

Dataquest Guided Project: New York Solar Resource Data

Cindy Zhang

1/22/2021

Contents

Introduction	1
Findings	1
API set up	1
Create dataframe from complex list	3

Introduction

This is my solution to Dataquest's Guided Project from the APIs in R course, which practices querying APIs in R.

More details such as the RMD and csv files can be found in the repository in GitHub.

Findings

API set up

This guided project involves querying the Data Gov API, which requires an access key. Enter your own data key; if you don't have one, you can sign up for one [here] (<https://api.data.gov/signup/>).

First, I stored my endpoint of interest and parameters in a variable and a list, respectively. Then I wrote a function to query the Solar data API and executed the function, storing the results.

```
solar_url <- "/api/solar/solar_resource/v1"
parameters_list <- (list(format = "json", api_key = cz_key, lat = "41", lon = "-75"))

#api get function
gov_api_json <- function(endpoint, queries = list()) {

  # Preparing the URL
  url <- modify_url("https://developer.nrel.gov", path = endpoint)

  # API requests
  response <- GET(url, query = queries)
```

```

# Tracking errors
if ( http_error(response) ){
  print(status_code(response))
  stop("Something went wrong.", call. = FALSE)
}

if (http_type(response) != "application/json") {
  stop("API did not return json", call. = FALSE)
}

# Extracting content
json_text <- content(response, "text")

# Store content as list
json_lists <- jsonlite::fromJSON(json_text)

# Return list
json_lists
}

# Execute function

complex_list <- gov_api_json(solar_url, queries = parameters_list)
str(complex_list)

```

```

## List of 6
## $ version : chr "1.0.0"
## $ warnings: list()
## $ errors : list()
## $ metadata:List of 1
## ..$ sources: chr "Perez-SUNY/NREL, 2012"
## $ inputs :List of 2
## ..$ lat: chr "41"
## ..$ lon: chr "-75"
## $ outputs :List of 3
## ..$ avg_dni :List of 2
## ...$ annual : num 3.69
## ...$ monthly:List of 12
## ...$ jan: num 3.12
## ...$ feb: num 3.36
## ...$ mar: num 4.1
## ...$ apr: num 4.07
## ...$ may: num 4.15
## ...$ jun: num 4.17
## ...$ jul: num 4.6
## ...$ aug: num 4.14
## ...$ sep: num 4.02
## ...$ oct: num 3.26
## ...$ nov: num 2.58
## ...$ dec: num 2.72
## ..$ avg_ghi :List of 2
## ...$ annual : num 3.87
## ...$ monthly:List of 12

```

```
## .. .. ..$ jan: num 1.97
## .. .. ..$ feb: num 2.69
## .. .. ..$ mar: num 3.86
## .. .. ..$ apr: num 4.7
## .. .. ..$ may: num 5.45
## .. .. ..$ jun: num 5.78
## .. .. ..$ jul: num 5.98
## .. .. ..$ aug: num 5.14
## .. .. ..$ sep: num 4.23
## .. .. ..$ oct: num 2.94
## .. .. ..$ nov: num 1.99
## .. .. ..$ dec: num 1.67
## ..$ avg_lat_tilt:List of 2
## .. ..$ annual : num 4.52
## .. ..$ monthly:List of 12
## .. .. ..$ jan: num 3.55
## .. .. ..$ feb: num 4.04
## .. .. ..$ mar: num 4.86
## .. .. ..$ apr: num 4.97
## .. .. ..$ may: num 5.18
## .. .. ..$ jun: num 5.24
## .. .. ..$ jul: num 5.58
## .. .. ..$ aug: num 5.24
## .. .. ..$ sep: num 5
## .. .. ..$ oct: num 4.11
## .. .. ..$ nov: num 3.26
## .. .. ..$ dec: num 3.13
```

Create dataframe from complex list

Since `complex_list` is a bunch of nested lists, I converted it into a work-able dataframe. I had to manually re-add the month columns.

```
# simplify list- avg_dni
u_complex_list <- unlist(complex_list$outputs)

# restructure the new list into a matrix
u_df <- as.data.frame(matrix(u_complex_list, nrow=13))

# remove first row
u_df <- u_df[-1,]

# rename columns
u_df <- u_df %>%
  rename(
    avg_dni = V1,
    avg_ghi = V2,
    avg_lat_tilt = V3
  )

# add month column
u_df <- u_df %>%
  mutate(month = month.abb) %>%
```

```
select(month, everything())
```