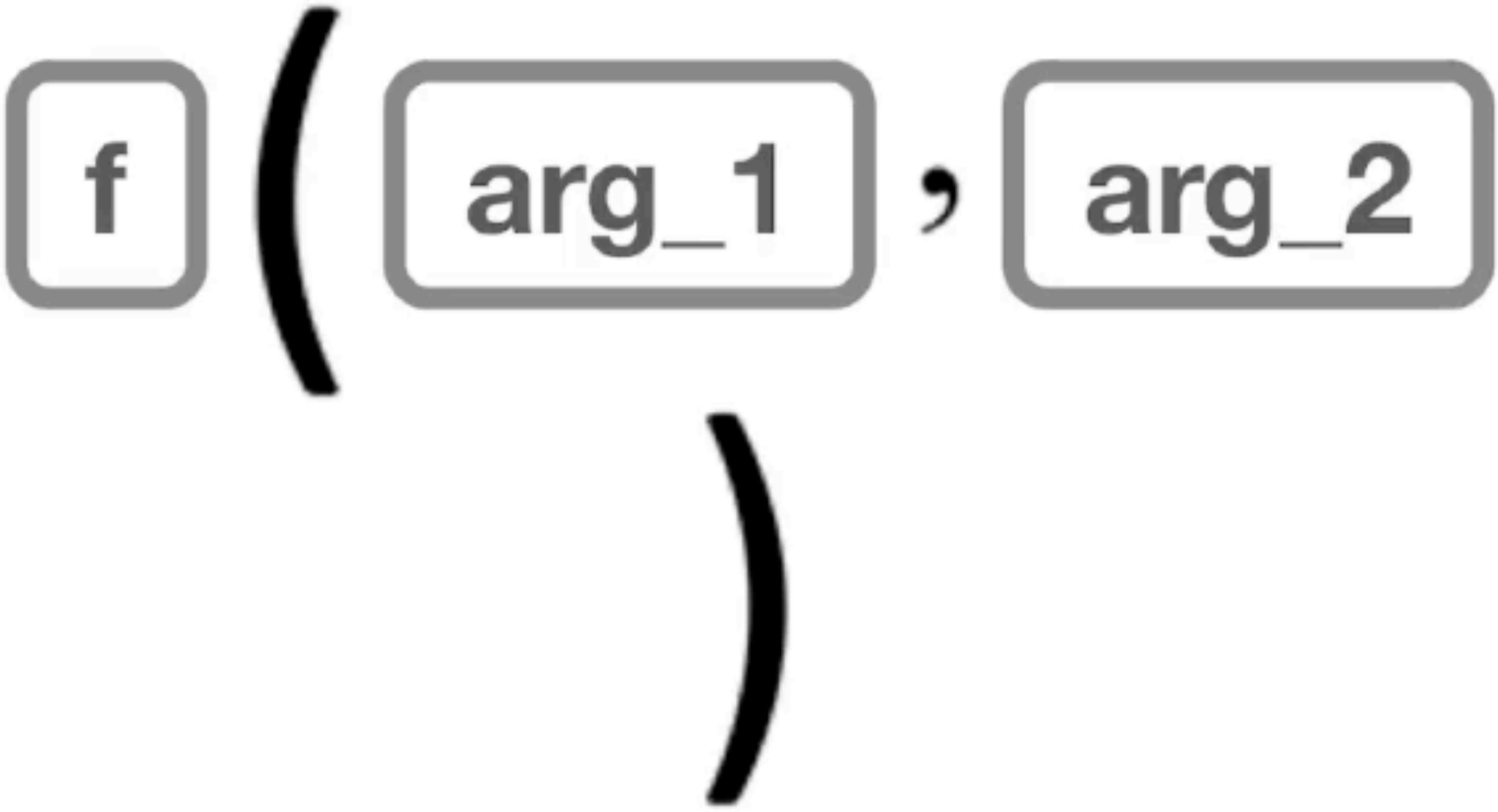
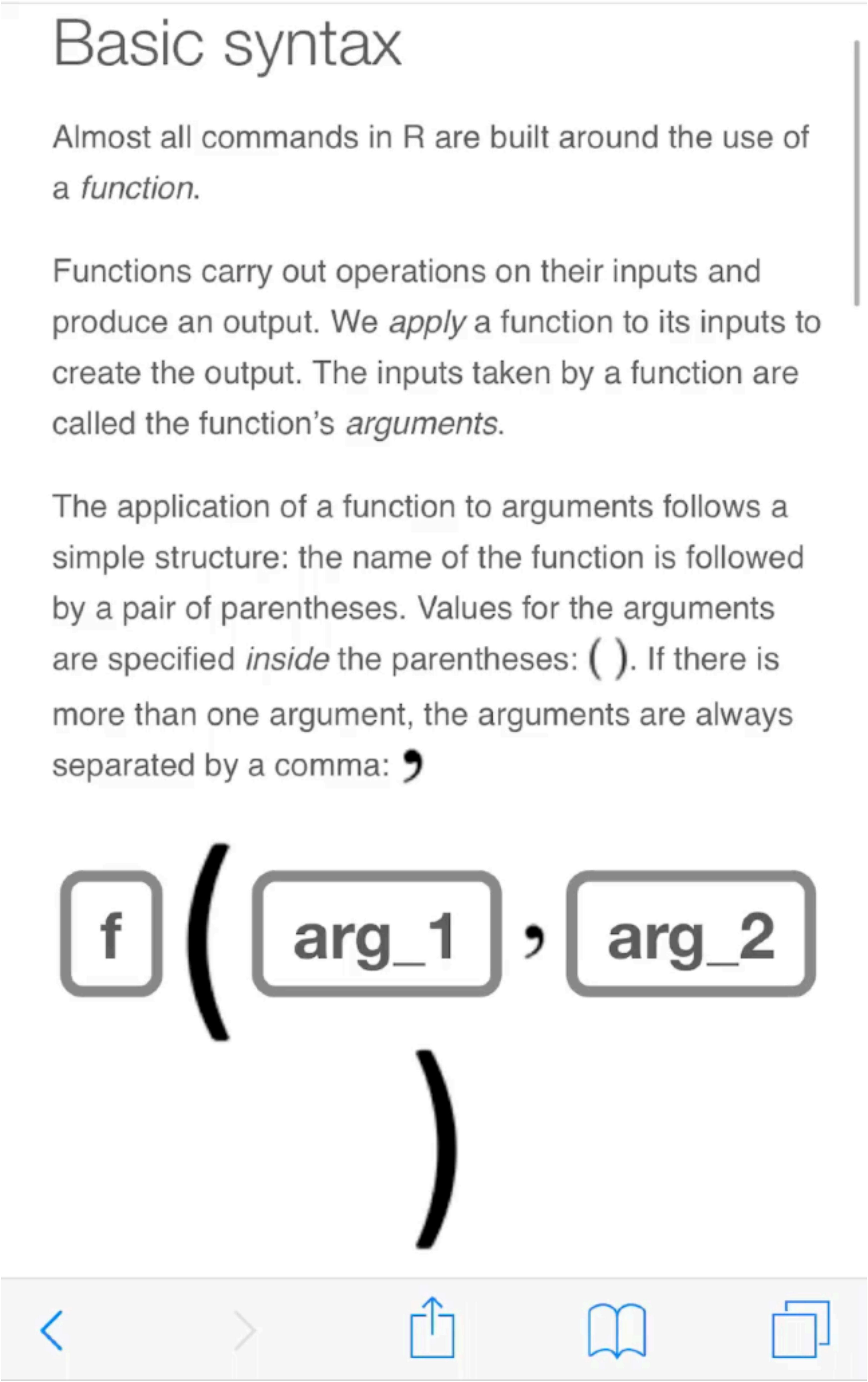
Basic syntax

Almost all commands in R are built around the use of a *function*.

Functions carry out operations on their inputs and produce an output. We *apply* a function to its inputs to create the output. The inputs taken by a function are called the function's *arguments*.

The application of a function to arguments follows a simple structure: the name of the function is followed by a pair of parentheses. Values for the arguments are specified *inside* the parentheses: ( ). If there is more than one argument, the arguments are always separated by a comma: ,





Basic syntax

Almost all commands in R are built around the use of a *function*.

Functions carry out operations on their inputs and produce an output. We *apply* a function to its inputs to create the output. The inputs taken by a function are called the function's *arguments*.

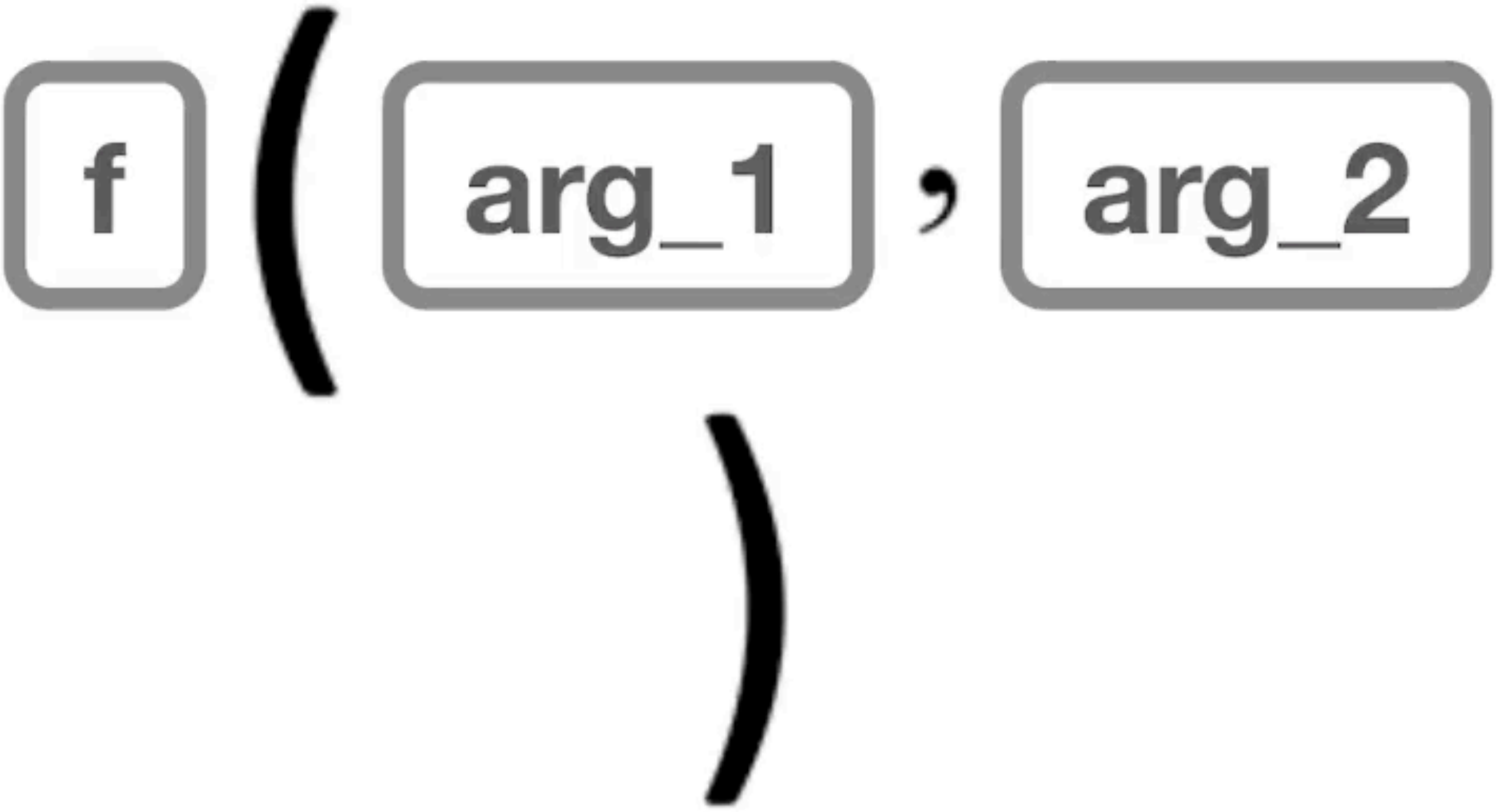
The application of a function to arguments follows a simple structure: the name of the function is followed by a pair of parentheses. Values for the arguments are specified *inside* the parentheses: ( ). If there is more than one argument, the arguments are always separated by a comma: **,**

Basic syntax

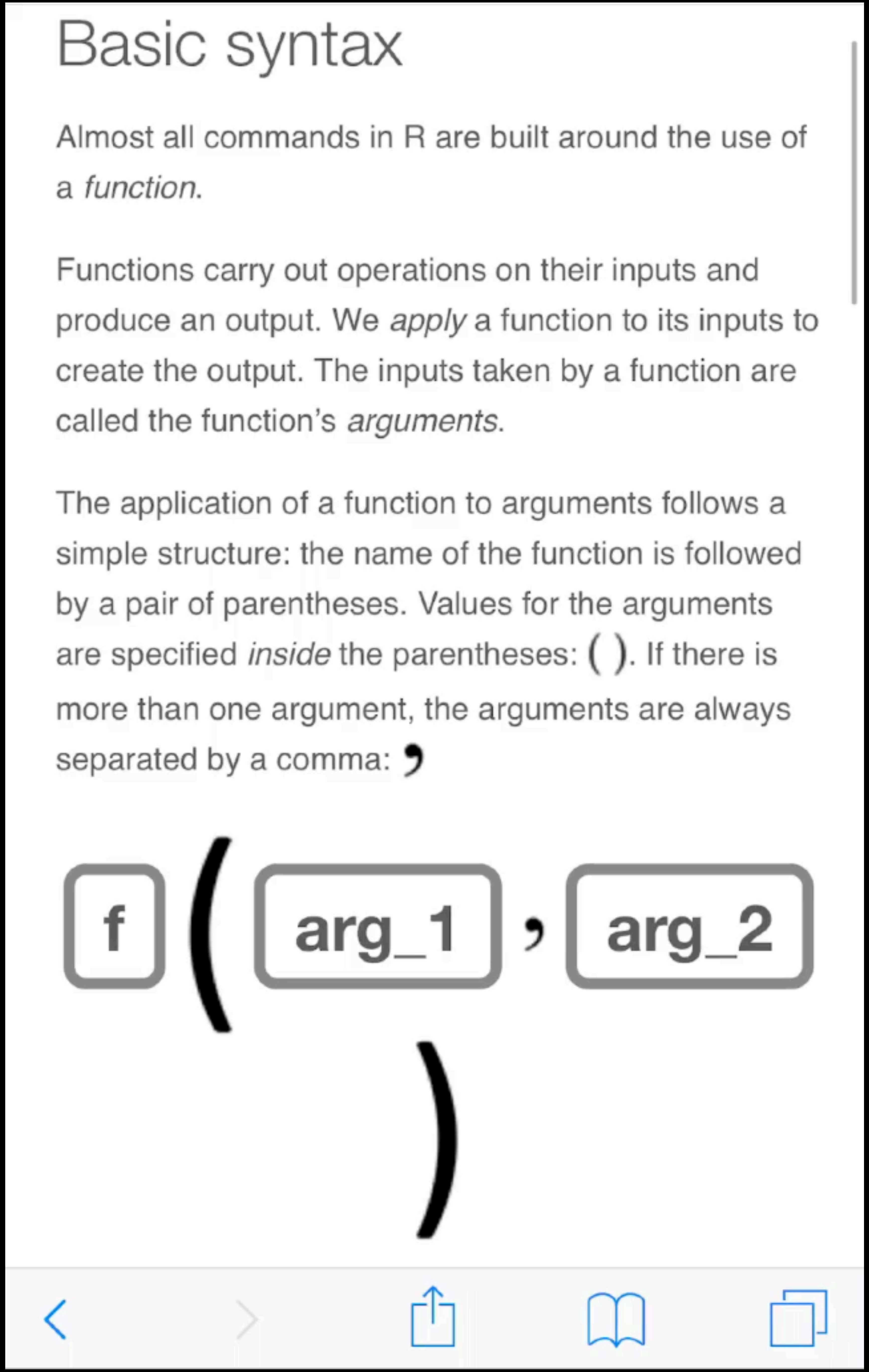
Almost all commands in R are built around the use of a *function*.

Functions carry out operations on their inputs and produce an output. We *apply* a function to its inputs to create the output. The inputs taken by a function are called the function's *arguments*.

The application of a function to arguments follows a simple structure: the name of the function is followed by a pair of parentheses. Values for the arguments are specified *inside* the parentheses: ( ). If there is more than one argument, the arguments are always separated by a comma: ,



https://dtkaplan.shinyapps.io/Functions/#section-basic-syntax



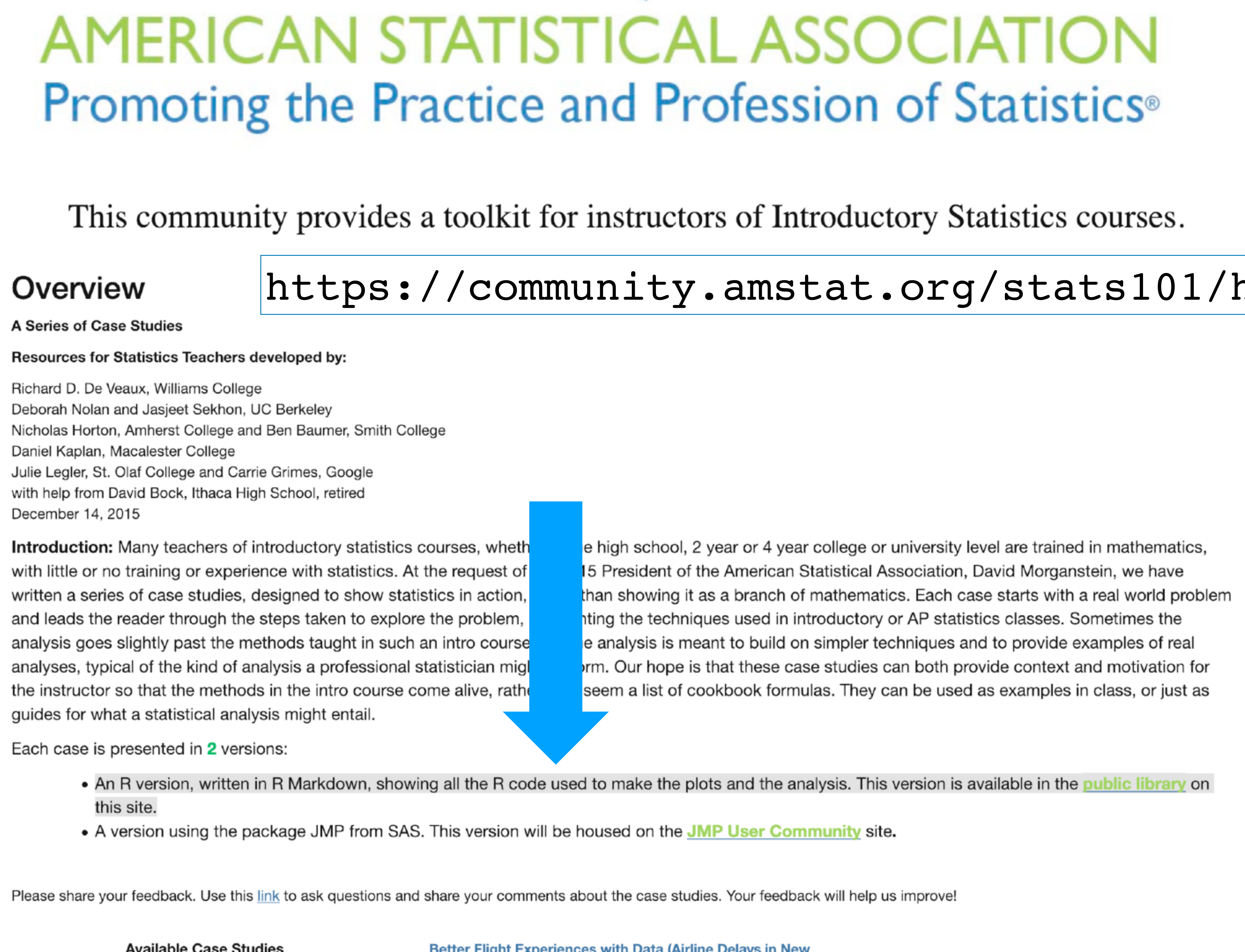
Basic syntax

Almost all commands in R are built around the use of a *function*.

Functions carry out operations on their inputs and produce an output. We *apply* a function to its inputs to create the output. The inputs taken by a function are called the function's *arguments*.

The application of a function to arguments follows a simple structure: the name of the function is followed by a pair of parentheses. Values for the arguments are specified *inside* the parentheses: ( ). If there is more than one argument, the arguments are always separated by a comma: ,

statprep.org



AMERICAN STATISTICAL ASSOCIATION

Promoting the Practice and Profession of Statistics®

This community provides a toolkit for instructors of Introductory Statistics courses.

https://community.amstat.org/stats101/home

Overview

A Series of Case Studies

Resources for Statistics Teachers developed by:

Richard D. De Veaux, Williams College
Deborah Nolan and Jasjeet Sekhon, UC Berkeley
Nicholas Horton, Amherst College and Ben Baumer, Smith College
Daniel Kaplan, Macalester College
Julie Legler, St. Olaf College and Carrie Grimes, Google
with help from David Bock, Ithaca High School, retired
December 14, 2015

Introduction: Many teachers of introductory statistics courses, wheth e high school, 2 year or 4 year college or university level are trained in mathematics, with little or no training or experience with statistics. At the request of 15 President of the American Statistical Association, David Morganstein, we have written a series of case studies, designed to show statistics in action, than showing it as a branch of mathematics. Each case starts with a real world problem and leads the reader through the steps taken to explore the problem, nting the techniques used in introductory or AP statistics classes. Sometimes the analysis goes slightly past the methods taught in such an intro course e analysis is meant to build on simpler techniques and to provide examples of real analyses, typical of the kind of analysis a professional statistician migl rm. Our hope is that these case studies can both provide context and motivation for the instructor so that the methods in the intro course come alive, rathe seem a list of cookbook formulas. They can be used as examples in class, or just as guides for what a statistical analysis might entail.

Each case is presented in **2** versions:

- An R version, written in R Markdown, showing all the R code used to make the plots and the analysis. This version is available in the public library on this site.
- A version using the package JMP from SAS. This version will be housed on the **JMP User Community** site.

Please share your feedback. Use this link to ask questions and share your comments about the case studies. Your feedback will help us improve!

Available Case Studies

Better Flight Experiences with Data (Airline Delays in New