



Dictionary tables- a programmer's friend?

Marcin Pracz

Quintiles

Introduction

- Dictionary tables- vast source of knowledge about our SAS environment
- 32 tables available in SAS® 9.4 covering information from defined libnames, tables (views, catalogues, indexes, table constraints) and variables to SAS options, macro variables, formats and defined functions

Dictionary.dictionaries

- Metadata about metadata
- **proc sql;**
- `select * from dictionary.dictionaries;`
- **quit;**

	Member Name	Data Set Label	Column Name	Column Type	Column Length	Column Position	Column Number in Table	Column Label
1	MEMBERS	Tables, catalogs, and views	LIBNAME	char	8	0	1	Library Name
2	MEMBERS	Tables, catalogs, and views	MEMNAME	char	32	8	2	Member Name
3	MEMBERS	Tables, catalogs, and views	MEMTYPE	char	8	40	3	Member Type
4	MEMBERS	Tables, catalogs, and views	DBMS_MEMTYPE	char	32	48	4	DBMS Member Type
5	MEMBERS	Tables, catalogs, and views	ENGINE	char	8	80	5	Engine Name
6	MEMBERS	Tables, catalogs, and views	INDEX	char	3	88	6	Indexes
7	MEMBERS	Tables, catalogs, and views	PATH	char	1024	91	7	Pathname
8	TABLES	Tables and table-specific information	LIBNAME	char	8	0	1	Library Name
9	TABLES	Tables and table-specific information	MEMNAME	char	32	8	2	Member Name
10	TABLES	Tables and table-specific information	MEMTYPE	char	8	40	3	Member Type
11	TABLES	Tables and table-specific information	DBMS_MEMTYPE	char	32	48	4	DBMS Member Type
12	TABLES	Tables and table-specific information	MEMLABEL	char	256	80	5	Data Set Label
13	TABLES	Tables and table-specific information	TYPEMEM	char	8	336	6	Data Set Type
14	TABLES	Tables and table-specific information	CRDATE	num	8	344	7	Date Created
15	TABLES	Tables and table-specific information	MODATE	num	8	352	8	Date Modified

- **proc sql;**
- **describe table dictionary.tables;**
- **quit;**

- (Excerpt from log)
- NOTE: SQL table DICTIONARY.TABLES was created like:
- create table DICTIONARY.TABLES
- (libname char(8) label='Library Name',
- memname char(32) label='Member Name',
- memtype char(8) label='Member Type',
- dbms_memtype char(32) label='DBMS Member Type',

Using dictionary tables for checking data issues

- Can check entire folders for common issues with one macro
- For some more obvious problems, don't need any information about the data
- Example- data issues related to dates:
 - ✓ Often mistakes in year of the procedure (2015 instead of 2016 etc.)
 - ✓ Start date swapped with end date
 - ✓ Can detect suspicious patterns (like 50% of adverse event start dates happening on a Tuesday)

Date variables

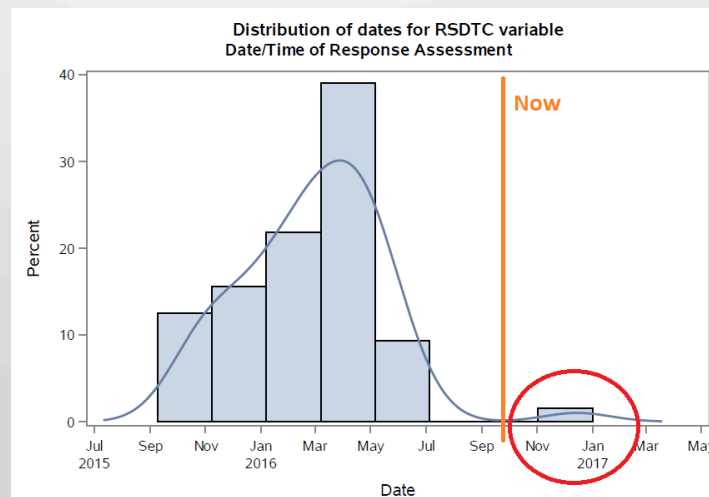
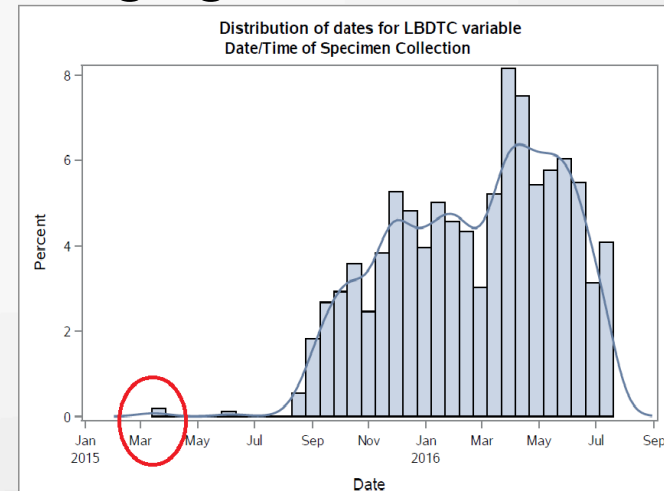
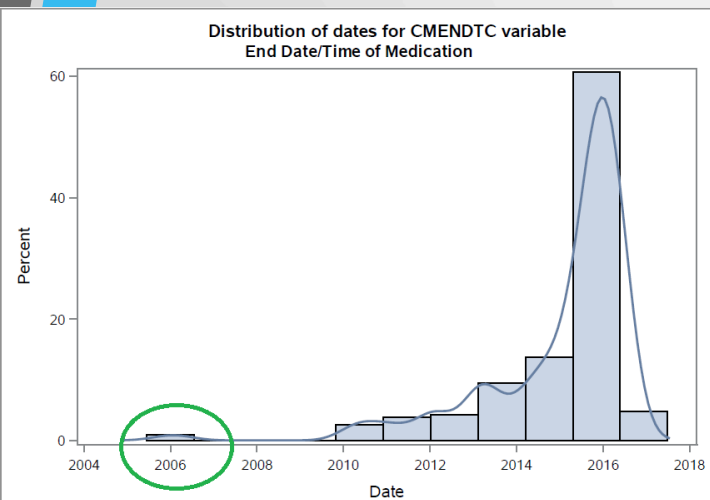
- Detect date variables by their label (use dictionary.columns)
- %macro get_date_vars(in_lib=,out_ds=);
 - proc sql;
 - create table pre_&out_ds. as
 - select * from dictionary.columns
 - where libname = upcase("&in_lib.")
 - order by memname;
 - quit;
 - data &out_ds.;
 - set pre_&out_ds.;
 - where index(upcase(label),'DATE')>0;
 - run;
- %mend get_date_vars;

- List of all date variables from all datasets in our data transfer
- Could put Member Name and Column Name into macro variables and source raw values for every date variable one by one
- If sorted by value- it would be easy to inspect for outliers
- Start and End date pairs- check if end date is not earlier than start

	Library Name	Member Name	Column Name	Column Type	Column Length	Column Position	Column Number in Table	Column Label
1	SDTM	AE	AEENDTC	char	16	642	36	End Date/Time of Adverse Event
2	SDTM	AE	AESTDTC	char	16	626	35	Start Date/Time of Adverse Event
3	SDTM	CM	CMENDTC	char	10	548	17	End Date/Time of Medication
4	SDTM	CM	CMSTDTC	char	10	538	16	Start Date/Time of Medication
5	SDTM	DM	BRTHDTC	char	10	141	14	Date/Time of Birth
6	SDTM	DM	DMDTC	char	10	242	25	Date/Time of Collection
7	SDTM	DM	DTHDTC	char	1	136	11	Date/Time of Death
8	SDTM	DM	RFENDTC	char	16	68	6	Subject Reference End Date/Time
9	SDTM	DM	RFICDTC	char	10	116	9	Date/Time of Informed Consent
10	SDTM	DM	RFPENDTC	char	10	126	10	Date/Time of End of Participation
11	SDTM	DM	RFSTDTC	char	16	52	5	Subject Reference Start Date/Time

Checking outliers

- Not easy to decide what is a valid outlier and what not
- LBDTC- ICF date for a patient was 2016-03-23 (data issue)
- CMENDTC- valid outlier, despite all other measurements coming at least 4 years later



Start/End dates

- Pairs of date variables with labels that differ only by either including the word „start“ or „end/stop“, coming from the same dataset
- If Member Name and Column Names for each pair were put in a macro variable, they could be sourced jointly (in a do loop for all such pairs)

	Library Name	Member Name	Column Name	Column Type	Column Length	Column Position	Column Number in Table	Column Label
1	SDTM	AE	AEENDTC	char	16	642	36	End Date/Time of Adverse Event
2	SDTM	AE	AESTDTC	char	16	626	35	Start Date/Time of Adverse Event

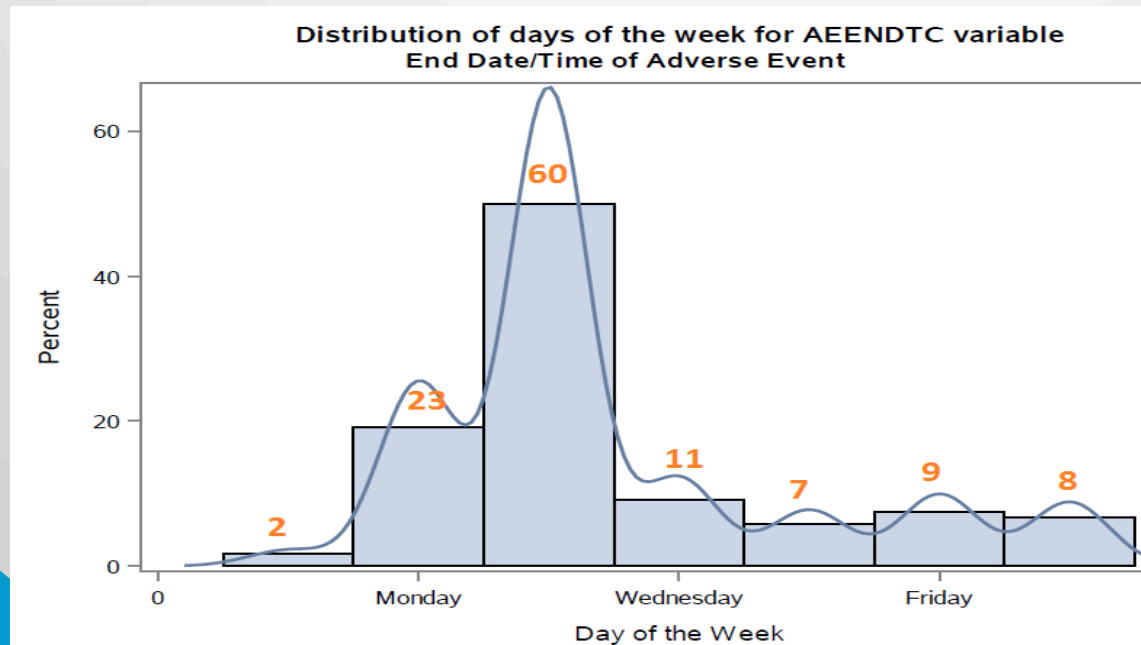
- Check if start dates not later than end/stop dates
- Check for outliers in event durations (duration = end date – start date + 1)
- Again, more difficult to decide when outlier is a data issue
- In general, expect big outliers in MH and CM. For other domains, outliers are usually data issues

DSET_NAME	VAR_NAMES	START_DATE	END_DATE	DURATION_DAYS
XH	XHENDTC XHSTDTC	2015-12-15T17:50	2015-12-17T12:00	3
XH	XHENDTC XHSTDTC	2016-01-11T15:32	2016-01-13T12:30	3
XH	XHENDTC XHSTDTC	2016-01-26T16:15	2016-01-28T12:00	3
XH	XHENDTC XHSTDTC	2015-01-19T17:27	2016-01-21T14:00	368
XH	XHENDTC XHSTDTC	2016-02-02T13:00	2016-02-04T12:00	3
XH	XHENDTC XHSTDTC	2015-01-26T16:59	2016-01-28T11:00	368
XH	XHENDTC XHSTDTC	2016-02-09T16:05	2016-02-11T12:00	3
XH	XHENDTC XHSTDTC	2016-03-08T19:30	2016-03-10T12:00	3
XH	XHENDTC XHSTDTC	2016-03-23T13:41	2016-03-25T11:18	3

**Data issue- wrong
XHSTDTC**

Possibly more advanced analysis?

- All date values in one place- more advanced techniques to check distribution of days of the week, time of day, date clusters
- How likely it is that out of 120 AEs with a populated end date, as much as 60 (exactly half) happened on a Tuesday?



Conclusion

- With dictionary tables, we can look for issues through multiple datasets (even entire librefs)
- Whole data transfers of raw/SDTM data could be inspected automatically for obvious issues, without any *a priori* knowledge about the data
- Some human input might be needed (like when deciding if an outlier is a data issue)
- If not, macro results could be exported straight into data issue log, simplifying data cleaning process

Thank you

