## R functions and packages

* The functions **warning(**) and **stop()**
  + The functions warning() and stop() are used inside functions to handle and report on unexpected situations
  + They both print a user defined message (which you supply as a character input argument to the warning() and stop() functions)
  + However, warning() will keep on going with running the function body code whereas stop() will terminate the action of the function
  + A common idiom is to use stop(“some message”) to report on unexpected input type or other problem early in a function,
    - i.e., “Fail early and loudly!”
* suggested steps for writing your functions
  + start with a simple problem and get a working snippet of code
  + rewrite to use temporary variables (e.g. x, y, df, m, etc.)
  + turn into an initial function with clear useful names
  + test on small well defined input and (subsets of) real input
    - test on various inputs (a.k.a. **eejit proofing**)
  + report on potential problem by failing early and loudly
  + refine and polish
  + document and comment within the code on your reasoning
* what makes a good function?
  + Correct
  + Understandable
  + Correct + understandable = **obviously correct**
  + Use **sensible names** throughout.
    - Good names make code understandable with minimal context. You should strive for self-explanatory names

## CRAN & Bioconductor (major repositories for **R packages** that extend R functionality)

* **CRAN**: comprehensive R archive network
  + CRAN is a network of mirrored servers around the world that administer and distribute R itself, R documentation and **R packages** (basically add on functionality!)
    - R packages can be of variable quality and often there are multiple packages with overlapping functionality
  + There are currently ~14,038 packages on CRAN in the area of finance, bioinformatics, machine learning, high performance computing, multivariate statistics, natural language processing, etc. etc.
  + <https://cran.r-project.org/>
  + Installing a package
    - Way 1: RStudio > Tools > Install Packages
    - Way 2:
      * > install.packages(“bio3d”)
      * > library(“bio3d”)
* **Bioconductor**
  + R packages and utilities for working with high-throughput genomic data
  + <http://bioconductor.org>
  + More pragmatic:
    - Bioconductor is a **software repository** of **R packages** **with some rules and guiding principles**
    - Version 3.8 had 1649 software packages
  + Bioconductor has emphasized **reproducible research** since its start, and has been an early adapter and driver of tools to do this
  + Installing a Bioconductor package
    - First get the BiocManager package from CRAN
      * > install.packages(“BiocManager”)
      * > library(BiocManager)
    - Install Bioconductor core packages with:
      * > install(“AnnotationDbi”)
    - See: <http://www.bioconductor.org/install/>
* Key Idea:
  + Using existing base functions in R is like riding a bus – it is relatively straightforward you just need to known which bus to use and know where to get on the get off
  + Being able to use CRAN & Bioconductor packages and functions is like having access to UBER – they can take you more places but may only cover big cities
  + Writing your own functions in R is like driving an SUV with kayak & bike on top – it takes more work, you need to know how to get there. Ultimately, however it will give you the flexibility to go completely new places

## R packages tutorial

* What is a package
  + Basic information about a package is provided in the DESCRIPTION file
  + **Finding the DESCRIPTION files**
    - cran.r-project.org
    - stat.ethz.ch
    - packageDescription(“package”)
    - help(package = “package”)
* what are repositories
  + a place where packages are located so you can install them from it
  + typically they are online and accessible to everyone
  + three of the most popular repositories for R packages:
    - **CRAN** (official repository)
    - **Bioconductor** (topic specific repository)
    - **Github** (most popular repository, but not R specific)
* How to install an R package
  + Installing packages from **CRAN**
    - Install.packages(“package”)
    - Install.packages(c(“vioplot”, “MASS”)) 🡨 install more than a package at a time
  + Installing from **CRAN Mirrors**
    - CRAN is a network of servers (each of them called a “mirror”), so you can specify which one you would like to use
      * Select your mirror by using: chooseCRANmirror()
      * Use the repo parameter: install.packages(“vioplot”, repo = <https://lib.ugent.be/CRAN/>”)
      * See the list of available mirrors: getCRANmirrors()
  + Installing **Bioconductor** packages
    - First install some basic functions needed to install Bioconductor packages
      * source(“https://bioconductor.org/biocLite.R”)
    - install the core packages of Bioconductor
      * biocLite()
    - install just a few particular packages from this repository
      * biocLite(c(“GenomicFeatures”, “AnnotationDbi”))
  + installing packages via **devtools**
    - use the devtools package to simplify the process the installing packages, because it contains specific functions for each repository, including CRAN
      * install devtools
        + install Rtools on Windows: <https://cran.r-project.org/bin/windows/Rtools/>
        + install.packages(“devtools”)
      * then you can use the utility functions to install another packages
        + install\_bioc() from Bioconductor
        + install\_cran() from CRAN
        + install\_git() from git repository
        + install\_github() from GitHub

e.g. devtools::install\_github(“hadley/babyname”)

* + - * + install\_local() from a local file
        + install\_url() from a URL
        + install\_version() from a specific version of a CRAN package
* how to update, remove, and check installed packages
  + check packages
    - installed.packages()
  + uninstalling a package
    - remove.packages(“vioplot”)
  + check what packages need an update
    - old.packages()
  + update all packages
    - update.packages()
  + update specific package
    - install.packages(“vioplot”)
* how to load packages
  + sporadic use of a few functions or data inside a package
    - packagename::functionname()
  + load package into memory for more intensive use
    - library(“package”)
    - require(“package”) 🡨 will install the package if nor installed
* how to load more than one package at a time
  + not possible with library() function
* how to unload a package
  + detach(“package::babynames”, unload = TRUE)
* what are the alternative sources of documentation and help?
  + **Help files**
    - library(babynames)
    - ls(“package:babynames”) 🡨 to see what is inside a loaded package
    - help(function, package = “package”)
  + **vignettes** (documents where the authors show some functionalities of their package ina more detailed way)
    - broseVignettes(package = “packagename”) 🡨 obtain the list of vigenettes included in your installed packages
    - vignette(package = “packagename”) 🡨 show you the list of vignettes
    - vignette(“vignettename”)
* how to choose the right R packages
  + browse categories of CRAN packages
    - **CRAN task view**: <https://cran.r-project.org/web/views/>
  + **RDocumentation** (a help documentation aggregator for R packages from CRAN, BioConductor, and GitHub)