03-01: Packages

1 - Purpose

- Introduce R packages
- Connect to the NOAA/NCDC database using a package
- Get weather data from the NOAA/NCDC database

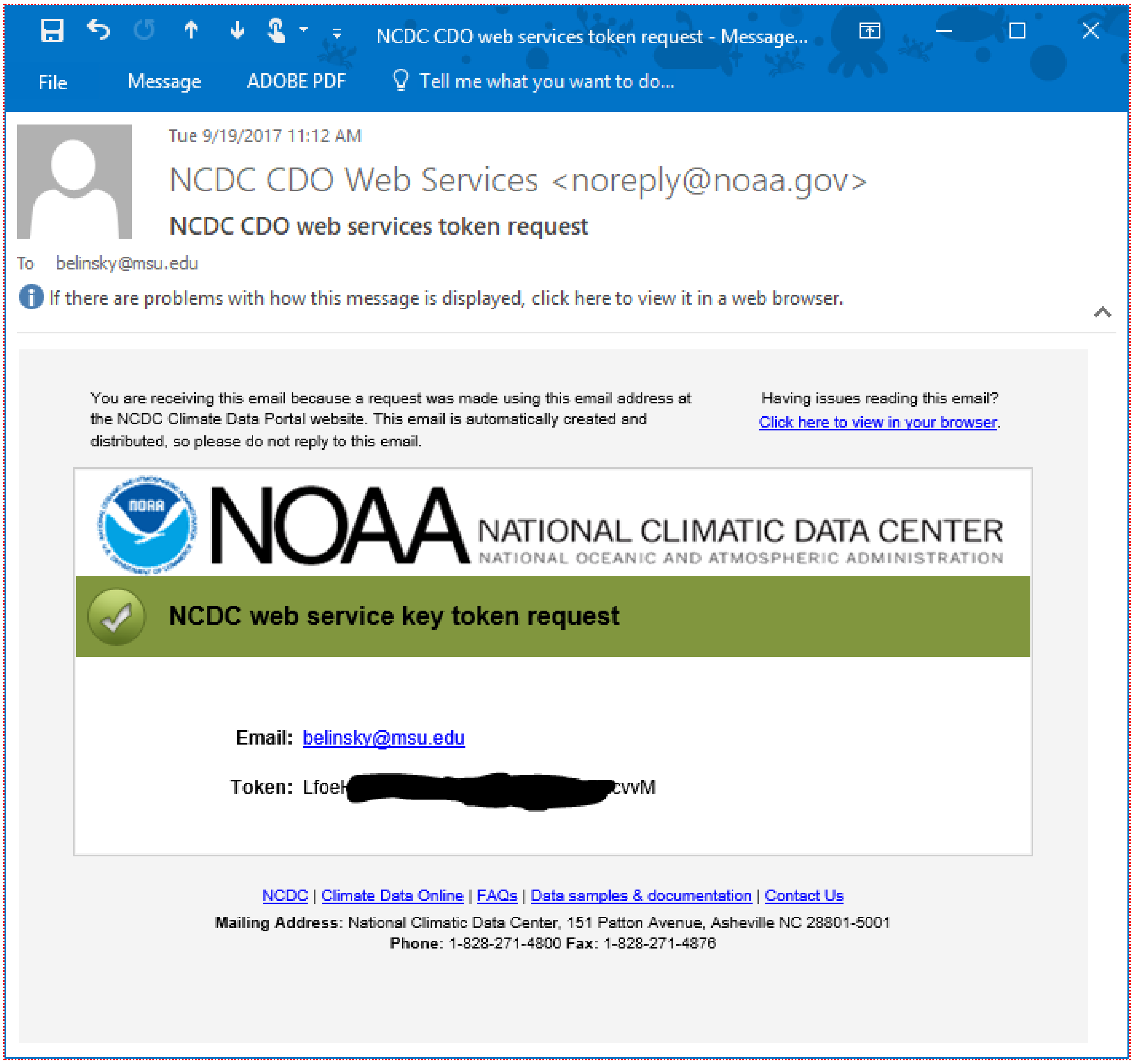
2 - Class Project or other questions...

If you have any questions about your Class Project or the material in this lesson *feel free to email them to the instructor here*.

3 - The NOAA/NCDC Climate database

For the majority of the unit, we will be using weather data from the NOAA/NCDC governmental databases. The weather databases are free to use but you first need to get a *token*, which is similar to a password. So, before we write script to connect to the database, let's get the token.

You can register for a token on this page. Just put in your email address and NOAA/NCDC will send a very random looking 32 character string to your email. This 32 character string is your token and allows you to access the climate database. Don't forget to check your SPAM folder for the email if you don't receive it right away. I have tested it a few times and got the email within 1 minute each time.


NCDC CDO web services token request - Message...
File
Message
ADOBE PDF
Tell me what you want to do...

Tue 9/19/2017 11:12 AM

NCDC CDO Web Services <noreply@noaa.gov>

NCDC CDO web services token request

To belinsky@msu.edu

If there are problems with how this message is displayed, click here to view it in a web browser.

You are receiving this email because a request was made using this email address at the NCDC Climate Data Portal website. This email is automatically created and distributed, so please do not reply to this email.

Having issues reading this email? Click here to view in your browser.

NOAA NATIONAL CLIMATIC DATA CENTER
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

NCDC web service key token request

Email: belinsky@msu.edu

Token: Lfoe[redacted]cvvM

NCDC | Climate Data Online | FAQs | Data samples & documentation | Contact Us

Mailing Address: National Climatic Data Center, 151 Patton Avenue, Asheville NC 28801-5001
Phone: 1-828-271-4800 **Fax:** 1-828-271-4876

Fig 1: The email from NOAA/NCDC with your token.

3.1 - Using toolbox.r for the token

The token is used as a parameter whenever you make a call to the NOAA/NCDC database in your script. It would be nice to keep the 32 character token in a place where you can access it easily and not have to remember it. One place to keep the token is inside your ***toolbox.r*** script file.

Just add this line at the top of the ***toolbox.r*** file:

```
myToken = "put your 32 character token here";
```

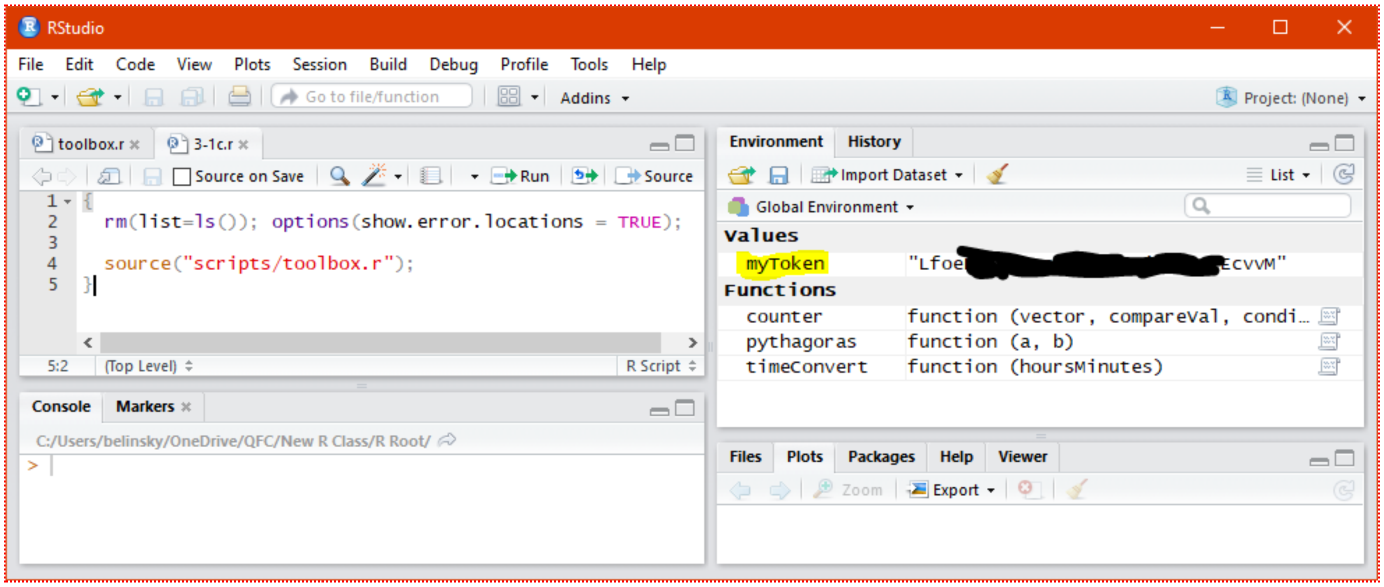
Now you just need to include ***toolbox.r*** in any script where you access the NOAA/NCDC database and you can use ***myToken*** to represent the 32 character token.

Let's start a new script that includes ***toolbox.r.*** To include a script file inside another script file, you use the ***source()*** function and set the parameter ***file*** to the location of **toolbox.r**. Note: *"scripts/toolbox.r"* assumes that you have saved ***toolbox.r*** inside the ***scripts*** folder in your ***R Root*** folder.

```
rm(list=ls());  options(show.error.locations = TRUE);

source(file="scripts/toolbox.r");
```

Because you have included ***toolbox.r***, you now have access to ***myToken*** inside your new script file.



```
{
  rm(list=ls()); options(show.error.locations = TRUE);

  source("scripts/toolbox.r");
}
```

```
Values
  myToken        "Lfoe[redacted]EcvvM"
Functions
  counter        function (vector, compareVal, condi…
  pythagoras     function (a, b)
  timeConvert    function (hoursMinutes)
```

```
C:/Users/belinsky/OneDrive/QFC/New R Class/R Root/
>
```

*Fig 2: Including the **toolbox.r** script with the **myToken** variable.*

4 - Packages (or, other peoples' functions)

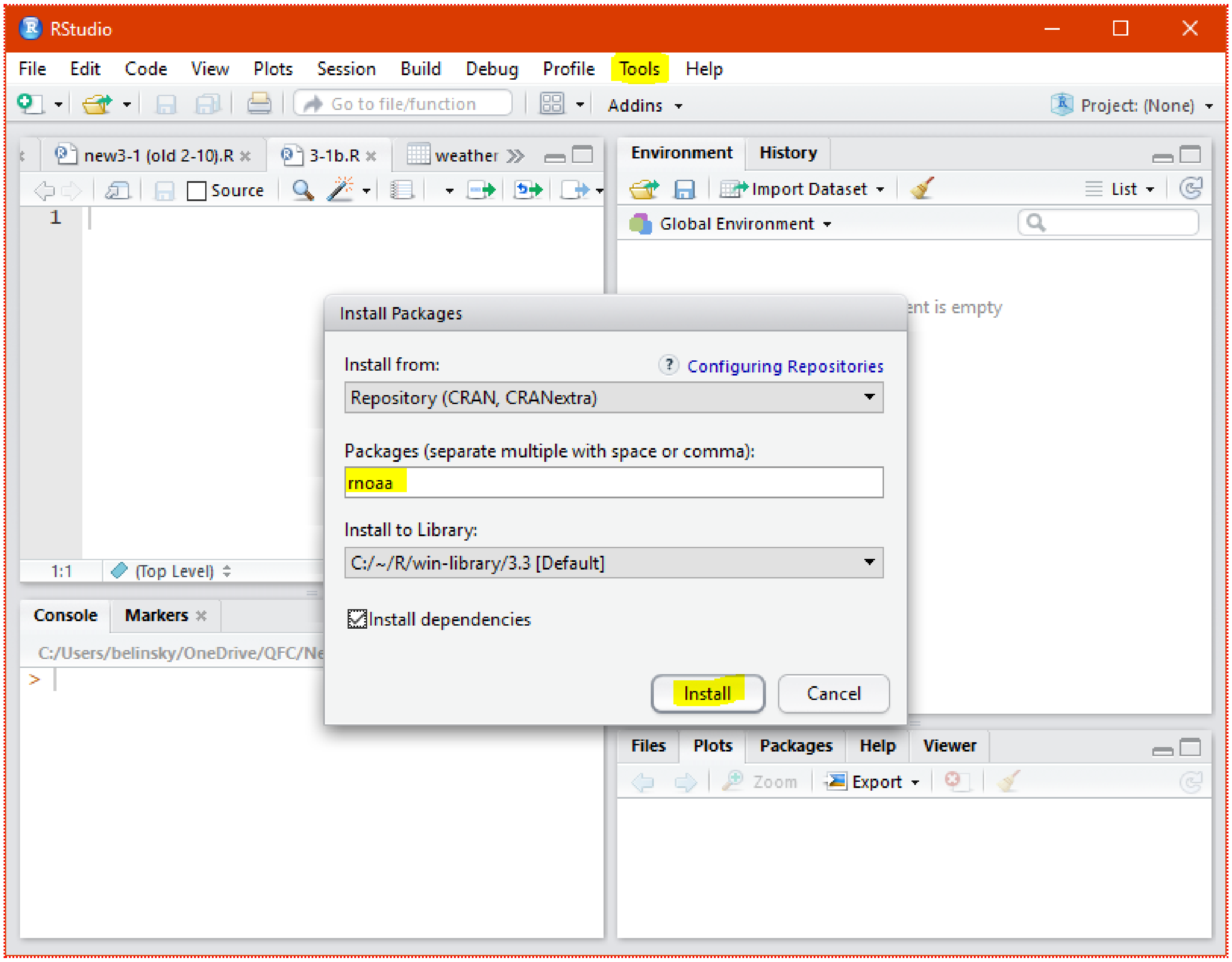
In the last lesson we used some plot functions like ***boxplot()*** and ***hist()***. These functions exist in the R Base Package -- the R Base Package is script that contains a large list of built-in functions. However, as we saw in the last lesson, you can create a file that contains your own functions. In fact, many people have done this and created their own public scripts that contain related functions. These public scripts are called *packages*.

In the next few lessons we will work with two packages: **rnoaa** and **reshape2**. The functions in the **rnoaa** package allow you to connect to, and gather weather information from, the NOAA/NCDC online weather database. The functions in the **reshape2** package allow you to manipulate (reshape) your data frame.

To use the functions inside the **rnoaa** and **reshape2** packages, you need to:
1. install **rnoaa** and **reshape2** in R
2. include **rnoaa** and **reshape2** in your script

4.1 - Installing the rnoaa and reshape2 package

To install the **rnoaa** package in R
1. click on ***Tools -> Install Packages***...
2. type in **rnoaa** in the Packages textbox
3. click on ***Install*** (make sure "Install Dependencies" is checked)



Install Packages

Install from: Repository (CRAN, CRANextra)

Configuring Repositories

Packages (separate multiple with space or comma): rnoaa

Install to Library: C:/~/R/win-library/3.3 [Default]

[✓] Install dependencies

Install   Cancel

*Fig 3: Installing the **rnoaa** package in R*

Repeat the above steps for the ***reshape2*** package. These steps will work for any package you want to install in the future.

Extension: Dependencies

4.2 - Including the packages in your script

You still need to tell your script that you are going to use the functions in the **rnoaa** and **reshape2** package, which is done using the ***library()*** function with the parameter ***package***.

```
rm(list=ls());  options(show.error.locations = TRUE);

source(file="script/toolbox.r");
library(package="rnoaa");    # include functions from the rnoaa package
library(package="reshape2"); # include functions from the reshape2 package
```

Now your script has access to all of the **rnoaa** and **reshape2** functions -- and, of course, your ***toolbox.r*** script, which has ***myToken***.

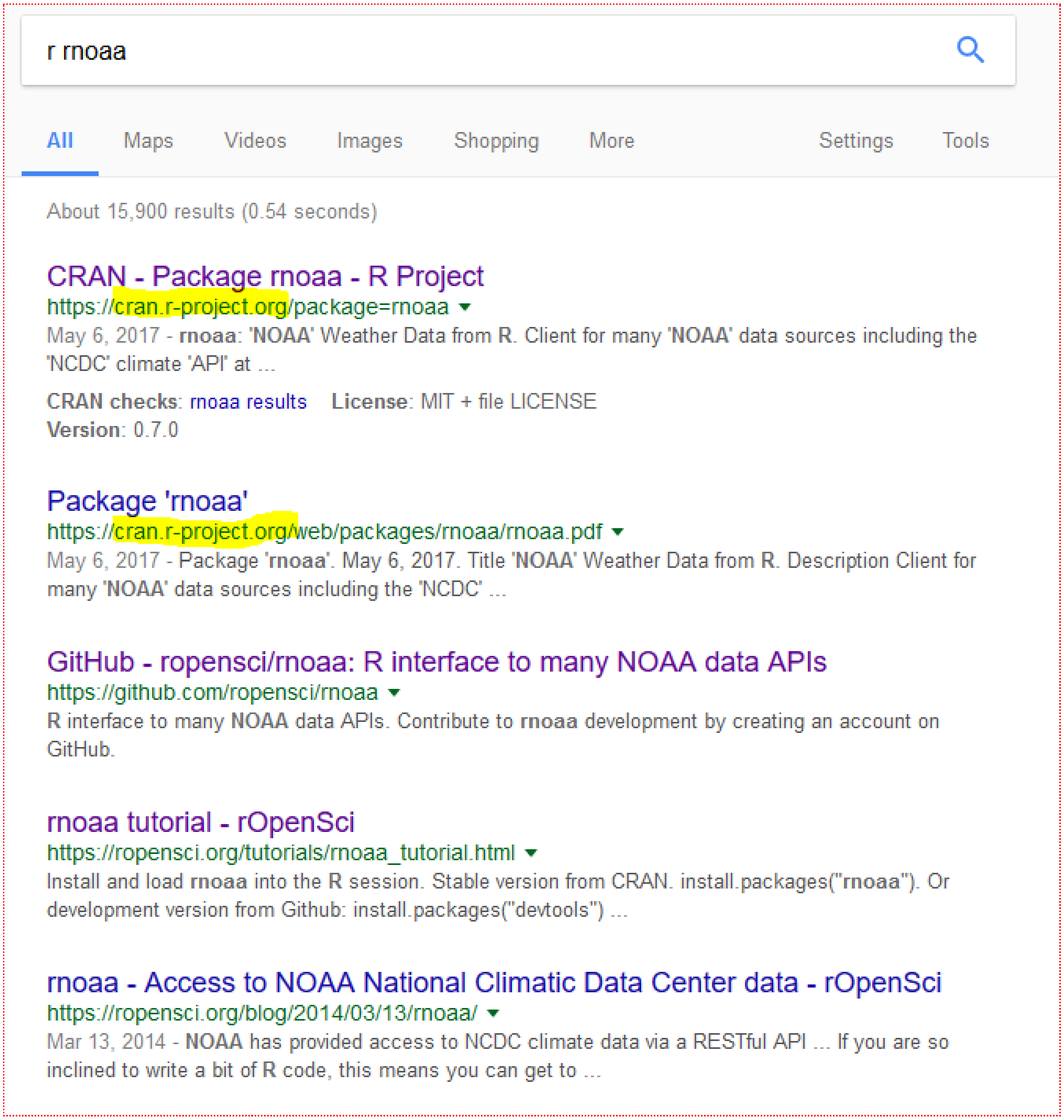
5 - The rnoaa package

A huge disclaimer: the ***rnoaa*** package espouses the problems that exists in many R packages, which includes poor documentation and unintuitive function naming.

The purpose of this lesson is not to give a complete treatise on how to use packages -- it is to give you a starting point and make using packages a little less intimidating.

The starting point for finding most R packages is The Comprehensive R Archive Network (CRAN) website. This website holds pretty much everything R related and you will often see it on R package-related internet searches.

When I do a search for "r rnoaa" in google I get:



r rnoaa

About 15,900 results (0.54 seconds)

CRAN - Package rnoaa - R Project
https://cran.r-project.org/package=rnoaa
May 6, 2017 - rnoaa: 'NOAA' Weather Data from R. Client for many 'NOAA' data sources including the 'NCDC' climate 'API' at ...
CRAN checks: rnoaa results License: MIT + file LICENSE
Version: 0.7.0

Package 'rnoaa'
https://cran.r-project.org/web/packages/rnoaa/rnoaa.pdf
May 6, 2017 - Package 'rnoaa'. May 6, 2017. Title 'NOAA' Weather Data from R. Description Client for many 'NOAA' data sources including the 'NCDC' ...

GitHub - ropensci/rnoaa: R interface to many NOAA data APIs
https://github.com/ropensci/rnoaa
R interface to many NOAA data APIs. Contribute to rnoaa development by creating an account on GitHub.

rnoaa tutorial - rOpenSci
https://ropensci.org/tutorials/rnoaa_tutorial.html
Install and load rnoaa into the R session. Stable version from CRAN. install.packages("rnoaa"). Or development version from Github: install.packages("devtools") ...

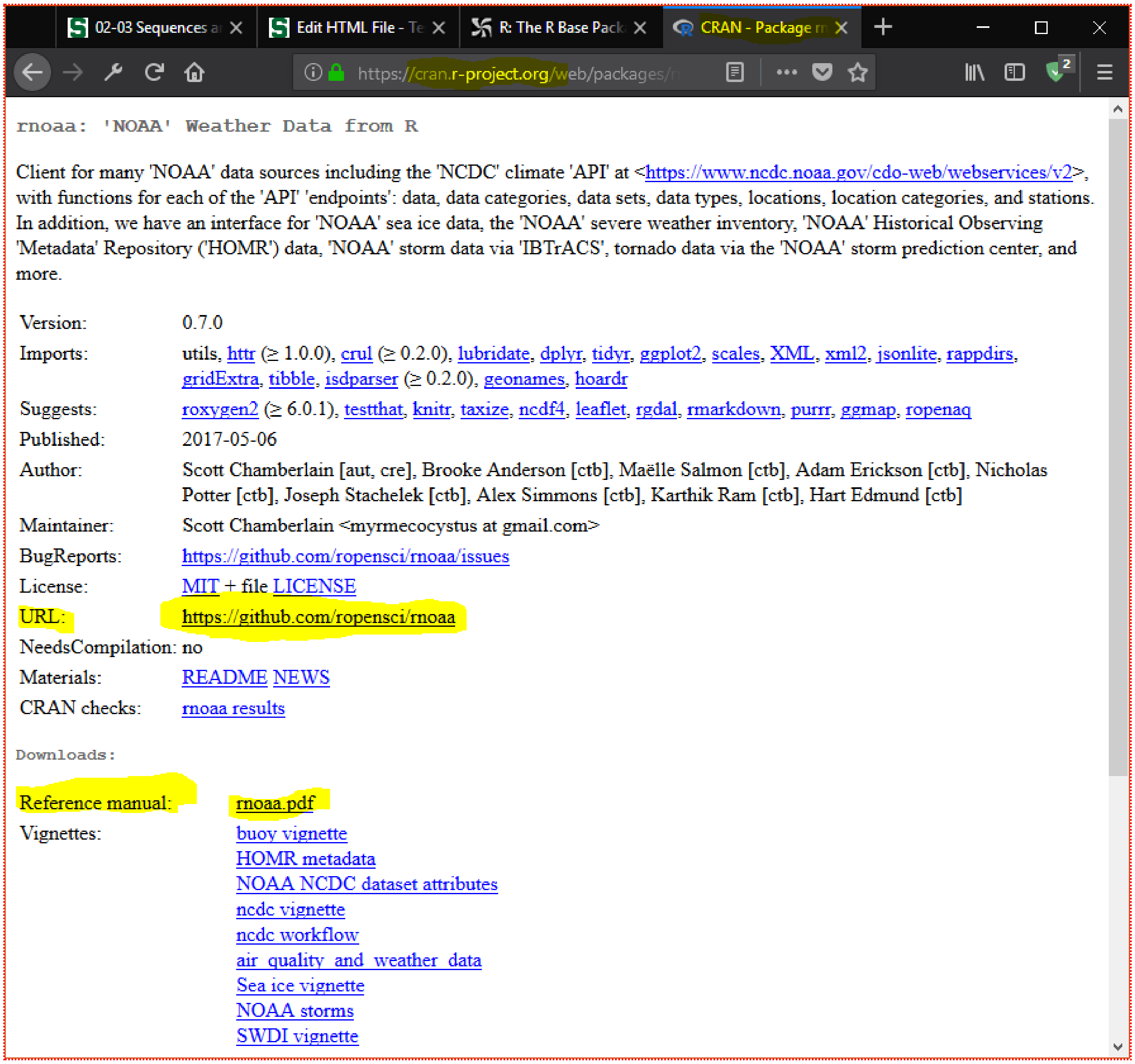
rnoaa - Access to NOAA National Climatic Data Center data - rOpenSci
https://ropensci.org/blog/2014/03/13/rnoaa/
Mar 13, 2014 - NOAA has provided access to NCDC climate data via a RESTful API ... If you are so inclined to write a bit of R code, this means you can get to ...

Fig 4: A google search for "r rnoaa" -- with results from the CRAN website.

The first two results are from the CRAN website (cran.r-project.org) and the next three are linked inside the CRAN website. So, when you see CRAN or cran.r-project.org, you probably have a good starting point for learning about the package.

5.1 - Package home page

For every package homepage inside CRAN, there are two links that are important: the **URL** and the **Reference Manual**. The URL takes you to the main page for the package and the reference manual is a pdf that lists every function inside the package.



rnoaa: 'NOAA' Weather Data from R

Client for many 'NOAA' data sources including the 'NCDC' climate 'API' at <https://www.ncdc.noaa.gov/cdo-web/webservices/v2>, with functions for each of the 'API' 'endpoints': data, data categories, data sets, data types, locations, location categories, and stations. In addition, we have an interface for 'NOAA' sea ice data, the 'NOAA' severe weather inventory, 'NOAA' Historical Observing 'Metadata' Repository ('HOMR') data, 'NOAA' storm data via 'IBTrACS', tornado data via the 'NOAA' storm prediction center, and more.

Version:	0.7.0
Imports:	utils, httr (≥ 1.0.0), crul (≥ 0.2.0), lubridate, dplyr, tidyr, ggplot2, scales, XML, xml2, jsonlite, rappdirs, gridExtra, tibble, isdparser (≥ 0.2.0), geonames, hoardr
Suggests:	roxygen2 (≥ 6.0.1), testthat, knitr, taxize, ncdf4, leaflet, rgdal, rmarkdown, purrr, ggmap, ropenaq
Published:	2017-05-06
Author:	Scott Chamberlain [aut, cre], Brooke Anderson [ctb], Maëlle Salmon [ctb], Adam Erickson [ctb], Nicholas Potter [ctb], Joseph Stachelek [ctb], Alex Simmons [ctb], Karthik Ram [ctb], Hart Edmund [ctb]
Maintainer:	Scott Chamberlain <myrmecocystus at gmail.com>
BugReports:	https://github.com/ropensci/rnoaa/issues
License:	MIT + file LICENSE
URL:	https://github.com/ropensci/rnoaa
NeedsCompilation:	no
Materials:	README NEWS
CRAN checks:	rnoaa results

Downloads:

Reference manual:	rnoaa.pdf
Vignettes:	buoy vignette HOMR metadata NOAA NCDC dataset attributes ncdc vignette ncdc workflow air_quality_and_weather_data Sea ice vignette NOAA storms SWDI vignette

Fig 5: The homepage for the rnoaa package.

The ***rnoaa*** pdf lists way more functions than you will ever use. Most of the functions listed are *helper functions*, or functions used by other functions inside the package. Helper function are rarely directly called by the user but the documentation does not differentiate the helper functions from the main functions.

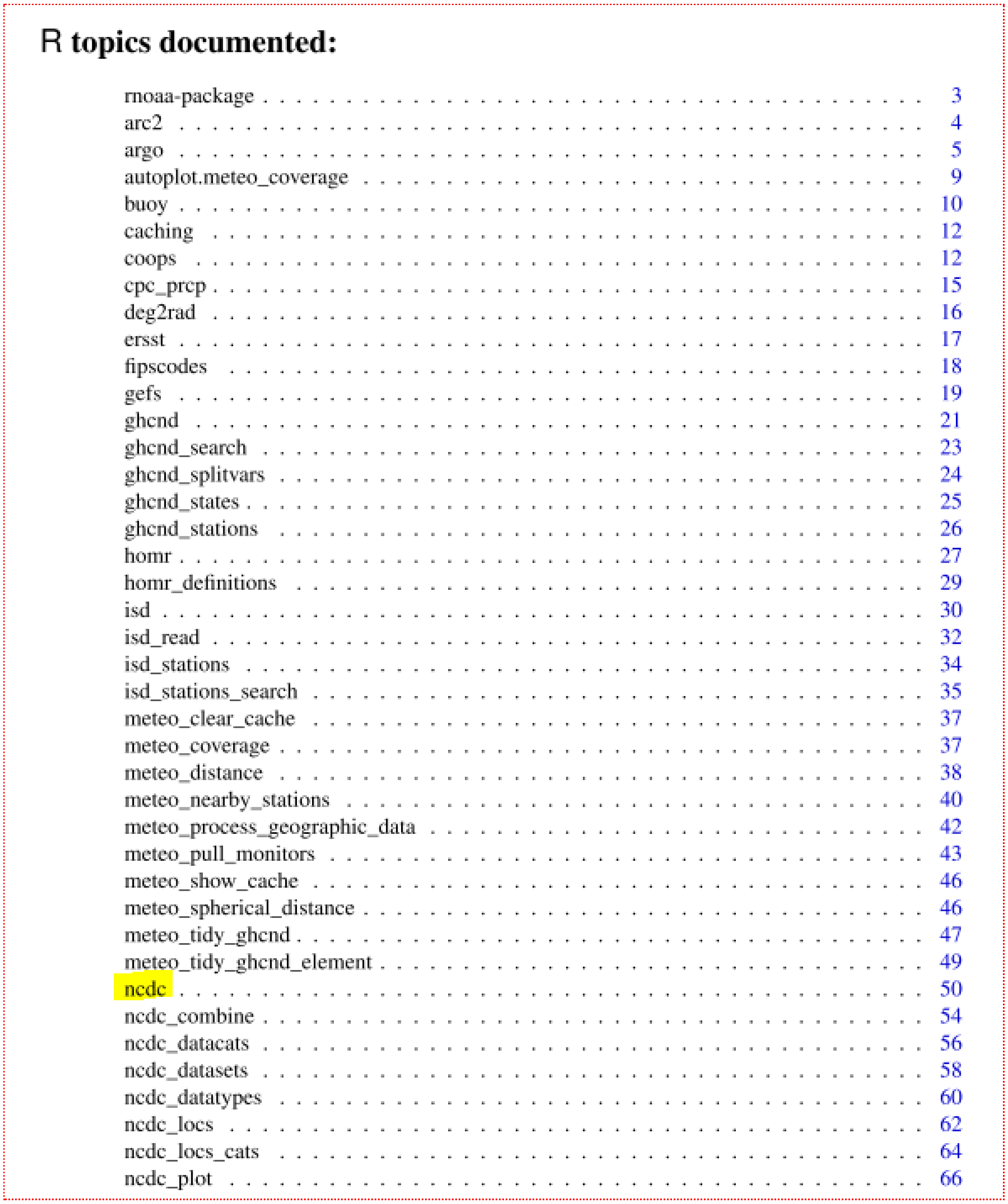

R topics documented:

Fig 6: A partial list of functions on page 2 of the rnoaa pdf.

Finding out which functions to use can be tricky but if you go through the community page at the **URL** you will notice that the function ***ncdc()*** is often used to get weather data. In fact ***ncdc()*** is meant to be the broad function that links to the NOAA/NCDC databases and this is the function we will use.

6 - Getting data using the rnoaa package

We want to get all of the weather information for 2016 from Lansing, Michigan. To do this we will use the ***ncdc()*** function in the **weatherData** package.

ncdc() looks like this (from page 9 on the rnoaa.pdf):

```
ncdc(datasetid = NULL, datatypeid = NULL, stationid = NULL,
    locationid = NULL, startdate = NULL, enddate = NULL, sortfield = NULL,
    sortorder = NULL, limit = 25, offset = NULL, token = NULL,
    dataset = NULL, datatype = NULL, station = NULL, location = NULL,
    locationtype = NULL, page = NULL, year = NULL, month = NULL,
    day = NULL, includemetadata = TRUE, results = NULL, ...)
```

The (**...**) at the end of ***ncdc()*** means there are many more optional parameters to change but we have no need to change the default values on any of these parameters.

The parameters we will be changing in ***ncdc()*** are (i.e., the parameters whose values we will change from the default value):

- ***datasetid***: the database we want the data from
- ***datatypeid***: the data we want from the database
- ***stationid***: the weather station we are getting data from
- ***startdate***: start date for the data
- ***enddate***: end date for the data
- ***token***: the 32 character token you got from NOAA/NCDC
- ***limit***: the maximum number of data points that will be returned to you

6.1 - Using your token

We have ***toolbox.r*** included in our script and ***toolbox.r*** has a variable named ***myToken*** that was assigned the 32 character token you got from NOAA/NCDC. So we can use ***myToken*** as the token:

```
token = myToken
```

6.2 - The limit parameter

The ***limit*** parameter is the maximum amount of data you will get. This is mostly to protect NOAA/NCDC from people accidentally (or maliciously) requesting millions of data points and crashing the server! The highest number you can put is 1000. You generally want to put in a number that is a little higher than the amount of data you expect to receive. We are getting 10 data points for every day for one month so we are getting around 300 data points. To be safe, we will set the limit to 400.

```
limit = 400
```

Note: If you do not receive all the data from NOAA/NCDC that you requested, there is a good chance that NOAA/NCDC cut off the data because there were too may data points. It is easy to miscalculate the number of data points, so it is always good idea to check the data you received for completeness.

6.3 - The start and end date parameters

Alright, lets get to the real data. ***startdate*** and ***enddate*** are just like they sound -- they give the date range for the data.

We want all of January 2016 so:

```
startdate = "2016-01-01", enddate = "2016-01-31"
```

note the YEAR-MONTH-DAY format for the dates.

6.4 - The datasetid parameter

datasetid *(data set id)* is the name of the database you want to get data from.

We are going to using the Global Historical Climatology Network-Daily database -- the id is ***GHCND***.

```
datasetid = "GHCND"
```

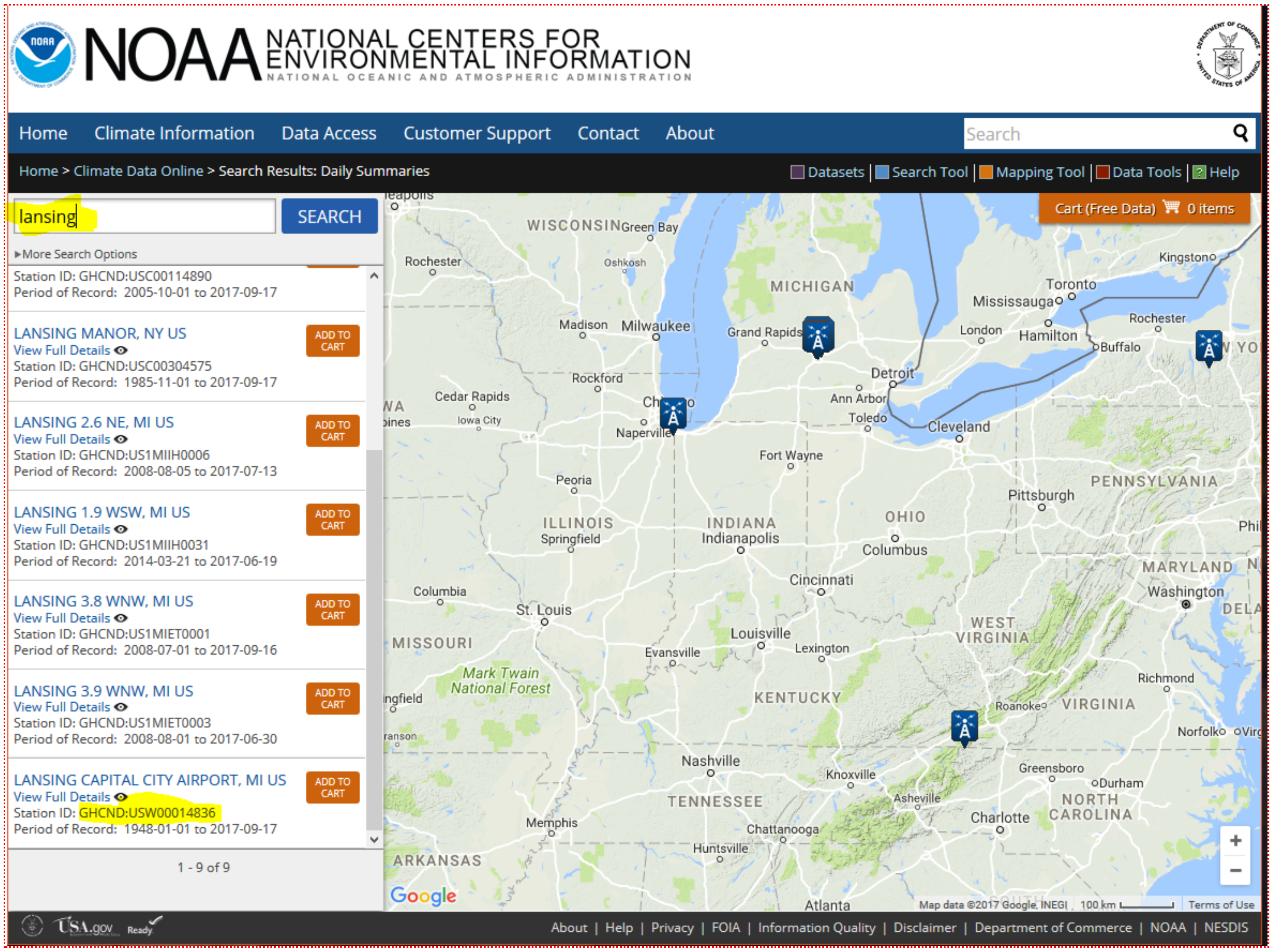
Extension: Finding the datasetid

6.5 - The stationid parameter

The ***stationid*** is a unique string for a weather station that is 17 characters long and starts with "GHCND:". You can find station ids on this page. Just type in a search parameter and click "Search".

For this lesson we will use the Lansing Capital City Airport station -- id: **GHCND:USW00014836**

```
stationid = "GHCND:USW00014836"
```


NOAA NATIONAL CENTERS FOR ENVIRONMENTAL INFORMATION
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
Home
Climate Information
Data Access
Customer Support
Contact
About
Search
Datasets
Search Tool
Mapping Tool
Data Tools
Help
SEARCH
More Search Options
Cart (Free Data) 0 items
ADD TO CART
1 - 9 of 9
About | Help | Privacy | FOIA | Information Quality | Disclaimer | Department of Commerce | NOAA | NESDIS

Home > Climate Data Online > Search Results: Daily Summaries

lansing

Station ID: GHCND:USC00114890
Period of Record: 2005-10-01 to 2017-09-17

LANSING MANOR, NY US
View Full Details
Station ID: GHCND:USC00304575
Period of Record: 1985-11-01 to 2017-09-17

LANSING 2.6 NE, MI US
View Full Details
Station ID: GHCND:US1MIIH0006
Period of Record: 2008-08-05 to 2017-07-13

LANSING 1.9 WSW, MI US
View Full Details
Station ID: GHCND:US1MIIH0031
Period of Record: 2014-03-21 to 2017-06-19

LANSING 3.8 WNW, MI US
View Full Details
Station ID: GHCND:US1MIET0001
Period of Record: 2008-07-01 to 2017-09-16

LANSING 3.9 WNW, MI US
View Full Details
Station ID: GHCND:US1MIET0003
Period of Record: 2008-08-01 to 2017-06-30

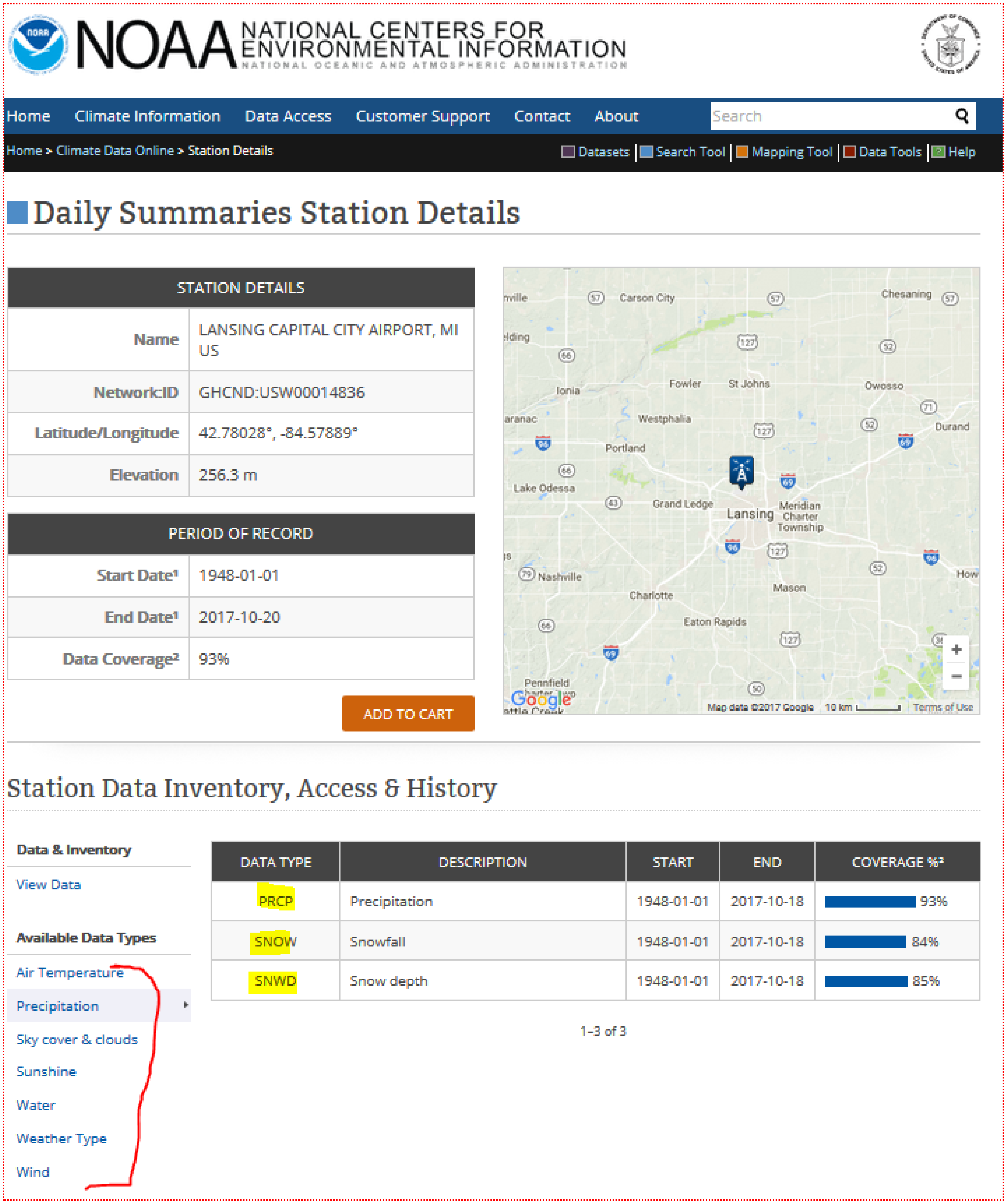
LANSING CAPITAL CITY AIRPORT, MI US
View Full Details
Station ID: GHCND:USW00014836
Period of Record: 1948-01-01 to 2017-09-17

Fig 7: Searching for a station id -- in this case we will use the one at Lansing Airport

6.6 - the datatypeid

Finally, we need to tell NOAA/NCDC what kind of data we want (e.g., temperature, rainfall, etc).

If you go to the Lansing Capital City Airport station page and scroll to the bottom, you will see a section called ***Available Data Type.*** Clicking on the categories produces a list of data types and their 4 digit code. It is these 4 digits codes that are used for the ***datatypeid***.


NOAA NATIONAL CENTERS FOR ENVIRONMENTAL INFORMATION
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

Home | Climate Information | Data Access | Customer Support | Contact | About | Search

Home > Climate Data Online > Station Details

Datasets | Search Tool | Mapping Tool | Data Tools | Help

Daily Summaries Station Details

STATION DETAILS	
Name	LANSING CAPITAL CITY AIRPORT, MI US
Network:ID	GHCND:USW00014836
Latitude/Longitude	42.78028°, -84.57889°
Elevation	256.3 m

PERIOD OF RECORD	
Start Date¹	1948-01-01
End Date¹	2017-10-20
Data Coverage²	93%

ADD TO CART

Station Data Inventory, Access & History

Data & Inventory
View Data

Available Data Types
Air Temperature
Precipitation
Sky cover & clouds
Sunshine
Water
Weather Type
Wind

DATA TYPE	DESCRIPTION	START	END	COVERAGE %²
PRCP	Precipitation	1948-01-01	2017-10-18	93%
SNOW	Snowfall	1948-01-01	2017-10-18	84%
SNWD	Snow depth	1948-01-01	2017-10-18	85%

1-3 of 3

*Fig 8: Location of the 4-digit **datatypeids**.*

For this example, we will use the following codes:
TMAX, TAVG, TMIN, PRCP, SNOW, AWND, WDF2, WSF2, WT01, WT09

We need to put these IDs in a vector to assign them to the parameter ***datatypeid***

```r
datatypeid=c("TMAX", "TAVG", "TMIN", "PRCP", "SNOW", "AWND",
             "WDF2", "WSF2", "WT01", "WT09")
```

Note: if you are really bored and want to know everything about the data contained with the GHCND data set, you can consult this document.

6.7 - Putting the function together

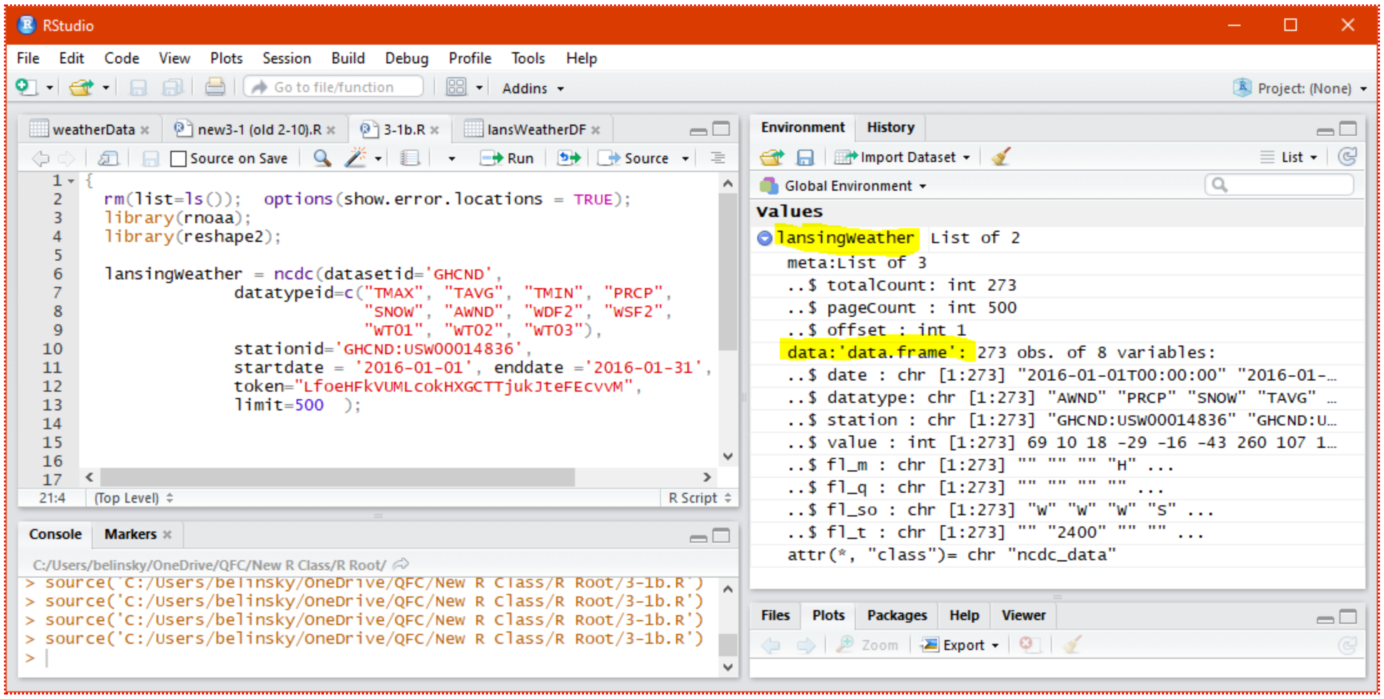
Here is the full function call to NOAA/NCDC with all the parameters. I assign the return value from ***ncdc()*** to the variable ***lansingWeather***

```r
{
  rm(list=ls());  options(show.error.locations = TRUE);

  source("scripts/toolbox.r");
  library("rnoaa");    # include functions from the rnoaa package
  library("reshape2"); # include functions from the reshape2 package

  lansingWeather =  ncdc(datasetid='GHCND',
                         datatypeid=c("TMAX", "TAVG", "TMIN", "PRCP",
                                      "SNOW", "AWND", "WDF2", "WSF2",
                                      "WT01", "WT02", "WT03"),
                         stationid = 'GHCND:USW00014836',
                         startdate = '2016-01-01', enddate ='2016-01-31',
                         token = myToken,
                         limit = 400);
}
```

ncdc() lansingWeather is a list. You can think of lists as folders that contain objects like vectors, data frames, matrices, or even other lists (e.g., folder inside a folder). In this case, the list ***weatherData*** contains two objects: ***meta****,* another list, and ***data***, a data frame. ***data*** contains the weather information we requested.



```
{
  rm(list=ls());  options(show.error.locations = TRUE);
  library(rnoaa);
  library(reshape2);

  lansingWeather = ncdc(datasetid='GHCND',
                  datatypeid=c("TMAX", "TAVG", "TMIN", "PRCP",
                               "SNOW", "AWND", "WDF2", "WSF2",
                               "WT01", "WT02", "WT03"),
                  stationid='GHCND:USW00014836',
                  startdate = '2016-01-01', enddate ='2016-01-31',
                  token="LfoeHFkVUMLcokHXGCTTjukJteFEcvvM",
                  limit=500  );
```

```
Values
lansingWeather  List of 2
  meta:List of 3
  ..$ totalCount: int 273
  ..$ pageCount : int 500
  ..$ offset : int 1
  data:'data.frame': 273 obs. of 8 variables:
  ..$ date : chr [1:273] "2016-01-01T00:00:00" "2016-01-…
  ..$ datatype: chr [1:273] "AWND" "PRCP" "SNOW" "TAVG" …
  ..$ station : chr [1:273] "GHCND:USW00014836" "GHCND:U…
  ..$ value : int [1:273] 69 10 18 -29 -16 -43 260 107 1…
  ..$ fl_m : chr [1:273] "" "" "" "H" ...
  ..$ fl_q : chr [1:273] "" "" "" "" ...
  ..$ fl_so : chr [1:273] "w" "w" "w" "s" ...
  ..$ fl_t : chr [1:273] "" "2400" "" "" ...
  attr(*, "class")= chr "ncdc_data"
```

```
C:/Users/belinsky/OneDrive/QFC/New R Class/R Root/
> source('C:/Users/belinsky/OneDrive/QFC/New R Class/R Root/3-1b.R')
> source('C:/Users/belinsky/OneDrive/QFC/New R Class/R Root/3-1b.R')
> source('C:/Users/belinsky/OneDrive/QFC/New R Class/R Root/3-1b.R')
> source('C:/Users/belinsky/OneDrive/QFC/New R Class/R Root/3-1b.R')
> |
```

*Fig 9: **list** object return from the function call to the NOAA/NCDC database.*

Note: We are making requests to the NOAA/NCDC servers and these severs are not the fastest, so it might take some time for your script to run.

6.8 - Subsetting a list

We want to extract the data frame ***data*** from the **list** object named ***lansingWeather***. Essentially we want to pull an object out of ***lansingWeather***. To pull an object out of a list, we use double brackets [[ ]]. Note: to subset a data frame we used single brackets [ ].

The following line gets the data frame named ***data*** from the ***lansingWeather*** list and saves it to the variable ***lansingWeatherDF***.

```
lansingWeatherDF = lansingWeather[["data"]];
```

7 - Application

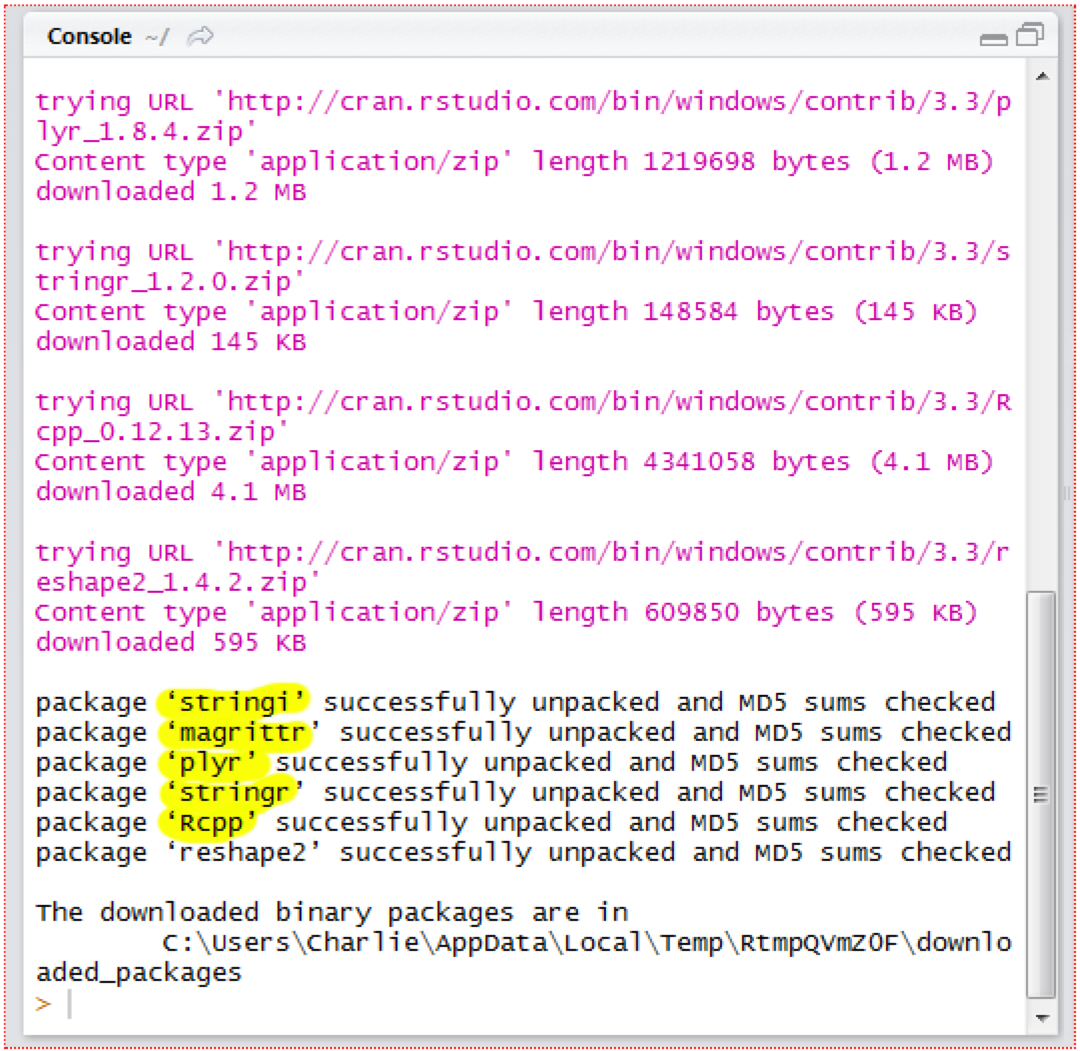
If you have any questions regarding your Class Project or this application, feel free to email them to the instructor here. You can attach the whole Root Folder as a zipped file.

1. Get the following data for December 2016 through February 2017 in Lansing, MI: **PRCP, SNOW, TAVG, TMAX, TMIN**
2. Save the data frame (***data***) from the list to a variable named ***lansingWinterWeather***

*Save you script file as **app 3-1.r** in the **scripts** folder of your RStudio Project for the class.*

8 - Extension: Dependencies

A package is just somebody else's script that you are including in your own script file. These packages are almost always dependent upon other scripts, or packages. These are its *dependencies* and the dependencies get installed with the package. That is why you might see something like this when installing a package:



```
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.3/p
lyr_1.8.4.zip'
Content type 'application/zip' length 1219698 bytes (1.2 MB)
downloaded 1.2 MB

trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.3/s
tringr_1.2.0.zip'
Content type 'application/zip' length 148584 bytes (145 KB)
downloaded 145 KB

trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.3/R
cpp_0.12.13.zip'
Content type 'application/zip' length 4341058 bytes (4.1 MB)
downloaded 4.1 MB

trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.3/r
eshape2_1.4.2.zip'
Content type 'application/zip' length 609850 bytes (595 KB)
downloaded 595 KB

package 'stringi' successfully unpacked and MD5 sums checked
package 'magrittr' successfully unpacked and MD5 sums checked
package 'plyr' successfully unpacked and MD5 sums checked
package 'stringr' successfully unpacked and MD5 sums checked
package 'Rcpp' successfully unpacked and MD5 sums checked
package 'reshape2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Charlie\AppData\Local\Temp\RtmpQVmZOF\downlo
aded_packages
> 
```

*Fig 10: Installing **reshape2** package, which has 5 package dependencies.*

Reshape2 is dependent upon 5 other packages and RStudio has installed those packages for you, too (because you had "Install Dependencies" checked).

9 - Extension: Finding the datasetid

Unfortunately, the help pages for the NOAA/NCDC databases are quite lacking, so finding what the names of the databases (i.e., the ***datasetid***) is not elementary.

This webpage presents the different data sets but does not give you the ids. I still have not found good documentation about this, however, ***GHCND*** is probably the most common ***datasetid***.