# Downstream processing of submissions

These GAMS programs are run after the initial verification that is done by Python

## ReadCSV.gms

Purpose: Read all of the submissions, merge into one database, and save as a GDX file.

Requires: AgMIPSets.gms—this file contains all of the valid set labels for each of the fields:

Mod Model name

Scen Scenario name

R Region

V Indicator

I Sector or items

T Year

U Unit

For each field, the AgMIPSets.gms file has the standard protocol labels, and then labels specific to individual submissions (one or more).

Input: The individual model submissions. N.B. There could be more than one submission per modeling team.

Output: The full data cube in GDX format.

## Extract.gms

Purpose: Reads the GDX-based database created in the first step and dumps it to a CSV cube applying filters to regions and indicators. (Other filters could/should be included.)

Requires: AgMIPSets.gms—this file contains all of the valid set labels for each of the fields.

Input: The full data cube in GDX format.

Output: The filtered data cube in CSV format.

## GetSets.gms

Purpose: Reads the GDX-based database created in the first step and gets all of the unique set labels. These are saved to CSV files—one for each field. The file ModStats.xlsx loads the individual CSV files and creates a table with the statistics of use of each label by model. This could readily be done with the initial Python code.

Requires: AgMIPSets.gms—this file contains all of the valid set labels for each of the fields.

Input: The full data cube in GDX format.

Output: The unique labels for each field in separate CSV files.