UM | Centre for Earth Observation Science

CanWIN AVOS Codebook

CENTRE FOR EARTH OBSERVATION SCIENCE

Document Control

0.1 Version History

Version	Author(s)	Type	Date Modified	Comments
1.0	Iyakoregha, Campbell, Y.	Old version	2020/05/24	Outdated template.
2.0	Friesen, K. L.	Working Copy	2022/01/13	Updated template.

0.2 Document Location

A digital copy of this document can be found on GitLab and in the CanWIN datahub.

0.3 License

Contents

1 Introduction

1.1 Description

This codebook describes the **AVOS_parse_v1.2.R** and **AVOS_parse_v1.3.R** script process which converts GPS data and meteorological data, collected from the Automatic Voluntary Observing Ship (AVOS) system onboard the NAMAO, to a readable file format (.csv). This system collects data continuously at 10 minute intervals, at discrete sample stations, every day throughout the duration of the research cruise. Read the *AVOS_Cookbook* for additional information on this dataset.

Variables

The data extracted from the AVOS system on the NAMAO is at Data Level 0, raw data.

Table 1.1: Variables in .AVMTD data files

Number	**Variable abbreviation**
V1	AVMTD
V2	DATE
V3	TIME
V5	lat
V6	lon
V7	ws
V8	wd_rel
V9	wd_true
V13	p_uncor
V15	Ta
V16	RH
V19	Tsrfc
V20	BV
V23	D...D
V24	Ship's heading (AVOS magnetic)
V25	sog_kts
V33	p_cor
V39	Ship's heading (Gyro compass)

Note: Row names V4, V10-12, V14, V17-18, V21-22, V26-32, V34-38, and V40 are unknown variables and are not relevant to this analysis.

1.2 Data Processing

This script was originally written from IDL program **AVOS_parse2020.R** by T. Papakyriakou. GPS coordinates are converted to decimal degrees in the IDL program and had problems handling missing data. Thus, these issues were corrected in this **AVOS_parse_v1.2** R script, and a third version (1.3) was created to standardize the variable names.

The R script takes the AVOS data in its original format (.AVMTD), converts the GPS coordinates to decimal degrees and saves this, along with the other variables, in a .csv file format.

Variables

The final conversion output will result in a.csv file at Data Level 1.0, first pass quality control. The variables included in this file are the following.

Table 1.2: Variables in converted .csv files

Column number	**Variable**
1	year
2	mon
3	day
4	hour
5	lat
6	lon
7	ws
8	wd_rel
9	wd_true
10	Ta
11	RH
12	Tsrfc
13	sog_kts
14	p_cor
15	heading

In the third version of the script (v1.3) the variables are standardized and thus become Data Level 1.1.

Table 1.4: Standardized variables in converted .csv files

Column number	**Variable**
1	Date_and_Time
2	latitude
3	longitude
4	wind_speed
5	RelWindDirFrom
6	WinDirFrom
7	air_pressure
8	air_temperature
9	relative_humidity
10	Temp
11	platform_speed_wrt_ground
12	air_pressure_at_mean_sea_level
13	platform_orientation

2 Setup

You will need to install R and RStudio to run the **AVOS_parse_v1.2.R** script. If installation instructions are needed, please reference the CanWIN Orientation Manual on GitLab for R, RStudio, and other software tools.

You can find these scripts in the pCO2 AOA data repository.

Before running this script, you will need to have structured directories for your AVOS data, which produces a folder. Use the structure below as an example:

- 2017
  - 01
    - 20170101.csv or 20170101.AVMTD ...
    - 20171231.csv or 20171231.AVMTD
  - 02
  - 03
  - 04
  - 05
  - 06
  - 07
  - 08
  - 09
  - 10
  - 11
  - 12

The converted AVOS files (.csv) are then placed in their corresponding month and year, then the file names listing the dates for that given month and year will populate the directory once the script has run succesfully, as shown above. The file structure is then repeated for all the months of that year (you require) as well as repeated for the years (you require). Additionally, you should organize the raw (.AVMTD) file in a similar fashion, as indicated above.

2.1 Code Specifics

Required packages for this script are coded into the script. Hence, if you do not have them it will install the packages automatically.

The following packages are installed when running the script:

1. **svDialogs** is a package that allows you to create quick standard dialog boxes for Windows, MacOS, and Linux operating systems.

2.1.1 Keyword Definition

There are no overarching keywords in this script.

Functions

This script does not have any main functions. This script is enclosed by curly braces with dialog boxes integrated within the code to help guide the user. The main operator in this script is a for-loop that cycles through the .AVMTD files.

3 Script Description

Script name(s): AVOS_parse_v1.2.R and AVOS_parse_v1.3.R

These files are located on Gitlab and the CanWIN datahub.

3.1 Inputing Working Directories

1. If you do not have the needed R packages installed, the script will warn you and proceed to install them for you.
2. Load svDialogs library.
3. Dialog box appears requesting you to select the directory or folder where the raw .AVMTD files are stored.
4. Directory pathway is then stored in `input_path`.
5. If no file names are in the script the script will stop.
6. Dialog box appears requesting you to select the directory or folder where to store the processed .csv files.
7. Directory pathway is then stored in `output_path`.
8. If no file names are in the script the script will stop.
9. Retrieve list of .AVMTD file names and store them in `file.names`.
10. If no file names are in the script the script will stop.

3.2 Main For-Loop

The main for-loop standardizes and subsets a portion of the variables in the .AVMTD file and renames column variables so they are readable by humans.

1. The main for-loop starts by creating an index `i` to iterate through the length of the `file.names`, starting with the first name in the list when `i = 1`.
2. A `message()` function is used to print the current file name being processed in the console and assists with tracking issues if an error occurs in the script.
3. The first .AVMTD file is read as a dataframe in .csv format and saved as `df`.

4. `df` is then filtered, keeping only the rows in the first column (V1) of the dataset that are “AVMTD” (AVOS) data.
5. The date is saved in separate variables, `year` (year), `mon` (month), `day` (day), which are subsetted using the `substr()` function and applying this to the second column (V2) of the file.
6. The time is saved in separate variables, `hour`(hour) and `minute` (minutes), which are subsetted using the `substr()` function and applying this to the third column (V3) of the file.
7. Convert latitude in the format “DDmm.mmmmn” where “n” = “N” or “S” (north or south).
8. Save latitude degree "DD" in variable `latdeg`
9. Save latitude minutes "mm.mmmm" in variable `latmin` and converted to decimal minutes (dividing by 60).
10. Hemisphere letter, "N" or "S", of latitude measurement is saved in the variable `NS_Hemisphere`.
11. Latitude is then converted into decimal degrees by addition of `latdeg` and `latmin` and saved in the new variable `lat`.
12. The index of latitude measurements that had an "S" in `NS_Hemisphere` using the `which()` function and are saved in the new variable `posn_where_S`.
13. The positions are then used to add a negative symbol to `lat` by using square brackets (R index method).
14. The same process is repeated for longitude, where it is converted to decimal degrees by adding `londeg` and `longmin` (minutes divided by 60 to get decimal mintues) which is saved into the new variable `lon`.
15. Sign of measurement in the Western Hemisphere are then changed to negative, by identifying positions of measurements in column with "W" in `EW_Hemisphere` using the `which()` function and R indexing method.
16. Remaining variables of interest (10) from the .AVMTD file are saved in new variables that cover more meaning.

3.3 Standardizing Variable Names

This portion of the script only applies to version 1.3 of the **AVOS_parse** script.

1. The date (`year`, `mon`, and `day`) and time (`hour` and `minutes`) variables are formatted together by ISO standards (YYYY-MM-DD hh:mm) using the `psate0()` function and saved in a new variable `UT_ISO8601`.
2. The remaining variables listed in Table 1.2 are then saved as the standardized variable names shown in Table 1.3.

3.4 Exporting Converted Files

1. All variables (non-standardized or standardized) are saved into a new dataframe `output_df`.
2. The `output_filename` saves the new dataframe name by using the raw file, removing the original extension (.AVMTD), and replacing it with .csv.
3. The `write.csv()` saves the `output_df` using the `output_path` and new file name.
4. A printed `message()` appears in the R console window to indicate the end of the file processing script and shows the output directory path where the .csv files have been located.
5. The code then exits the final curly brace that ends the script.

4 Execution Guidelines

4.1 Initializer File

There is no initialization file for you to input your directory pathways and it does not require you to set your working directory to a specific location. This is taken care of via the svDialogs package, which was used in created dialog boxes to prompt users to input the directory pathway to raw AVOS files, and the directory pathway to where they would like the files to be saved. Please, have the directories structured as indicated in Chapter 2.

4.2 Running Script

To run the script all you need to do is place the cursor outside below the text description and above the first curly brace in the code. Then you simply press "Run" on the right side of the ribbon on top of the script window.

This script has no descrepancies for any operating system. It should run successfully, under a minute, on all platforms.

5 Log

Otherwise, you can provide additional information here for running your script. For more information how to run an R script click here for a short resource on execution or visit Data Carpentry to start understanding basic RStudio features.

A Glossary of Options and Packages

A.1 R Packages

Visit https://cran.r-project.org/web/packages/available_packages_by_name.html to learn more about R packages

- **Package 1** - Description
- **Package 2** - Description

A.2 Python

A.2.1 Python Script-Specific Options

- **Option 1** - Description
- **Option 2** - Description

A.2.2 Python Packages

Visit https://docs.python.org/3/library/ to learn more about python packages

- **Package 1** - Description
- **Package 2** - Description

Example: Section 2.1 from Victory's semi-hemi codebook