

# DOSEON: Fuzzy Matching DOSE Date Intervals ON Analysis Dates Across SAS (PROC SQL, SAS Macro, DATA Step, and PROC FCMP)

Inka Leprince & Troy Martin Hughes



# Introducing....

## Troy Martin Hughes

Troy is an independent consultant specializing in implementing SAS and Python solutions. He has authored three SAS textbooks, with *PROC FCMP User-Defined Functions, Second Edition* published in 2026. He is a US Navy veteran with two deployments to Afghanistan.

## Inka Leprince, MS

Inka Leprince is an independent biostatistician & statistical programmer with over five years of experience consulting. In her free time, she likes to cross stitch, talk to her succulent collection, and walk through Costco.



US Connect 2026 SM15

# What is CDISC

- Clinical Data Interchange Standards Consortium (CDISC) develops and promotes data standards that facilitate the collection, organization, and analysis of data in the clinical and health research (CDISC Website, 2024)
- Analysis Data Model (ADaM) is optimized to support data derivation and analysis. A framework that enables analysis of the data, while at the same time providing traceability from collection to analysis to results (ADaM v2.1, 2009).



# DOSEON

| VARIABLE NAME | VARIABLE LABEL                 | TYPE | CODELIST | CT | SubClass<br>ADVERSE<br>EVENT<br>Core | CDISC Notes                                                                                                                                                                          |
|---------------|--------------------------------|------|----------|----|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DOSEON        | Treatment Dose at Record Start | Num  |          |    | Perm                                 | Dose received at the point in time of the record start date<br>Example derivation:<br>Obtained from EX.EXDOSE where<br>--STDTC falls between the values of EX.EXSTDTC and EX.EXENDTC |
| DOSEU         | Treatment Dose Units           | Char | (UNIT)   |    | Cond                                 | The units associated with DOSEON and/or DOSCUMA. Conditional on whether DOSEON and/or DOSCUMA are included.                                                                          |

Source: ADAM OCCDS v1.1 Table 3.2.7.1 Treatment/Dose Variables

# Not these variables --DOSExx

| VARIABLE NAME | VARIABLE LABEL                       | TYPE | CORE | CDISC NOTES                                                                                                                           |
|---------------|--------------------------------------|------|------|---------------------------------------------------------------------------------------------------------------------------------------|
| DOSExxP       | Planned Treatment Dose for Period xx | Num  | Perm | Subject-level identifier that represents the planned treatment dosage for period xx.                                                  |
| DOSExxA       | Actual Treatment Dose for Period     | Num  | Perm | Subject-level identifier that represents the actual treatment dosage for period xx.                                                   |
| DOSExxU       | Units for Dose for Period xx         | Char | Perm | The units for DOSExxP and DOSExxA. It is permissible to use suffixes such as "P" and "A" if needed, with labels modified accordingly. |

Source: ADAMIG v1.3 Table 3.2.5  
ADSL Dose Variables

| VARIABLE NAME | VARIABLE LABEL                    | TYPE | CORE | CDISC NOTES                                                                                                                                          |
|---------------|-----------------------------------|------|------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| DOSEP         | Planned Treatment Dose            | Num  | Perm | DOSEP represents the planned treatment dosage associated with the record.                                                                            |
| DOSCUMP       | Cumulative Planned Treatment Dose | Num  | Perm | Cumulative planned dosage of treatment for the subject at the point in time of the record (e.g., ADT).                                               |
| DOSEA         | Actual Treatment Dose             | Num  | Perm | DOSEA represents the actual treatment dosage associated with the record.                                                                             |
| DOSCUMA       | Cumulative Actual Treatment Dose  | Num  | Perm | Cumulative actual dosage of treatment for the subject at the point in time of the record (e.g., ADT).                                                |
| DOSEU         | Treatment Dose Units              | Char | Perm | The units for DOSEP, DOSCUMP, DOSEA, and DOSCUMA. It is permissible to use suffixes such as "P" and "A" if needed, with labels modified accordingly. |

Source: ADAMIG v1.3 Table 3.3.2.2. Record-Level Dose Variables for BDS Data Sets



US Connect 2026 SM15

# DOSEON Requires Fuzzy Matching

“ A merge in which there is no combination of key variables available that will match up the observations to be joined is known as a fuzzy merge. ”

Art Carpenter, 2014

| VARIABLE NAME | VARIABLE LABEL                 | TYPE | CDISC Example Derivation                                                                    |
|---------------|--------------------------------|------|---------------------------------------------------------------------------------------------|
| DOSEON        | Treatment Dose at Record Start | Num  | Obtained from EX.EXDOSE where --STDTC falls between the values of EX.EXSTDTC and EX.EXENDTC |

Source: ADAM OCCDS v1.1 Table 3.2.7.1 Treatment/Dose Variables

US Connect 2026 SM15

# Agenda

**1. PROC SQL**

**2. SAS MACRO**

**3. DATA STEP**

**4. PROC FCMP**



A CASE STUDY

# CAMELOT

A Phase 1, open-label, dose escalation study to evaluate safety of camel milk (CAMILK) in adult subjects with advanced cancer



# ADEX

| #  | SUBJID            | ACYCLE                | EXTRT                    | EXDOSE      | EXDOSEU           | ASTDT                      | AENDT                    | EXREASND               |
|----|-------------------|-----------------------|--------------------------|-------------|-------------------|----------------------------|--------------------------|------------------------|
|    | <i>Subject ID</i> | <i>Analysis Cycle</i> | <i>Name of Treatment</i> | <i>Dose</i> | <i>Dose Units</i> | <i>Analysis Start Date</i> | <i>Analysis End Date</i> | <i>Reason Not Done</i> |
| 1  | DUNEY             | CYCLE 01              | CAMILK                   | 500         | mL                | 2025-06-29                 | 2025-07-19               |                        |
| 2  | DUNEY             | CYCLE 02              | CAMILK                   | 500         | mL                | 2025-07-20                 | 2025-08-09               |                        |
| 3  | DUNEY             | CYCLE 03              | CAMILK                   | 500         | mL                | 2025-08-10                 | 2025-08-29               |                        |
| 4  | DUNEY             | CYCLE 04              | CAMILK                   | 500         | mL                | 2025-08-30                 | 2025-09-16               |                        |
| 5  | SANDY             | CYCLE 01              | CAMILK                   | 1000        | mL                | 2026-03-27                 | 2026-04-16               |                        |
| 6  | SANDY             | CYCLE 02              | CAMILK                   | 1000        | mL                | 2026-04-17                 | 2026-04-21               |                        |
| 7  | SANDY             | CYCLE 02              | CAMILK                   | 0           | mL                | 2026-04-22                 | 2026-04-23               | FORGOT TO DRINK CAMILK |
| 8  | SANDY             | CYCLE 02              | CAMILK                   | 1000        | mL                | 2026-04-24                 | 2026-04-26               |                        |
| 9  | SANDY             | CYCLE 02              | CAMILK                   | 0           | mL                | 2026-04-27                 | 2026-04-30               | ADVERSE EVENT          |
| 10 | SANDY             | CYCLE 02              | CAMILK                   | 1000        | mL                | 2026-05-01                 | 2026-05-02               |                        |



# ADAE

| # | SUBJID     | DOSEON                         | DOSEU                | ASTDT               | AENDT             | AETERM                      |
|---|------------|--------------------------------|----------------------|---------------------|-------------------|-----------------------------|
|   | Subject ID | Treatment Dose at Record Start | Treatment Dose Units | Analysis Start Date | Analysis End Date | Adverse Event Reported Term |
| 1 | DUNEY      |                                |                      | 2025-06-15          |                   | BACK PAIN                   |
| 2 | DUNEY      | 500                            | mL                   | 2025-07-06          |                   | WORSENING NAUSEA            |
| 3 | DUNEY      | 500                            | mL                   | 2025-07-23          |                   | CONSTIPATION                |
| 4 | DUNEY      | 750                            | mL                   | 2025-12-27          | 2025-12-29        | SEEING MIRA                 |

# ADLIVER

| # | SUBJID     | DOSEON                         | DOSEU                | AVISIT         | ADT           | PARAMCD        | AVAL           |
|---|------------|--------------------------------|----------------------|----------------|---------------|----------------|----------------|
|   | Subject ID | Treatment Dose at Record Start | Treatment Dose Units | Analysis Visit | Analysis Date | Parameter Code | Analysis Value |
| 1 | DUNEY      |                                |                      | Screening      | 2025-06-20    | ALP            | 101            |
| 2 | DUNEY      | 500                            | mL                   | Cycle 01 Day 1 | 2025-06-29    | ALP            | 101            |
| 3 | DUNEY      | 500                            | mL                   | Cycle 02 Day 1 | 2025-07-19    | ALP            | 101            |
| 4 | DUNEY      | 500                            | mL                   | Cycle 03 Day 1 | 2025-08-09    | ALP            | 101            |
| 5 | DUNEY      | 500                            | mL                   | Cycle 04 Day 1 | 2025-08-29    | ALP            | 98             |
| 6 | DUNEY      | 0                              | mL                   | Cycle 05 Day 1 | 2025-09-20    | ALP            | 112            |



US Connect 2026 SM15

# DOSEON Derivation

| VARIABLE NAME | VARIABLE LABEL                 | TYPE | SOURCE  | DERIVATION                                                                                                                             |
|---------------|--------------------------------|------|---------|----------------------------------------------------------------------------------------------------------------------------------------|
| DOSEON        | Treatment Dose at Record Start | Num  | Derived | If the analysis (start) date (ADT or ASTDT) falls between the values of ADEX.ASTDT and ADEX.AENDT (inclusive), then set to ADEX.EXDOSE |

Note: Depending on the analysis requirements for the study, the derivation of DOSEON may vary

# Exploding ADEX

```
data explode_adex;  
  set adex (rename=(astdt=ex_stdtdt  
                    aendt=ex_endtdt));  
  do ASTDTC = ex_stdtdt to ex_endtdt;  
    output;  
  end;  
run;
```

**NOTE:** There were 192 observations read from the data set WORK.ADEX.

**NOTE:** The data set WORK.EXPLODE\_ADEX has 1871 observations and 17 variables.

In all solutions, the fuzzy match is operationalized by exploding keys, such that each potential key (in a range of keys) is explicitly declared and associated key-value pairs created in the lookup table (either temporarily or permanently).

# PROC SQL

```
PROC SQL ;  
  /* ADAE.DOSEON derivation */  
  create table ADAE_SQL as  
    select ae.*,  
           ex.exdose as DOSEON,  
           ex.exdoseu as DOSEU  
  from adae as ae  
  LEFT JOIN adex as ex  
  ON ex.subjid=ae.subjid and  
     ae.astdt BETWEEN ex.astdt AND ex.aendt and  
     NOT MISSING(ae.astdt)  
  order by ae.subjid, ae.aeseq  
  ;
```

# PROC SQL

```
/* ADLIVER.DOSEON derivation */  
create table ADLIVER_SQL as  
  select lb.*,  
         ex.exdose as DOSEON,  
         ex.exdoseu as DOSEU  
  from adliver as lb  
  LEFT JOIN adex as ex  
  ON ex.subjid=lb.subjid and  
     lb.adt BETWEEN ex.astdt AND ex.aendt and  
     NOT MISSING(lb.adt)  
  order by lb.subjid, lb.adt, lb.paramcd  
;  
QUIT ;
```

# SAS Macro

```
/* Define the SAS macro DERIVE_DOSEON */
%macro DERIVE_DOSEON(indata =, indate = );
PROC SQL;
    create table &INDATA._MACRO as
        select indata.*,
               ex.exdose as DOSEON,
               ex.exdoseu as DOSEU
        from &INDATA as indata
        left join adex as ex
        on ex.subjid=indata.subjid and
           &INDATA..&INDATE BETWEEN ex.astdt AND ex.aendt and
           NOT MISSING(&INDATA..&INDATE)
    ;
QUIT ;
%mend ;

/* Call DERIVE_DOSEON for ADAE and ADLIVER */
%derive_doseon(indata = ADAE, indate = ASTDT) ;
%derive_doseon(indata = ADLIVER, indate = ADT ) ;
```

# DATA Step Hash Object with Hash Iterator

```
data ADAE_data_step_hash (drop=_dt _rc _aendt_temp _skip rename=(exdose=DOSEON exdoseu=DOSEU));
  length _dt 8 subjid $10 astdt 8 aendt 8 exdose 8 exdoseu $20 _aendt_temp 8;
  if _n_ = 1 then do;
    * initialize the hash object using the lookup table;
    declare hash h(dataset: 'adex', ordered: 'a');
    declare hiter iter('h');
    _rc = h.defineKey('subjid', 'astdt');
    _rc = h.defineData('subjid', 'astdt', 'aendt', 'exdose', 'exdoseu');
    call missing(exdose, exdoseu);
    _rc = h.defineDone();
    * explode the date ranges into new obs (keys);
    do while(iter.next()=0);
      _dt = astdt;
      do astdt = _dt to aendt;
        _rc = h.add();
      end;
      * skip past key-values that were just added;
      do _skip = 1 to (aendt-_dt);
        _rc = iter.next();
      end;
    end;
    * reset variables used to initialize hash object;
    call missing(subjid, astdt, aendt, exdose, exdoseu);
  end;
  * read the transactional data;
  set adae;
  * perform lookup operation;
  _aendt_temp = aendt;
  _rc = h.find();
  aendt = _aendt_temp;
run;
```

- hash object is instantiated before ADAE is read, and the ADEX data set is ingested into the hash
- hiter object explodes hash object from 192 to 1,871 observations
- SET statement iteratively reads each observation of ADAE
- hash FIND method performs lookup by SUBJID and ASTDT, and returns matched values

# Procedural Design in SAS

```
data ADAE_data_step_hash (drop=_dt _rc _aendt_temp _skip rename=(exdose=DOSEON exdoseu=DOSEU));
  length _dt 8 subjid $10 astdt 8 aendt 8 exdose 8 exdoseu $20 _aendt_temp 8;
  if _n_ = 1 then do;
    * initialize the hash object using the lookup table;
    declare hash h(dataset: 'adex', ordered: 'a');
    declare hiter iter('h');
    _rc = h.defineKey('subjid', 'astdt');
    _rc = h.defineData('subjid', 'astdt', 'aendt', 'exdose', 'exdoseu');
    call missing(exdose, exdoseu);
    _rc = h.defineDone();
    * explode the date ranges into new obs (keys);
    do while(iter.next()=0);
      _dt = astdt;
      do astdt = _dt to aendt;
        _rc = h.add();
      end;
      * skip past key-values that were just added;
      do _skip = 1 to (aendt-_dt);
        _rc = iter.next();
      end;
    end;
    * reset variables used to initialize hash object;
    call missing(subjid, astdt, aendt, exdose, exdoseu);
  end;
  * read the transactional data;
  set adae;
  * perform lookup operation;
  _aendt_temp = aendt;
  _rc = h.find();
  aendt = _aendt_temp;
run;
```

- comprises one or more DATA steps and/or SAS procedures
- exposes the end user to low-level mechanics of *how* functionality is delivered
- often constructed in monolithic code behemoths
- code “reuse” is conceptualized as cutting and pasting code

# Functional Design in SAS

```
data ADAE_function_hash_v1 drop=rc
  rename=(exdose=DOSEON exdoseu=DOSEU));
set adae;
length exdose 8 exdoseu $20;
call missing(exdose, exdoseu);
* initialize the EXDOSE and EXDOSEU
  variables from the ADEX lookup table;
rc=set_doseon_doseu_from_adex(subjid,
  astdt, aendt, exdose, exdoseu);
run;
```

- functions and subroutines are called in a single statement
- exposes the end user only to high-level functionality (i.e., *what* is done, not *how*)
- modular design, with PROC FCMP functions and subroutines
- code reuse is conceptualized as calling these compiled modules



# Defining the SET\_DOSE Function

```
proc fcmp outlib=work.funcs.pharma;
  function set_doseon_doseu_from_adex(subjid $, astdt, aendt,
    exdose, exdoseu $);
    outargs exdose, exdoseu;
    static called 0;

    * instantiate the hash object and iterator object;
    declare hash h(dataset: 'adex', ordered: 'a');
    declare hiter iter('h');
    rc = h.defineKey('subjid', 'astdt');
    rc = h.defineData('subjid', 'astdt', 'aendt', 'exdose', 'exdoseu');
    rc = h.defineDone();

    * explode the dates when function is first called;
    if called=0 then do;
      called = 1;
      call explode_hash(h, iter);
    end;

    * query hash object to return values;
    rc = h.find();
    return(rc);
endfunc;
quit;
```

- SET\_DOSE declares the hash object, instantiates it, and iterates it using hiter object
- SET\_DOSE calls a user-defined subroutine (EXPLODE\_HASH) to explode the lookup table dates
- hash FIND method retrieves EXDOSE and EXDOSEU from the hash object
- SET\_DOSE returns EXDOSE and EXDOSEU to the DATA step

# Defining the EXPLODE\_HASH Subroutine

```
proc fcmp outlib=work.funcs.pharma;
  subroutine explode_hash(h hash, iter hiter);
    length dt 8 subjid $10 astdt aendt exdose 8
      exdoseu $20;
    do while(iter.next()!=0);
      dt=astdt;
      * explode date to add key-value pairs;
      do astdt = dt to aendt;
        rc = h.add();
        end;
      * skip past key-values that were just added;
      do i = 1 to (aendt-astdt);
        rc = iter.next();
        end;
      end;
    endsub;
quit;
```

- SUBROUTINE declares subroutine name and its parameters
- HASH and HITER keywords declare the parameter data types
- hash NEXT method iterates the hash object
- hash ADD method adds new hash key-value pairs
- changes made to hash and hiter persist in SET\_DOSE function



# Contact the Authors

Inka Leprince

E-mail: [leprince@inkastat.com](mailto:leprince@inkastat.com)

Website: [www.inkastat.com](http://www.inkastat.com)

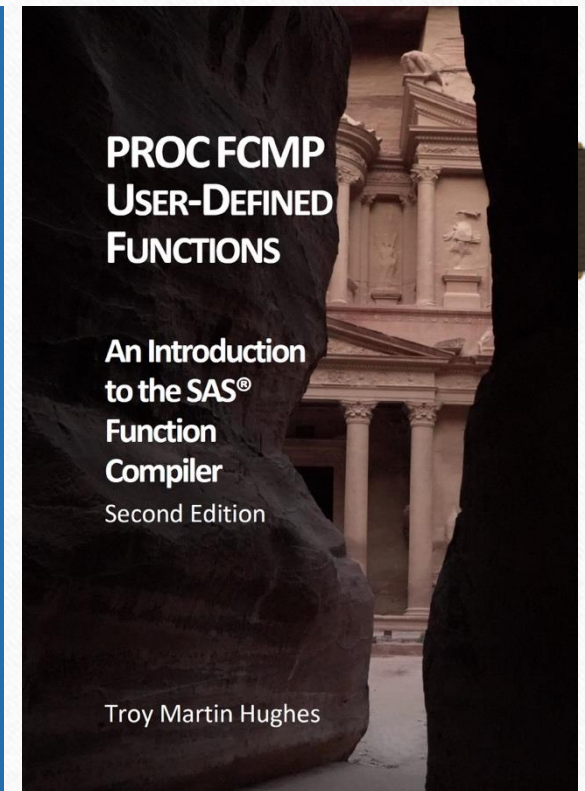
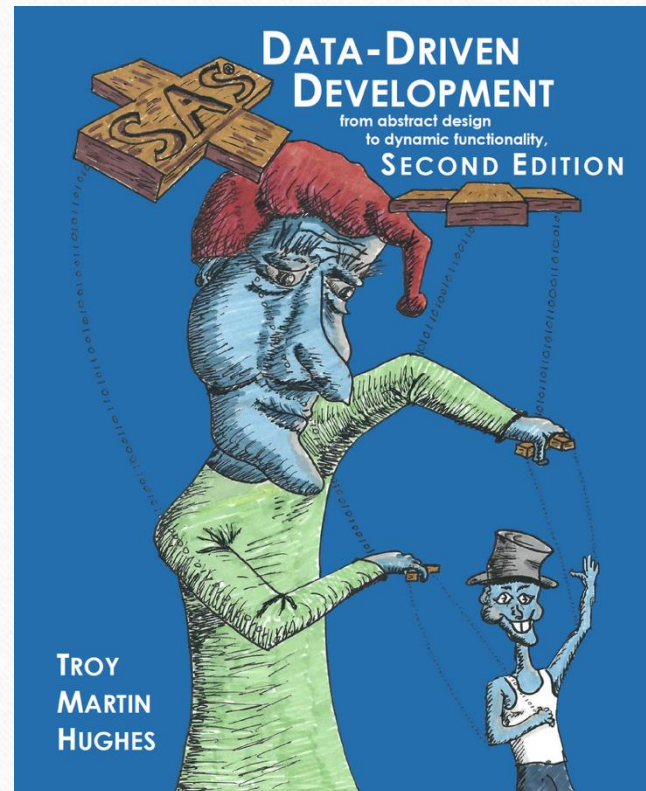
LinkedIn:



Troy Martin Hughes



US Connect 2026 SM15



# References

- Andrew Henrick, D. E. (2013). Hashing in PROC FCMP to Enhance Your Productivity. SAS Global Forum (SGF). SAS Institute.
- Carpenter, A. L. (2014). DATA Step Merging Techniques: From Basic to Innovative. SAS Global Forum. Washington, D.C.: sas.com. Retrieved from <https://support.sas.com/resources/papers/proceedings14/1271-2014.pdf>
- Carpenter, A. L. (2015). Using Arrays to Quickly Perform Fuzzy Merge Look-ups: Case Studies in Efficiency. PharmaSUG. Orlando: PharmaSUG. Retrieved January 2026, from <https://pharmasug.org/proceedings/2015/TT/PharmaSUG-2015-TT06.pdf>
- CDISC ADaM Team. (2009, December 7). ADaM v2.1. Retrieved April 2, 2024, from CDISC: <https://www.cdisc.org/standards/foundational/adam/adam-v2-1>
- CDISC ADaM Team. (2016, February 12). ADaM Structure for Occurrence Data Implementation Guide v1.1. Austin, Texas, USA: Clinical Data Interchange Standards Consortium. Retrieved April 2, 2024, from CDISC: <https://www.cdisc.org/standards/foundational/adam/adam-structure-occurrence-data-implementation-guide-v1-1>
- CDISC Analysis Data Model Team. (2021, November 29). Analysis Data Model Implementation Guide Version 1.3 (Final). Retrieved March 26, 2024, from CDISC: <https://www.cdisc.org/standards/foundational/adam/adamig-v1-3>
- CDISC Website. (2024). Standards. Retrieved April 2, 2024, from CDISC: <https://www.cdisc.org/standards>
- Hughes, T. M. (2022). Data-Driven Development from Abstract Design to Dynamic Functionality (2nd ed.). Cary, North Carolina, USA: SAS Institute, Inc.
- Hughes, T. M. (2026). PROC FCMP User-Defined Functions: An Introduction to the SAS Function Compiler, Second Edition (1 ed.). San Diego, CA, North Carolina, United States of America.
- Leprince, I., & Watson, R. J. (2024). Around the Data DOSE-y Doe, How Much Fun Can Your Data Can Be: Using DOSExx Variables within ADaM Data Sets. PharmaSUG. Baltimore. Retrieved from <https://pharmasug.org/proceedings/2024/DS/PharmaSUG-2024-DS-193.pdf>
- SAS Institute Inc. (2016). SAS® 9.4 SQL Procedure User's Guide, Fourth Edition (4 ed.). Cary, NC, North Carolina, USA: SAS Institute Inc. Retrieved from <https://documentation.sas.com/api/collections/pgmbasecdc/9.4/docsets/sqlproc/content/sqlproc.pdf?locale=en>
- SAS Institute Inc. (n.d.). SAS® SQL 1: Essentials. Cary, North Carolina, USA. Retrieved September 2019