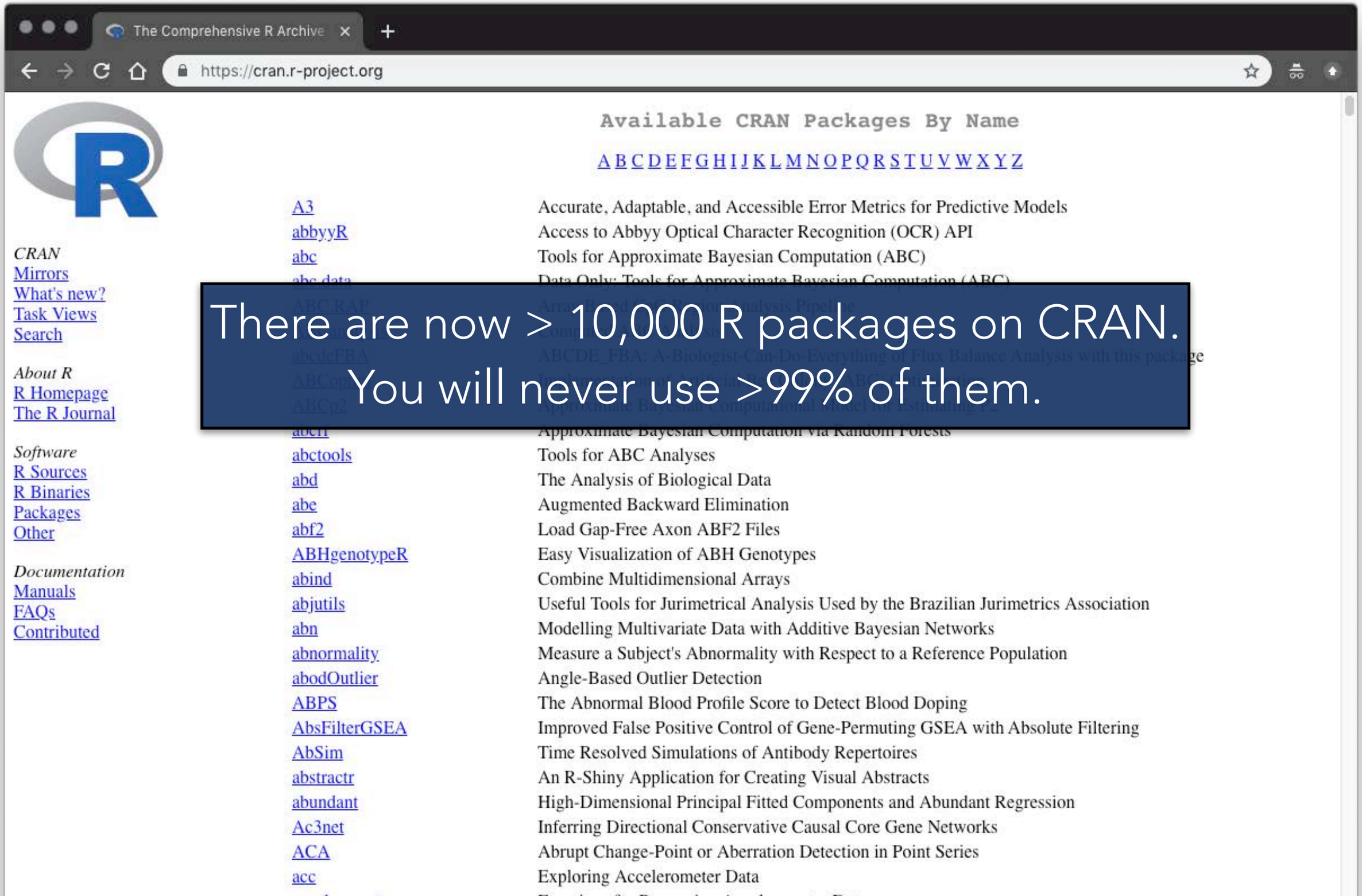


Base R

R PACKAGES



R PACKAGES



There are now > 10,000 R packages on CRAN.
You will never use >99% of them.

USING PACKAGES

1

```
install.packages("foo")
```

Downloads files to computer

1 x per computer

2

```
library("foo")
```

Loads package

1 x per R Session