

# R PACKAGES

# OVERVIEW OF R PACKAGES

# WHAT IS AN R PACKAGE?

- From Writing R Extensions: “A directory of files which extend R”.
- Files bundled together using tar and compressed using gzip. The file extension is `.tar.gz`. These are the *source files* for the package, which then must be installed from this source code locally prior to use.
- Sometimes also called an *extension* of R.

# WHAT IS AN R PACKAGE?

Example R package:

Folders	Documents	Developer
<div>weathermetrics</div> <div>PDF Documents</div> <div>weathermetrics.pdf</div> <div>Other</div> <div>weathermetrics_1.2.0.tar.gz</div> <div>weathermetrics_1.2.2.tar.gz</div>	<div>cran-comments.md</div> <div>NEWS.md</div> <div>README.md</div> <div>Folders</div> <div>data</div> <div>inst</div> <div>man</div> <div>R</div> <div>vignettes</div> <div>Other</div> <div>DESCRIPTION</div> <div>NAMESPACE</div> <div>README.Rmd</div> <div>weathermetrics.Rproj</div>	<div>data.R</div> <div>heat_index.R</div> <div>moisture_conversions.R</div> <div>rainmeasure_conversion.R</div> <div>temperature_conversions.R</div> <div>weathermetrics.R</div> <div>wind_conversions.R</div>

# WHAT IS AN R PACKAGE?

You can also have “binary packages” for a certain operating system. From Writing R Extensions:

*A binary package is “a zip file or tarball containing the files of an installed package which can be unpacked rather than installing from sources.”*

# MOTIVATION

## SOFTWARE DEVELOPMENT IN BIOSTATISTICS

So I have a new policy when evaluating CV's of candidates for jobs, or when I'm reading a paper as a referee. If the paper is about a new statistical method or machine learning algorithm and there is no software available for that method - I simply mentally cross it off the CV. If I'm reading a data analysis and there isn't code that reproduces their analysis - I mentally cross it off. In my mind, new methods/analyses without software are just [vapor ware](#). Now, you'd definitely have to cross a few papers off my CV, based on this principle. I do that. But I'm trying really hard going forward to make sure nothing gets crossed off.

Source: Jeff Leek, Simply Statistics

# MOTIVATION

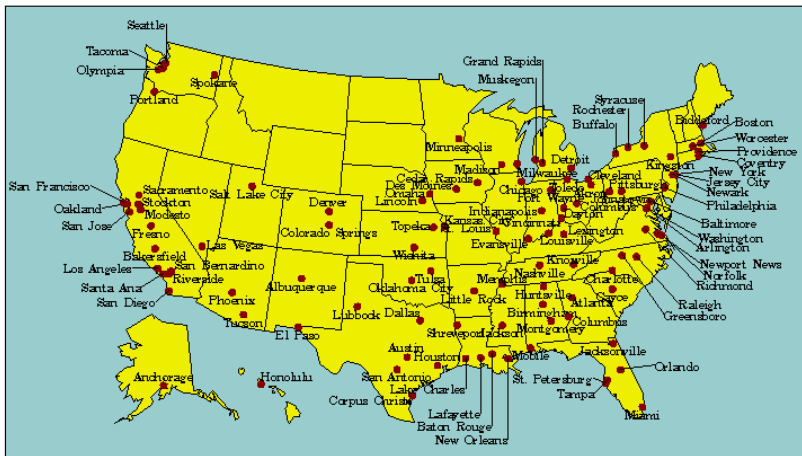
## CONSIDER DEVELOPING SOFTWARE WHEN

- You have developed a new method you want to share
- You have data you'd like to make publicly available
- You find yourself doing the same task repeatedly

## Why create an R package?

- Share some functions broadly
- Share some functions with a small group
- Create a version of code for yourself that's more organized and easier to use
  - Includes documentation (vignettes, help files)
  - Function names linked to package namespace
  - Once library is installed, can load easily

# NMMAPS PACKAGE.



Source: [www.ihapss.jhsph.edu](http://www.ihapss.jhsph.edu)



# CONTENTS OF NMMAPS PACKAGE

## NMMAPSdata package

### Data

- akr
- albu
- Anch  
*and 105 other US cities*
- *Meta-data on cities  
(population,  
location, counties,  
Census variables)*

### Functions

- readCity
- getMetaData  
*and various other  
functions for different  
versions of the package*

### Documentation

- PDF users' manual
- Instructions for each function within R
- Examples for each function within R
- Website

# IMPACT OF NMMAPS PACKAGE

## RESEARCH IMPACTS OF NMMAPS PACKAGE

*Source: Barnett, Huang, and Turner, "Benefits of Publicly Available Data", Epidemiology 2012*

- As of November 2011, 67 publications had been published using this data, with 1,781 citations to these papers
- Research using NMMAPS has been used by the US EPA in creating regulatory impact statements for air pollution (particulates and ozone)
- "Thanks to NMMAPS, there is probably no other country in the world with a greater understanding of the health effects of air pollution and heat waves in its population."

# SHARING AN R PACKAGE

If you want to share your R package, there are a number of ways you can do that:

- CRAN
- GitHub
- Bioconductor: “Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data.” (from the Bioconductor website.)
- Other repositories
  - Private(-ish) repositories: e.g., ROpenSci’s repository (for more, see <https://ropensci.org/blog/blog/2015/08/04/a-drat-repository-for-ropensci>)
  - drat repository: Make your own R package repository, including through GitHub pages.
- Compressed file: You can save a source tarball or binary package file with others without posting to a repository.

# CRAN

## Sharing on CRAN:

- Traditional way to share an R package widely
- Easiest way for others to get your package (`install.packages`)
- Some barriers:
  - Size constraint on packages (5 MB)
  - Must follow CRAN policies
  - All packages must pass a submission process. This is not a guarantee that a package does what it says, just a check that required files are where they should be and that the package more or less doesn't break things.

# CRAN CHECKS



Romain François @romain\_francois · Nov 11

technically trump can go to [#cran](#).

[github.com/romainfrancois...](#)

```
Maintainer: 'Romain François <romain@purple.cat>'
```

```
New submission
```

```
Version contains leading zeroes (2016.11.08)
```

```
Version contains large components (2016.11.08)
```

```
* checking package namespace information ... OK
```

```
* checking package dependencies ... OK
```

```
* checking if this is a source package ... OK
```

```
* checking if there is a namespace ... OK
```

```
* checking for executable files ... OK
```

```
* checking for hidden files and directories ... OK
```

```
* checking for portable file names ... OK
```

```
* checking for sufficient/correct file permissions ... OK
```

```
* checking whether package 'trump' can be installed ... OK
```

```
* checking installed package size ... OK
```

```
* checking package directory ... OK
```

```
* checking DESCRIPTION meta-information ... OK
```

```
* checking top-level files ... OK
```

```
* checking for left-over files ... OK
```

```
* checking index information ... OK
```

```
* checking package subdirectories ... OK
```

```
* checking R files for non-ASCII characters ... OK
```

```
* checking R files for syntax errors ... OK
```

```
* checking whether the package can be loaded ... OK
```

```
* checking whether the package can be loaded with stated dependencies ... OK
```

```
* checking whether the package can be unloaded cleanly ... OK
```

```
* checking whether the namespace can be loaded with stated dependencies ... OK
```

```
* checking whether the namespace can be unloaded cleanly ... OK
```

# GITHUB

GitHub is becoming more and more common as a place to share R packages, both development packages that eventually are posted to CRAN and packages that are never submitted to CRAN.

- No restrictions / submission requirements
- GitHub repository size restrictions (1 GB, no files over 100 MB) much larger than CRAN package size restrictions (5 MB)
- GitHub packages can be installed using `install_github` from the `devtools` package
  - Requires `devtools` package, which has some set-up requirements (XCode for Mac, Rtools for Windows)
- Packages on CRAN cannot depend on packages available only on GitHub

# FIND OUT MORE

To find out more about writing R packages, useful sources are:

- Writing R Extensions: Guidelines for R packages from the R Core Team.
- R Packages by Hadley Wickham
- R package development cheatsheet

## BASIC EXAMPLE PACKAGE



# WEATHERMETRICS

## **weathermetrics: Functions to Convert Between Weather Metrics**

Functions to convert between weather metrics, including conversions for metrics of temperature, air moisture, wind speed, and precipitation. This package also includes functions to calculate the heat index from air temperature and air moisture.

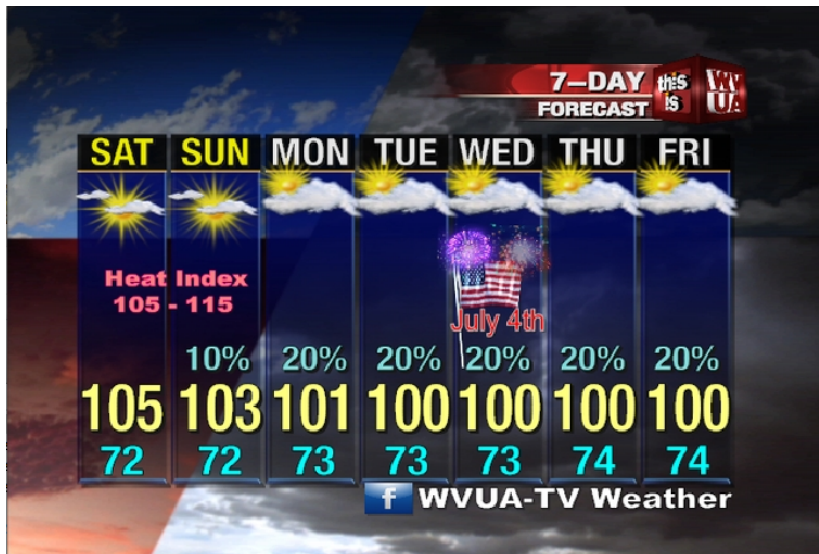
Version: 1.2.2  
Depends: R ( $\geq$  2.10)  
Suggests: [knitr](#), [rmarkdown](#)  
Published: 2016-05-19  
Author: Brooke Anderson [aut, cre], Roger Peng [aut], Joshua Ferreri [aut]  
Maintainer: Brooke Anderson <brooke.anderson at colostate.edu>  
BugReports: <https://github.com/geanders/weathermetrics/issues>  
License: [GPL-2](#)  
URL: <https://github.com/geanders/weathermetrics/>  
NeedsCompilation: no  
Citation: [weathermetrics citation info](#)  
Materials: [NEWS](#)  
CRAN checks: [weathermetrics results](#)

# WEATHERMETRICS

Key functions:

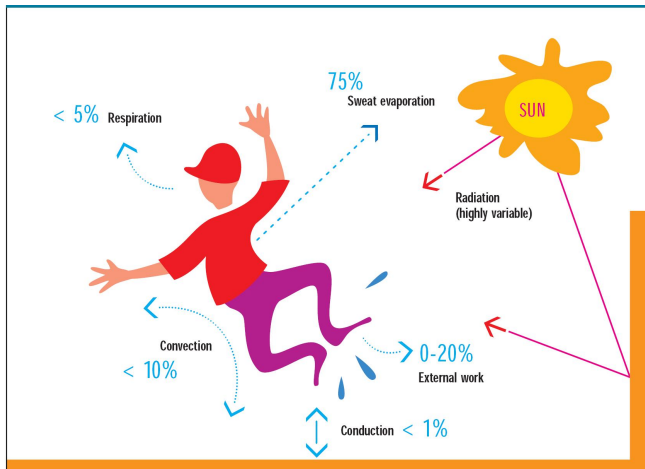
- `convert_temperature`: Convert between temperature metrics
- `convert_precip`: Convert between precipitation metrics
- `convert_wind_speed`: Convert between wind speed metrics
- `heat.index`: Calculates heat index from air temperature and a measure of air moisture (dew point temperature or relative humidity)

# HEAT INDEX



# BODY-ENVIRONMENT TEMPERATURE EXCHANGE

Avenues of temperature exchange between the body and the environment.



Source: Koppe et al., 2003, adapted from Havenith, 2003

# HEAT INDEX AS A MEASURE OF HEAT EXPOSURE

## NOAA's National Weather Service

### Heat Index

Temperature (°F)

	80	82	84	86	88	90	92	94	96	98	100	102	104	106	108	110
40	80	81	83	85	88	91	94	97	101	105	109	114	119	124	130	136
45	80	82	84	87	89	93	96	100	104	109	114	119	124	130	137	
50	81	83	85	88	91	95	99	103	108	113	118	124	131	137		
55	81	84	86	89	93	97	101	106	112	117	124	130	137			
60	82	84	88	91	95	100	105	110	116	123	129	137				
65	82	85	89	93	98	103	108	114	121	128	136					
70	83	86	90	95	100	105	112	119	126	134						
75	84	88	92	97	103	109	116	124	132							
80	84	89	94	100	106	113	121	129								
85	85	90	96	102	110	117	126	135								
90	86	91	98	105	113	122	131									
95	86	93	100	108	117	127										
100	87	95	103	112	121	132										

Likelihood of Heat Disorders with Prolonged Exposure or Strenuous Activity

 Caution

 Extreme Caution

 Danger

 Extreme Danger

# HEAT INDEX ALGORITHMS

The new model is now given by

$$HI = T - 1.0799e^{0.037557[1 - e^{0.0801(D-14)}]}, \quad (4a)$$

where HI,  $T$ , and  $D$  are all in degrees Celsius.

For  $T_a$  we ignore the

effects of wind and radiation and employ Steadman's (1984) regression equation

$$T_a = -1.3 + 0.92T + 2.2e,$$

where  $T$  is in Celsius and  $e$  is in kPa.

$$\begin{aligned} HI = & -42.379 + 2.04901523T + 10.14333127R - 0.22475541TR - 6.83783 \times 10^{-3}T^2 \\ & - 5.481717 \times 10^{-3}R^2 + 1.22874 \times 10^{-3}T^3R + 8.5282 \times 10^{-4}TR^3 - 1.99 \times 10^{-6}T^3R^3 \end{aligned}$$

where  $T$  is an air temperature ( $^{\circ}\text{F}$ ) and  $R$  is a relative humidity (%).

An

equation (available as a FORTRAN program from NCDC) was also provided:

$$\begin{aligned} H_i = & 16.923 + 0.185\,212T + 5.379\,41R \\ & - 0.100\,254TR + 9.4169 \times 10^{-3}T^2 \\ & + 7.288\,98 \times 10^{-3}R^2 + 3.453\,72 \times 10^{-4}T^2R \\ & - 8.149\,71 \times 10^{-4}TR^2 + 1.021\,02 \times 10^{-5}T^2R^2 \\ & - 3.8646 \times 10^{-5}T^3 + 2.915\,83 \times 10^{-5}R^3 \\ & + 1.427\,21 \times 10^{-6}T^3R + 1.974\,83 \times 10^{-7}TR^3 \\ & - 2.184\,29 \times 10^{-8}T^3R^2 + 8.432\,96 \times 10^{-10}T^2R^3 \\ & - 4.819\,75 \times 10^{-11}T^3R^3 + 0.5, \end{aligned}$$

where  $T$  is air temperature ( $^{\circ}\text{F}$ ) and  $R$  is relative humidity (%).

Minimum apparent temperature is a discomfort index based on air and dew point temperatures (14). It is defined as the minimum daily value of the 3-hour apparent temperature values, calculated by using the following formula:  $AT = -2.653 + 0.994 \times T + 0.0153 \times (DT)^2$ , where  $AT$  is apparent temperature,  $T$  is air temperature in  $^{\circ}\text{C}$ , and  $DT$  is dew point temperature in  $^{\circ}\text{C}$ .

The Weather Stress Index [51] is a summer season algorithm and is a derived form of apparent temperature ( $AT$ ):

$$AT = -2.653 + (0.994T_a) + 0.368(T_d)^2, \quad (3)$$

where  $T_a$  = air temperature ( $^{\circ}\text{C}$ );  $T_d$  = dewpoint temperature ( $^{\circ}\text{C}$ ).

# NWS HEAT INDEX ALGORITHM

The screenshot shows a web browser window titled "Heat Index Calculation" with the URL <http://www.hpc.ncep.noaa.gov/html/heatindex.shtml>. The page header features the NOAA logo and the text "National Weather Service Hydrometeorological Prediction Center". A navigation bar includes links for "Site Map", "News", "Organization", and a "Search" field. Below the navigation bar, a menu lists "DOC NOAA NWS" and "NCEP Centers: AWC CPC EMC HPC NGO NHC OPC SPC SWPC".

The main content area is titled "Meteorological Conversions and Calculations" in red. Below this is the "Heat Index Calculator" section, which instructs users to "Choose the appropriate calculator and enter the values. Then click calculate." There are two calculator options:

- Using Dew Point Temperature:** This calculator has input fields for "Air Temperature" (in °F and °C) and "Dew Point Temperature" (in °F and °C). It includes "Calculate" and "Reset" buttons and a "Heat Index =" output field.
- Using Relative Humidity:** This calculator has input fields for "Air Temperature" (in °F and °C) and "Relative Humidity" (in %). It includes "Calculate" and "Reset" buttons and a "Heat Index =" output field.

The left sidebar contains additional links and information, including "Local forecast by 'City, St' or Zip Code", "Search HPC", "Find us on Facebook", "HPC on Facebook", "NCEP Quarterly Newsletter", and a list of "HPC Home" links such as "Analyses and Forecasts", "National Forecast Charts", "National High & Low", "HPC Discussions", "Surface Analysis", "Days 1-2 CONUS", "Days 3-7 CONUS", "Days 4-8 Alaska", "QPF", "PQPF", "Excessive Rainfall", and "Rainfall".

# CONTENTS OF WEATHERMETRICS PACKAGE

## weathermetrics package

### Data

- lyon
- newhaven
- norfolk
- suffolk

*Small weather  
datasets to use in  
examples for function  
(Weather  
Underground)*

### Functions

- heat.index
- convert\_temperature
- convert\_wind\_speed
- convert\_precip
- dewpoint.to.humidity
- humidity.to.dewpoint

### Documentation

- PDF users' manual
- Instructions for each function within R
- Examples for each function within R



# CONVERT FROM CELSIUS TO FAHRENHEIT

EQUATION TO CONVERT FROM CELSIUS TO FAHRENHEIT

$$T_F = \frac{9}{5} T_C + 32$$

```
celsius.to.fahrenheit
```

```
## function (T.celsius, round = 2)
##      {
##          T.fahrenheit <- (9/5) * T.celsius + 32
##          T.fahrenheit <- round(T.fahrenheit, digits =
##          return(T.fahrenheit)
##      }
## <environment: namespace:weathermetrics>
```

# CONVERT TEMPERATURES

`convert_temperature {weathermetrics}`

R Documentation

## Convert from one temperature metric to another

### Description

This function allows you to convert a vector of temperature values between Fahrenheit, Celsius, and degrees Kelvin.

### Usage

```
convert_temperature(temperature, old_metric, new_metric, round = 2)
```

### Arguments

**temperature** A numeric vector of temperatures to be converted.

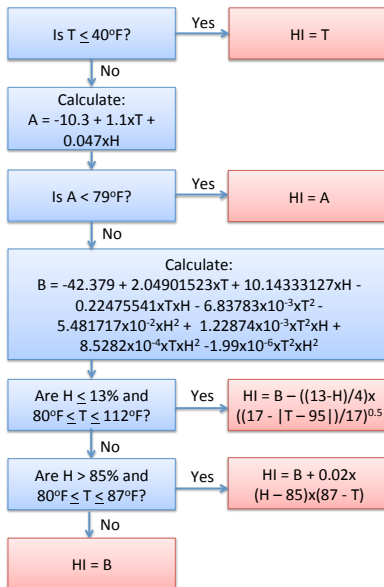
**old\_metric** The metric from which you want to convert. Possible options are:

- `fahrenheit, f`
- `kelvin, k`
- `celsius, c`

**new\_metric** The metric to which you want to convert. The same options are possible as for `old_metric`.

**round** An integer indicating the number of decimal places to round the converted value.

# NWS HEAT INDEX ALGORITHM



# NWS HEAT INDEX ALGORITHM

```
head(heat.index.algorithm, 10)
```

```
##  
## 1  function (t = NA, rh = NA)  
## 2  {  
## 3      if (is.na(rh) | is.na(t)) {  
## 4          hi <- NA  
## 5      }  
## 6      else if (t <= 40) {  
## 7          hi <- t  
## 8      }  
## 9      else {  
## 10         alpha <- 61 + ((t - 68) * 1.2) + (rh * 0.094)
```

# NWS HEAT INDEX ALGORITHM

```
data(suffolk)
suffolk %>%
  mutate(heat_index = heat.index(t = TemperatureF,
                                  rh = Relative.Humidity)) %>%
  slice(1:5)
```

		Date	TemperatureF	Relative.Humidity	heat_index
##	1	1998-07-12	72	69	72
##	2	1998-07-13	73	66	73
##	3	1998-07-14	74	74	75
##	4	1998-07-15	78	86	80
##	5	1998-07-16	78	100	81

## ELEMENTS OF AN R PACKAGE

# BASIC ELEMENTS

R packages can include a number of different elements. We'll cover more of them in later classes. There are a few common elements, though. They can be split into two groups: things you edit directly, and things that are automatically written.

Things you edit directly:

- DESCRIPTION file: The package's "Title page". Metadata on the package, including names and contacts of authors, package name, and description. This file also lists all the package *dependencies* (other packages with functions this package uses).
- R folder: R code defining functions in the package. All code is included in one or more R scripts. If you use Roxygen for help documentation, all of that is also included in these files.

Things that are automatically written:

- man folder: Help documentation for each function. These files are automatically rendered if you use Roxygen.
- NAMESPACE file

# DESCRIPTION FILE

Required elements:

- Package: Name of the package
- Version: Number of the current version of the package (e.g., 0.1.0)
- Title: Short title for the package, in title case and in 65 characters or less.
- Author and Maintainer (these two sections can be replaced with Authors@R section that uses the person function)
- Description: Paragraph describing the package
- License: Name of the license the package is under. If necessary, you can also refer to a LICENSE file included as another file in the package. Only some licenses are easily accepted by CRAN.



# DESCRIPTION FILE

Other elements that are common but not required:

- Date: Release date of this version of the package.
- Imports: A list of the packages on which this package depends: other packages with functions used by the code in this package.
- URL: If there is a webpage associated with the package, the address for it. Often, this is the web address of the package's GitHub repository.
- BugReports: Where users can submit problems they've had. Often, the web address of the "Issues" page of the GitHub repository for the package.

# PACKAGE NAME

The format requirements for a package name are, based on Writing R Extensions:

*“This should contain only (ASCII) letters, numbers and dot, have at least two characters and start with a letter and not end in a dot.”*

Hadley Wickham's additional guidelines:

- Make it easy to Google.
- Make it all uppercase or all lower case
- Base it on a word that's easy to remember, but then tweak the spelling to make it unique (and easier to Google).
- Abbreviate.
- Add an “r”.

# PACKAGE MAINTAINER

A package can have many authors, but only one maintainer. The maintainer is in charge of fixing any problems that come up with CRAN checks over time to keep the package on CRAN. The maintainer is also the person who will be emailed about bugs, etc., by other users.

The package can have other authors, as well as people in other roles (e.g., contributor). See the helpfile for the `person` function for more on the codes used for different roles.

# DESCRIPTION FILE

Package: weathermetrics

Type: Package

Title: Functions to Convert Between Weather Metrics

Version: 1.2.2

Date: 2016-05-19

Authors@R: c(person("Brooke", "Anderson",  
email = "brooke.anderson@colostate.edu",  
role = c("aut", "cre")),  
person("Roger", "Peng",  
email = "rdpeng@gmail.com", role = c("aut")),  
person("Joshua", "Ferrerri",  
email = "joshua.m.ferrerri@gmail.com", role = c("aut")))

Description: Functions to convert between weather metrics,  
including conversions for metrics of temperature, air  
moisture, wind speed, and precipitation. This package also  
includes functions to calculate the heat index from  
air temperature and air moisture.

(cont. on next slide)
























# DESCRIPTION FILE

```
URL: https://github.com/geanders/weathermetrics/
BugReports: https://github.com/geanders/weathermetrics/issues
License: GPL-2
LazyData: true
RoxygenNote: 5.0.1
Depends:
  R (>= 2.10)
Suggests: knitr,
  rmarkdown
VignetteBuilder: knitr
```

# R FOLDER

The R folder of the package includes:

- R scripts with code defining all functions for the package
- Help documentation for each function (if using Roxygen)
- Help documentation for the package data in “data.R”

Folders	Documents	Developer
 weathermetrics	 cran-comments.md	 data.R
PDF Documents	 NEWS.md	 heat_index.R
 weathermetrics.pdf	 README.md	 moisture_conversions.R
Other	Folders	 rainmeasure_conversion.R
 weathermetrics_1.2.0.tar.gz	 data	 temperature_conversions.R
 weathermetrics_1.2.2.tar.gz	 inst	 weathermetrics.R
	 man	 wind_conversions.R
	 R	
	 vignettes	
	Other	
	 DESCRIPTION	
	 NAMESPACE	
	 README.Rmd	
	 weathermetrics.Rproj	

# R FOLDER

You define functions in the R scripts just as you would anytime you want to define a function in R. For example, “temperature\_conversions.R” includes the following code to define converting from Celsius to Fahrenheit:

```
celsius.to.fahrenheit <- function (T.celsius, round = 2) {  
  T.fahrenheit <- (9/5) * T.celsius + 32  
  T.fahrenheit <- round(T.fahrenheit, digits = round)  
  return(T.fahrenheit)  
}
```

Only exception: use `package::function` syntax to call functions from other packages (e.g., `dplyr::mutate()`).

# R FOLDER

Using `roxygen2`, you put all information for the help files directly into a special type of code comments right before defining the function.

- Start each line with `#'`.
- To render into help files, use the `document` function from the `devtools` package.
- This will write out help files in the `man` folder of the package.
- Use these comments to specify which functions should be *exported* from the package using the `@export` tag. This information will be used to render the `NAMESPACE` file for the package.



# R FOLDER

```
#' Convert from Celsius to Fahrenheit.  
#'  
#' \code{celsius.to.fahrenheit} creates a numeric vector of  
#'   temperatures in Fahrenheit from a numeric vector of  
#'   temperatures in Celsius.  
#'  
#' @param T.celsius Numeric vector of temperatures in Celsius.  
#' @inheritParams convert_temperature  
#'  
#' @return A numeric vector of temperature values in Fahrenheit.  
#'  
#' @note Equations are from the source code for the US National  
#'   Weather Service's  
#'   \href{http://www.wpc.ncep.noaa.gov/html/heatindex.shtml}  
#'   {online heat index calculator}.
```

(cont. on next slide)

# R FOLDER

```
#' @author
#' Brooke Anderson \email{brooke.anderson@@colostate.edu},
#' Roger Peng \email{rdpeng@gmail.com}
#'
#' @seealso \code{\link{fahrenheit.to.celsius}}
#'
#' @examples # Convert from Celsius to Fahrenheit.
#' data(lyon)
#' lyon$TemperatureF <- celsius.to.fahrenheit(lyon$TemperatureC)
#' lyon
#'
#' @export
```

# R FOLDER

Once you run document, this is all rendered as a help file. Now, when you run `?celsius.to.fahrenheit`, you'll get:

`celsius.to.fahrenheit` {weathermetrics}

R Documentation

## Convert from Celsius to Fahrenheit.

### Description

`celsius.to.fahrenheit` creates a numeric vector of temperatures in Fahrenheit from a numeric vector of temperatures in Celsius.

### Usage

```
celsius.to.fahrenheit(T.celsius, round = 2)
```

### Arguments

**T.celsius** Numeric vector of temperatures in Celsius.

**round** An integer indicating the number of decimal places to round the converted value.

### Value

A numeric vector of temperature values in Fahrenheit.

### Note

Equations are from the source code for the US National Weather Service's [online heat index calculator](#).

### Author(s)

Brooke Anderson [brooke.anderson@colostate.edu](mailto:brooke.anderson@colostate.edu), Roger Peng [rdpeng@gmail.com](mailto:rdpeng@gmail.com)

# R FOLDER

The start of the NAMESPACE fill will be automatically written when you run document and will look like:

```
# Generated by roxygen2: do not edit by hand
```

```
export(celsius.to.fahrenheit)
export(celsius.to.kelvin)
export(convert_precip)
export(convert_temperature)
export(convert_wind_speed)
export(dewpoint.to.humidity)
```

# R FOLDER

Some of the most common tags you'll use for roxygen2 are:

- `@param`: Use to explain parameters for the function.
- `@inheritParam`: If you have already explained a parameter for the help file for a different function, you can use this tag to use the same definition for this function.
- `@return`: Explanation of the object returned by the function.
- `@examples`: One or more examples of using the function.
- `@export`: Export the function, so it's available when users load the package.

By default, the first line in the roxygen2 comments is the function title and the next section is the function description. For more on roxygen2, see: <https://cran.r-project.org/web/packages/roxygen2/vignettes/roxygen2.html>

# R FOLDER

If you are automating helpfile documentation, you must also include an R script with the documentation for each data set that comes with the package.

This file will include `roxygen2` documentation for each data set, followed by the name of the dataset in quotation marks.

As an example, the next slide has the documentation in the “data.R” file for the “lyon” data set.

# R FOLDER

```
#' Weather in Lyon, France
#'
#'| Daily values of mean temperature (Celsius) and mean dew
#'| point temperature (Celsius) for the week of June 18, 2000,
#'| in Lyon, France.
#'|
#'| @source \href{http://www.wunderground.com/}
#'|           {Weather Underground}
#'|
#'| @format A data frame with columns:
#'|   \describe{
#'|     \item{Date}{Date of weather observation}
#'|     \item{TemperatureC}{Daily mean temperature in Celsius}
#'|     \item{DewpointC}{Daily mean dewpoint temperature in
#'|                       Celsius}
#'|   }
#'|
"lyon"
```

# OTHER COMMON ELEMENTS

Some other elements, while not required, are common in many R packages:

- data folder: R objects with data that goes with the package. Often, these are small-ish data files for examples of how to use package functions. However, more “scientific” packages may include more substantive data in this folder. Some packages are created solely to deliver data.
- vignettes folder: One or more tutorials on why the package was created and how to use it. These can be written in RMarkdown.
- NEWS file: Information about changes in later versions of the package.
- .Rbuildignore file: Lists files and directories that should not be included in the package build
- LICENSE file: With certain licenses (MIT is a common example), you need a separate LICENSE file, to supplement the license information in the DESCRIPTION file.



# LESS COMMON ELEMENTS

- `src` folder: Sources and headers for compiled code (e.g., C++).
- `demo` folder: R scripts that give demonstrations of using the package.
- `tests` folder: Test code for the package. Currently, the best way to create tests for a package are with the `testthat` package.
- `inst` folder: Various and sundries, including a CITATION file to tell others how to cite your package and executable scripts not in R (e.g., shell scripts, Perl or Python code).

## CREATING AN R PACKAGE

# CREATING AN R PACKAGE

Invaluable tools when creating an R package:

- The `devtools` package: Various utility functions that help you develop an R package.
- *R Packages* by Hadley Wickham. Available from O'Reilly or free online at <http://r-pkgs.had.co.nz>
- GitHub: When in doubt of how to structure something, look for examples in code for other R packages. GitHub is currently the easiest way to browse through the code for many R packages.