

...Fortunately, I've had the chance to get to know two particularly impressive R packages which are available to the scientific community through CRAN. Together, the two packages can be used to get fast, reliable information on the climate conditions and solar irradiation profiles from specific locations. After quickly introducing each, I show below one quick way to combine them in a short script.

1. The first of these is [hddtools](#), developed/maintained by Claudia Vitolo, a hydrologist and data scientist. Hddtools, or "Hydrological Data Discovery Tools", is quite versatile and powerful but I was most interested in using it to find the KGC Classification of specific points on the globe defined by their latitude/longitude. In a word, the KGC Classification Scheme allows any point on the globe to be defined according to a three-character code which in conjunction fully describe that point's climate: one defines its overall band in terms of latitude; the second defining the pattern of precipitation, for example if most rainfall in a given location occurs during a specific season (like in a monsoon); and the third level categorises a given location by whether its most defining weather was during a hot summer or a cold winter. Retrieving information from this app is straightforward, as the following snippet which retrieves the KGC classification for Qatar shows:

```
qatarboundingbox <- raster::extent(50.75, 51.25, 25, 26)
kgcclassqatar <- hddtools::KGClimateClass(qatarboundingbox)

#The output looks like this
  ID Class Frequency
1 26   Dfb         50
```

The bounding box we define takes arguments of: minimum/maximum longitude and then minimum/maximum latitude, expressed in decimal units. The output is saved as a data frame which holds the classification(s) for the bounding box we just queried. Qatar is a pretty straightforward place, but in a later example I will work with areas which have a diverse climate.

2. The second funky package I want to talk about is ["nasapower"](#), maintained/developed by Adam Sparks. This package has a more straightforward use, it is simply a wrapper for the API requests to the NASA climatology data which is already made publicly available. Of course, you can equally write your own API requests but if, like me, you find dealing with the API too cumbersome, this is one great package. Compared the hddtools above, we only need to rearrange the order of latitude and longitude points a little but the rest is similarly straightforward.

```
#The below example queries the NASA API for all sky irradiation
# data for a flat solar plate
# (see "pars" argument)
#The units are adjusted for the sustainable
#energy community (kWh/m^2*day) and we are using an inter-annual
#average

nasaresponse <- nasapower::get_power(
  community = "SSE",
  lonlat = c(qatarboundingbox[1], qatarboundingbox[3],
             qatarboundingbox[2], qatarboundingbox[4]),
  dates = c("1999", "2000"),
  temporal_average = "INTERANNUAL",
  pars = ("ALLSKY_SFC_SW_DWN")
)
```

Again, all we had to was rearrange the order of longitude and latitude coordinates fed into

the API.

Finally, the code snippet below will show one quick and easy way to apply both of these methods to an arbitrary number of locations on the map. Of course, the bounding box's size (measured in degrees longitude/latitude) will be a determining factor in the coarseness of the data gained, but one of the beautiful things about the hddtools package in particular is that it can be used with few restrictions on the longitude/latitude definition (not the case for all such packages). Anyway, below, we open the spreadsheet file which has rows for a variety of locations, with the bounding box being defined by columns of data for each entry. We then use the combination of hddtools and nasapower to re-write the spreadsheet with two additional values: one for the defining climate classification and one for the annual average solar irradiation.

```
#Save the spreadsheet to a dataframe
datalocations <- gdata::read.xls("locations.xls", sheet = "locationsdata",
header = TRUE, stringsAsFactors = FALSE)

#Make sure spreadsheet has right column names

for(i in 1:nrow(datalocations))
{ bindbox <- raster::extent(datalocations$minlon[i],
  datalocations$maxlon[i], datalocations$minlat[i], datalocations$maxlat[i])

#get KGC for each row
kgctemp <- hddtools::KGClimateClass(bindbox)

allowedclimates %lt;- c("Af", "Am", "As", "Aw", "Aw/As", "BWh",
  "BWh", "BSk", "BSh", "Cfa", "Cfb", "Cfc", "Csa", "Csb",
"Csc", "Cwa", "Cwb", "Cwc", "Dfa", "Dfb", "Dfc", "Dfd", "Dsa", "Dsb", "Dsc",
"Dsd", "Dwa", "Dwb", "Dwc", "Dwd", "EF", "ET")

levels(datalocations$V6) <- allowedclimates

dominantclimate <- max(kgctemp[,3]))

#for complex climates, define the dominant one
for(j in 1:nrow(kgctemp))
{
  if(kgctemp[j,3] == dominantclimate)
    datalocations[i, 6] <- as.character
}

#Now for solar irradiation
solardata %lt;- nasapower::get_power(
community = "SSE",
  lonlat = c(qatarboundbox[1], qatarboundbox[3],
    qatarboundbox[2], qatarboundbox[4]),
  dates = c("1999", "2000"), he
  temporal_average = "INTERANNUAL",
  pars = ("ALLSKY_SFC_SW_DWN")
)

datalocations[i, 7] <- mean(solardata$ANN)

}
```