Validation Report

Adamski (Version 0.0.5)

Ryo Nakaya

25DEC2025

General Information

Adamski is a SAS package inspired by the R package {admiral}. It aims to bring the same flexible and modular ADaM derivation framework to the SAS environment. The package follows the {admiral} design principles while adapting to SAS syntax and workflows. It enables consistent, reproducible ADaM dataset creation in compliance with CDISC standards. Adamski serves as a bridge between open-source R implementations and traditional SAS programming.

Validation Environment

OS: WIN

SAS: 9.04.01M7P080520

Required Packages: -

Execution Datetime: 25DEC2025:13:55:21

Authors

[Yutaka Morioka],[Hiroki Yamanobe],[Ryo Nakaya],[Sharad Shhetri]

Requirements

- %derive_vars_dy :
Generates study day (DY) variables from given date variables using a specified reference date, for example TRTSDT.
- %derive_var_merged_exist_flag :
Creates a character flag variable indicating whether the current DATA step row`s keys exist in another dataset.
- %derive_var_age_years :
Converts a set of age values from the specified time unit to years.
- %derive_vars_duration :
Derives duration between two dates, specified by the variables present in the input dataset, for example duration of adverse events, relative day, age, etc..
- %derive_locf_records :
Adds LOCF records (Last Observation Carried Forward) to a dataset based on an `expected observations` reference dataset.

Validation Records

Test Description	Result	Comments
(%derive_locf_records)[test01] Compare expected and test results for case of imputation=add	PASS	MP_ASSERTDATASET: proc compare base=out_expected compare=out_test
(%derive_locf_records)[test02] Compare expected and test results for case of imputation=	PASS	MP_ASSERTDATASET: proc compare base=out_expected compare=out_test
(%derive_locf_records)[test03] Compare expected and test results for case with DTYPE of LOG	PASS	MP_ASSERTDATASET: proc compare base=out_expected compare=out_test
(%derive_locf_records)[test04] Compare expected and test results for case of imputation=update	PASS	MP_ASSERTDATASET: proc compare base=out_expected compare=out_test
(%derive_locf_records)[test05] Compare expected and test results for case of imputation=update_add	PASS	MP_ASSERTDATASET: proc compare base=out_expected compare=out_test

Test Description	Result	Comments
(%derive_locf_records)[test06] Compare expected and test results for case with ADY and id_vars_ref	PASS	MP_ASSERTDATASET: proc compare base=out_expected compare=out_test
(%derive_locf_records)[test07] Compare expected and test results for case with multiple ADY in AVISIT	PASS	MP_ASSERTDATASET: proc compare base=out_expected compare=out_test
(%derive_locf_records)[test08] Compare expected and test results for case with keep_vars	PASS	MP_ASSERTDATASET: proc compare base=out_expected compare=out_test
(%derive_locf_records)[test09] Compare expected and test results for case with keep_vars	PASS	MP_ASSERTDATASET: proc compare base=out_expected compare=out_test
[mp_assertdataset] (%derive_var_age_years) Compare expected and test results for case of age_unit=variable name	PASS	MP_ASSERTDATASET: proc compare base=e_data compare=o_data
[mp_assertdataset] (%derive_var_age_years) Compare expected and test results for case of age_unit=character literal	PASS	MP_ASSERTDATASET: proc compare base=e_data compare=o_data
[mp_assertdataset] (%derive_var_age_years) Compare expected and test results for case of digits=2	PASS	MP_ASSERTDATASET: proc compare base=e_data compare=o_data
(%derive_vars_duration)[test01] Compare expected and test results for case of trunc_out=Y	PASS	MP_ASSERTDATASET: proc compare base=test1_exp compare=test1_op
(%derive_vars_duration)[test02] Compare expected and test results for case of datetime data with floor_in=Y	PASS	MP_ASSERTDATASET: proc compare base=test2_exp compare=test2_op
(%derive_vars_duration)[test03] Compare expected and test results for case of datetime data with floor_in=N	PASS	MP_ASSERTDATASET: proc compare base=test3_exp compare=test3_op
(%derive_vars_duration)[test04] Compare expected and test results for case of type=interval	PASS	MP_ASSERTDATASET: proc compare base=test4_exp compare=test4_op
[mp_assertdataset] (%derive_vars_dy) Compare expected and test results	PASS	MP_ASSERTDATASET: proc compare base=_adsl_expected compare=_adsl_test

Additional comments

NA

References

https://github.com/PharmaForest/adamski
https://github.com/PharmaForest/adamski/blob/main/Adamski_and_Admiral.md