ACMT Example: Using external data

Weipeng Zhou

2/22/2021

Introduction

There might be times you find context measurement variables you are interested but not available in ACMT. In that scenario, you can configure ACMT to include variables in external data.

Here we show how to configure ACMT to include variables in a dataset not originally available in ACMT – Modified Retail Food Environment Index (https://www.cdc.gov/obesity/resources/reports.html)

We take the following steps:

1. Write a function for downloading Modified Retail Food Environment Index (mRFEI) in ACMT
2. Write a function for processing the downloaded data into the ACMT-friendly format
3. Install the file downloading and processing functions into ACMT
4. Configure ACMT to let it include variables in mRFEI

Example

1. Write a function for downloading mRFEI in ACMT

Requirement:

- The downloaded files should be stored in `workspace/external_data/`.

Take notes of the name of the downloaded file (`downloaded_mrfei.xls`), it will be used for configuring ACMT later.

```
download_file_mrefi <- function () {
  download.file(url = "https://www.cdc.gov/obesity/downloads/2_16_mrfei_data_table.xls", destfile = "ex
}
```

2. Write a function for processing the downloaded data into the ACMT-friendly format

Requirements:

- The content of the processed file (`processed_mrfei.csv`) should match the following format.

`GEOID` is the unique identifier for a geographical area; `variable` is the name of the variable; `estimate` is the value of the variable for the corresponding to the `GEOID` and `variable`.

```
GEOID       variable   estimate
53033000100 population     6282
53033000100 population     2914
09011650100 commute_time   2910
```

```
09011650100 population     9913
...         ...            ...
```

- The processed file should be stored in `workspace/external_file/`.
- The processed file should be named as `processed_<dataset_name>.csv`, where `<dataset_name>` should be replaced by the `external_data_name` (first level name) in `external_data_name_to_info_list` (See 3. for detail).

In this example, `external_data_name` is `mrfei` and the processed file should be named as `processed_mrfei.csv`.

```
process_file_mrefi <- function () {  # process the file and name it processed_mrfei.csv (mrfei can chan
  library(readxl)
  raw_mrfei_dataframe <- read_excel("external_data/downloaded_mrfei.xls")

  ## select the GEOID and estimate columns (rename with proper names), and insert the varaible column
  processed_dataframe <- raw_mrfei_dataframe %>%
          dplyr::select(fips, mrfei) %>%
          rename(GEOID=fips, estimate=mrfei) %>%
          add_column(variable="mRFEI", .after = "GEOID")

  processed_dataframe$estimate[is.na(processed_dataframe$estimate)] <- 0  # impute NA with 0

  write_csv(processed_dataframe, "external_data/processed_mrfei.csv")
}
```

3. Install the file downloading and processing functions into ACMT

Put the file downloading and proessing functions into `workspace/external_data-file_downloader_and_processor.R`.

4. Configure ACMT to let it include variables of mRFEI

When calling ACMT, construct a list that contains information about the external dataset and call ACMT with the list.

Below shows an example of the list.

```
external_data_name_to_info_list <- list(
        mrfei=list(vector_of_expected_downloaded_file_name=c("downloaded_mrfei.xls"),  # files that sho
                   download_file=download_file_mrefi,  # function to download file
                   process_file=process_file_mrefi,   # function to process file
                   geoid_type="Census Tract",
                   variable_name_to_interpolate_by_sum_boolean_mapping=c(mRFEI=TRUE)
        )
)
```

Requirements:

- The first level name of the list (`mrfei` in this case) should make up the `<dataset_name>` part of the processed file (`processed_mrfei.csv` in this case). More specifically, `paste0("processed_", names(external_data_name_to_info_list)[0], ".csv") == 'processed_<dataset_name>.csv'`.
- `vector_of_expected_downloaded_file_name` should be the names of the files downloaded by the download function (for checking if downloading succeeds).

- `geoid_type` should be carefully chosen by the user. There are various levels of GEOID and ACMT currently only supports two, "Census Tract" and "Block Group". For example, mRFEI is at "Census Tract" level while National Walkability Index is at "Block Group" level.
- `variable_name_to_interpolate_by_sum_boolean_mapping` is a named logical vector specifying the mapping between an external variable and its interpolation type. For example, if we have a variable `population of an area`, according to the definition of `tigris`, is an extensive variable and we should set its interpolation type to sum; in contrast, the variable `population density of an area` is not an extensive variable and we should set its interpolation type to mean.

Results

When the 4 steps are done, we can run ACMT with external data.

```
source("setup-acmt.R")
external_data_name_to_info_list <- list(
        mrfei=list(vector_of_expected_downloaded_file_name=c("downloaded_mrfei.xls"),
                   download_file=download_file_mrefi,
                   process_file=process_file_mrefi,
                   geoid_type="Census Tract",
                   variable_name_to_interpolate_by_sum_boolean_mapping=c(mRFEI=TRUE)
        )
)
context_measurements <- get_acmt_standard_array(long=-122.333, lat=47.663, radius_meters = 200, year=20
```

```
print(filter(context_measurements, context_measurements$names == "mRFEI"))
```

```
##       names   values
## mrfei mRFEI 1.303625
```

FAQ

I hate downloading and unzipping files in ACMT; this runs very slow; what should I do?

You are right. ACMT runs in Docker which is a virtual machine, and it runs slower than your local computer.

When the dataset is large, we recommend downloading and processing the file in your local computer, and then pull the processed file it into ACMT (into `workspace/external_data/`).

If you have done that, you can even skip writing the file downloading and processing function.

And configure ACMT like this (GEOID type is still mandatory).

```
external_data_name_to_info_list <- list(
  mrfei=list(vector_of_expected_downloaded_file_name=NULL,  # the files should be downloaded for mrfei
             download_file=NULL,  # function to download file
             process_file=NULL,   # function to process file
             geoid_type="Census Tract",
             variable_name_to_interpolate_by_sum_boolean_mapping=mrfei_variable_name_to_interpolate_by_
  )
)
```

I do not like writing R functions to make ACMT use external data, what should I do?

Good decision. Do file processing outside (using your favorite programming language) and pull it into ACMT.

Just make sure the processed file fits the format described in Step 2.

ACMT outputs NA's for some external variables, is ACMT broken?

Please let me know if this occurs.

But assuming ACMT is working correctly, there are two possibilities.

The first possibility is that the processed file contains NA entries. This could be fixed by doing imputation before pulling it into ACMT.

The second possibility is that for the targeted geographic coordinate and radius, the external does not contain census tracts/block groups for certain areas. For example, in mRFEI data, there are no records (not even GEOID) for some census tracts.

The mRFEI and National Walkability Index datasets sound interesting to me. I don't usually do this but could you set them up for me?

Yes, we have done it! ACMT comes with the processed files for both datasets. Use the below `external_data_name_to_info_list` to let ACMT use the datasets.

```
external_data_name_to_info_list <- list(
  mrfei=list(vector_of_expected_downloaded_file_name=NULL,
             download_file=NULL,
             process_file=NULL,
             geoid_type="Census Tract",
             variable_name_to_interpolate_by_sum_boolean_mapping=mrfei_variable_name_to_interpolate_by_s
  ),
  walkability=list(vector_of_expected_downloaded_file_name=NULL,
             download_file=NULL,
             process_file=NULL,
             geoid_type="Block Group",
             variable_name_to_interpolate_by_sum_boolean_mapping=walkability_variable_name_to_interpola
  )
)
```