

エクセル読み込みの xlsxエンジン

マルホ株式会社 山野辺

ディスクレーマー

- ▶ 本発表は個人の見解に基づくものであり、所属組織を代表するものではありません。あらかじめご了承ください。
- ▶ 確認環境はPC-SAS, SAS Studioです。

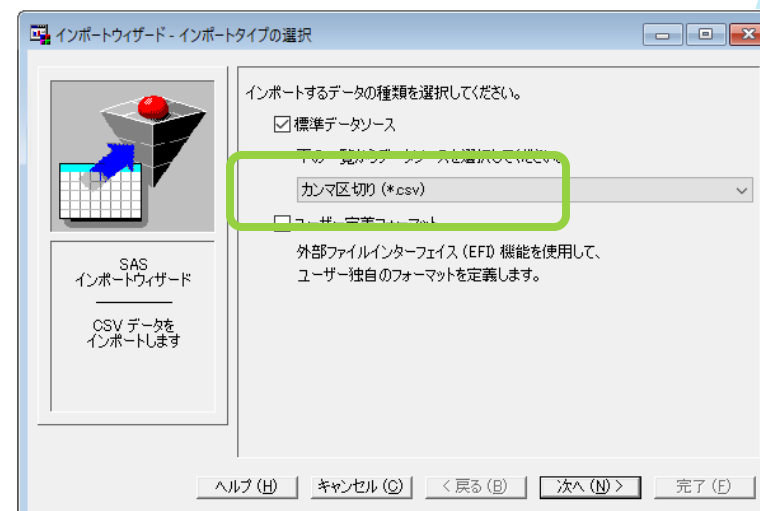
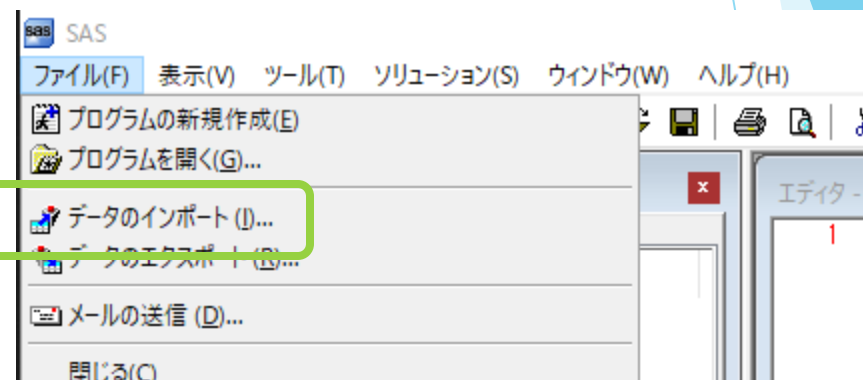
Libname エンジン？

- ▶ 業務環境にSAS/ACCESSライセンスがないが、業務上エクセルを読み込むことが多く、エクセルファイルの取扱いに苦労していた
- ▶ 使えないからこそ、メリット/デメリットを把握しておきたかった
- ▶ 調べてみての差分、挙動の違いについて報告する

SASでのエクセルファイルの読み込み

SASでのExcelファイルの読み込み方-1

- ▶ そもそもSASでExcelを読み込むには？
- ▶ インポートウィザードに従えば、
GUI上で読み込み、かつGUIで指定したオプション
を反映した
SASプログラムまで生成してくれる
(DBMSを指定したproc importプログラム)



SASでのExcelファイルの読み込み方-2

- ▶ そもそもSASでExcelを読み込むには？
- ▶ DDE(Dynamic Data Exchange)を使って読み込む
- ▶ MSでのサポートが明示されておらず、参照文献も限られ新規で作成するには難易度が高い

SASでのExcelファイルの読み込み方-3

- ▶ そもそもSASでExcelを読み込むには？
- ▶ Proc importやlibnameステートメントを用いて読み込む
 - 保存可能で、再現性もあり正道
 - proc importは読み込みにしか使えないが、libnameは書き込みも可能
 - この際に使用するエンジンを指定するがエンジンは[excel], [xlsx]が代表的に知られる

xlsxエンジンとは

- ▶ xlsxエンジンとは、libname statement中で指定するエンジン
- ▶ SAS/ACCESS 9.4M2 Interface to PC Filesからあり、普及している認識
- ▶ それまでのエクセル読み込みは[excelエンジン]を使っており、エクセルファイルに対するデフォルトは引き続き [excelエンジン]となっている。
- ▶ 例) libname FILE “./test.xlsx”;



libname FILE **excel** “./test.xlsx”;

エンジンの違い: excelエンジン

- ▶ [excelエンジン]は仕様が古く使うメリットが低い

メリット :

- ・ xlsファイルが読み込める
- ・ インポートウィザード生成のプログラムから、コピペで作成できる
- ・ .xls、.xlsx、.xlsb、.xlsm、.mdb、.accdbが対応可能

デメリット :

- ・ (xlsxエンジンと比較しても)エンジンによる自動変換が生じる
- ・ 変数型の判定が自動で行われ、型はデータ先頭8行目までの値で判断
- ・ 一部の値の読み込みが不完全（小数点深くの値が途中でロストする）
- ・ Windows環境下での動作

エンジンの違い: xlsxエンジン

- ▶ [xlsxエンジン]は新しいものの、下位互換はない

メリット :

- ・ UNIX、Windowsで動作 (SAS Studioでも動く！)
- ・ エクセルファイルの値をそのまま読みこむことが出来る。

デメリット :

- ・ xls形式は読み込み不可,XLSXファイルのみ可能。
- ・ excelエンジンで使えていた一部のオプションが使用不可
- ・ 変数名がEXCEL準拠(F1, F2,...やCol1, Col2...ではなくA, B, C,...,AA,AB,...)

	A	B	C	D
1	STUDYID	DOMAIN	USUBJID	LB
2	STUDY01	LB	ZZ001	BIC
3	STUDY01	LB	ZZ001	BIC

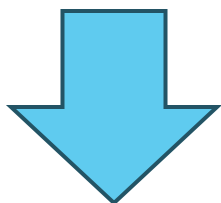
他の違い

- ▶ XLSXエンジンは逐次読み込みのため、時間がかかる（ランダムアクセス不可）。
- ▶ 上記理由に関連して、データセットに落とさず直接プロシジャで呼び出すと実行不可能なものがある（RANKプロシジャ等）
- ▶ XLSXエンジンはDBCS(double byte character set)をサポート
等、別エンジンであるため、違いはいくつもある。

エンジン間のオプションの違い

共通オプションは
GETNAMESのみ。

また、excelエンジンでは、
getnamesはlibname時にも使えた
が、xlsxエンジンではlibnameで
使えずに、proc import時にしか
使えない等の挙動の違いもある



単純なエンジン名の置換のみでは
excel→xlsxへの変更は不可！

Available Statements for Importing and Exporting Excel Files Using DBMS=XLS and DBMS=XLSX

DBMS= Identifier	Option	Valid Value	Default Value	PROC IMPORT	PROC EXPORT
XLS	ENDCOL	Last column for data	Last column that contains data	Yes	No
	ENDNAMEROW	Last row for variable names	Same as NAMEROW	Yes	No
	ENDROW	Last row for data	Last row that contains data	Yes	No
	GETNAMES	Yes No	Yes	Yes	No
	NAMEROW	First row for variable names	First row that contains variable names	Yes	No
	NEWFILE	Yes No	No	No	Yes
	PUTNAMES	Yes No	Yes	No	Yes
	RANGE	name sheet\$ul:lr	First row	Yes	No
	SHEET	Sheet name	First sheet	Yes	Yes
	STARTCOL	First column for data	Last column that contains data	Yes	No
XLSX	STARTROW	First row for data	First row that contains data	Yes	No
	DATAROW	1....	When GETNAMES=NO: 1, when GETNAMES=YES: 2	Yes	No
	GETNAMES	Yes No	Yes	Yes	No
	RANGE	name sheet\$ul:lr	First row	Yes	No
	SHEET	Sheet name	First sheet	Yes	Yes

excelエンジン→xlsxエンジンへの移行

- ▶ 別エンジンであるため、値やフォーマット、変数名のルールに互換はない
- ▶ SASに読み込んだデータセットを同一にするにはデータステップでの加工が必須

乗せ換え例

例:

▶ 変換元A : Excelエンジン

```
libname DT1 excel "test.xlsx" access=readonly;  
data SASDT1;  
    set DT1.sheet1;  
run;
```

▶ 変換元B : Excelエンジン

```
proc import datafile="test.xlsx" out=SASDT1 DBMS=excel;  
    range="sheet1 ";  
run;
```

▶ 変換後 : XLSXエンジン

```
proc import datafile="test.xlsx" out=SASDT1 DBMS=xlsx;  
    range="sheet1 ";  
run;
```

→作成したSASDT1は完全に同じとは言えず、変数名や値、当てられるformatが異なる

→コンマ以下の深い値がなければ、変数名を整え、日付にさえ気を遣えば、似たように取り扱えそう？

乗せ換え例

- ▶ 先頭行がラベル行の場合は、
(SAS変数名ルールから逸れない限り) 似たような形で読み込める

- ▶ 変換元A : Excelエンジン

```
libname DT1 excel "test.xlsx" getnames=YES;  
data SASDT1;  
    set DT1.sheet1;  
run;
```

- ▶ 変換元B : Excelエンジン

```
proc import datafile="test.xlsx" out=SASDT1 DBMS=excel;  
    getname=YES; range="sheet1 ";  
run;
```

- ▶ 変換後 : XLSXエンジン

```
proc import datafile="test.xlsx" out=SASDT1 DBMS=xlsx;  
    getname=YES; range="sheet1 ";  
run;
```

乗せ換え例

- ▶ 先頭行がラベル行ではない場合(Getnames=No)の場合は、変数名が「F1, F2, F3...」と「A, B, C,...,AA,AB」と異なるので、同じように扱おうとするとrenameの必要性・・・

- ▶ 変換後：XLSXエンジン

```
proc import datafile="test.xlsx" out=SASDT1 DBMS=xlsx;  
  getname=NO;  
run;
```

```
data SASDT2;  
  set SASDT1;  
  rename  A=F1  
         B=F2  
         ....;
```

```
run;
```

	A	B	C	D	E	F	G	H	I
1	STUDYID	DOMAIN	USUBJID	LBCAT	LBTESTCD	LBTEST	LBORRES	LBORRESU	LBDTC
2	STUDY01	LB	ZZ001	BIOCHEMISTRY	BILI	Bilirubin	1.23456789012345	mg/dL	45582.76666
3	STUDY01	LB	ZZ001	BIOCHEMISTRY	BILDIR	Direct Bilirubin	1.234	mg/dL	44064
4	STUDY01	LB	ZZ001	BIOCHEMISTRY	ACT	Aspartate Aminotransferase	1.234	U/L	44064

まとめ

- ▶ XlsxファイルをPC-SASから読み込む場合は、xlsxエンジンが無難。
- ▶ 古いシステムから出力されたファイルで[.xls]形式の場合は、excelエンジンを使わざるを得ない
- ▶ Excelエンジンからxlsxエンジンへの容易な置き換えは不可
- ▶ エクセルファイルの読み込みは、SASによる自動変換に十分注意

参照・参考文献

LIBNAME Statement: Access and Excel Engines

https://documentation.sas.com/doc/ja/pgmsascdc/9.4_3.5/acpcref/n1wvmggexroxygn17rp61jml3cvn.htm

LIBNAME Statement: XLSX Engine

https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/acpcref/n0oj9f6i838mymn148890ckla700.htm

Microsoft Workbook Files

https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/acpcref/n0msy4hy1so0ren1acm90iijxn8j.htm