04. 导入数据Ⅱ——Excel文件

（一）导入Excel数据文件

**一、import语句导入**

语法：

**proc** **import** datafile=’文件路径+文件名’ OUT=输出数据集名 DBMS=EXCEL REPLACE;

<可选参数>;

注：（1）REPLACE告诉SAS若“输出数据集”同名文件已经存在，则替换它；

（2）可选参数：

a. 指定要读取的是哪一个工作表

SHEET = ‘工作表名’;

b. 若只读取工作表的一部分范围

RANGE = ‘ $A1:H10’;

c. 是否从工作表的第一行读取数据集的列变量名？

GETNAMES=YES——是；

GETNAMES=NO——否；

d. 读取字符和数值混合的数据表时，是否将所有数据转化为字符？

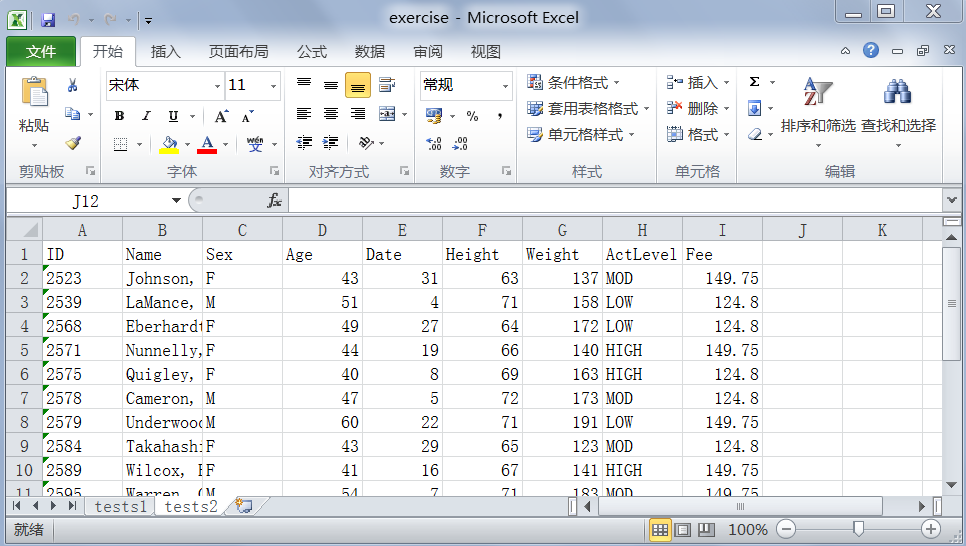
MIXED=YES——是；

MIXED=NO——否；

示例：

**proc** **import** DATAFILE = 'c:\MyRawData\OnionRing.xls' OUT=sales DBMS=XLS REPLACE;

**例1** 路径“D:\我的文档\My SAS Files\9.3\”下的数据文件exercise.xlsx，内容如下：



读取工作表test2中从A1到H10的数据，第一行作为数据集的列变量名。

代码：

**proc** **import** datafile = 'D:\我的文档\My SAS Files\9.3\exercise.xlsx' DBMS=EXCEL OUT = results REPLACE;

SHEET = 'tests2';

RANGE = '$A1:H10';

GETNAMES = YES;

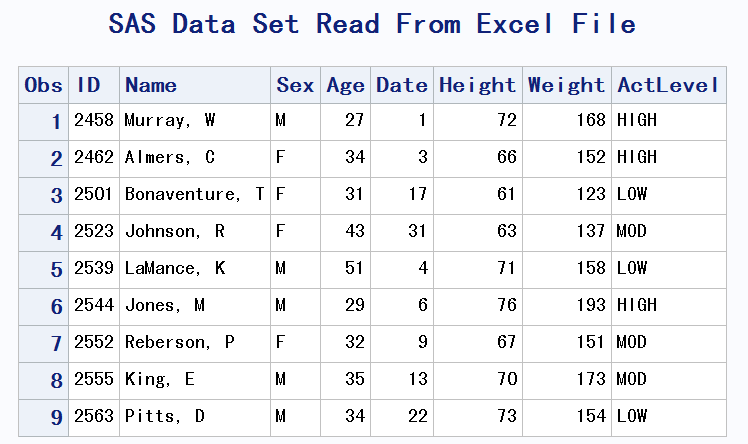
**run**;

**proc** **print** data = results;

title 'SAS Data Set Read From Excel File';

**run**;

运行结果：



**二、libname语句读入**

**1. 基本语法**

用libname语句引用一个Excel文件（“工作簿”），其中的“工作表”作为数据集，数据集名称为：’工作表名$’n

语法：

libname 引用名 ‘文件路径+文件名’ <可选参数>;

注：（1）访问数据集用：引用名. ’工作表名$’n

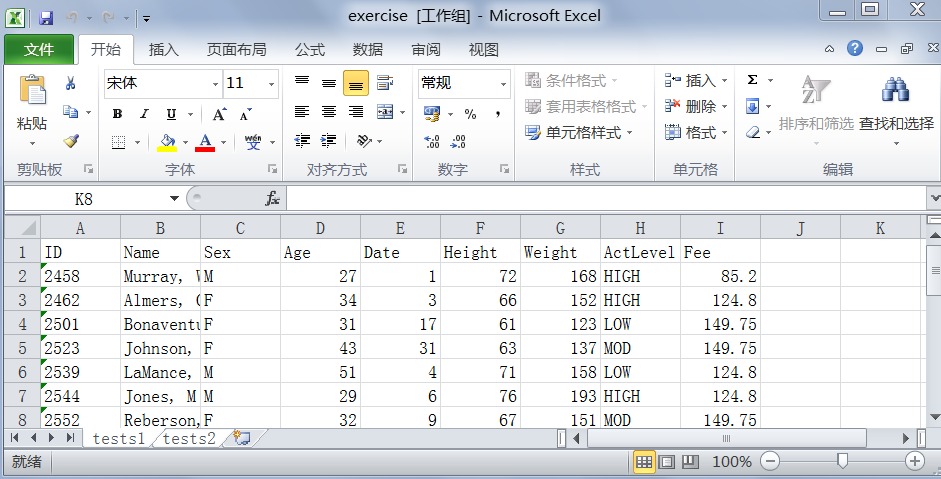
（2）工作表若有“名称框”（Named Range：单独命名的一部分区域），将单独作为数据集，区别是数据集名没有$

示例：

libname results 'D:\My SAS Files\exercise.xlsx';

**proc** **print** data=results.'tests1$'n;

**例2** 路径“D:\我的文档\My SAS Files\9.3\”下的数据文件exercise.xlsx，内容如下：



读取工作表tests1中的数据。

代码：

libname results 'D:\我的文档\My SAS Files\9.3\exercise.xlsx';

**proc** **print** data=results.'tests1$'n;

**run**;

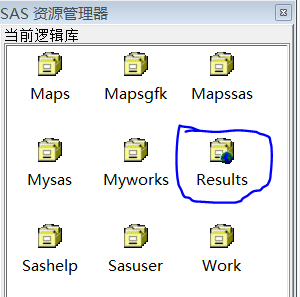
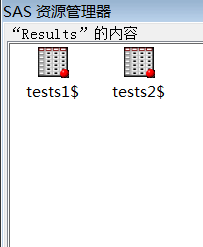
libname results clear;

运行结果（部分）：



程序说明：

（1）执行libname语句之后，将会在逻辑库里创建新逻辑库results，里面存放用数据表生成的数据集test1$和tests2$:

（2）libname语句提交之后，exercise.xlsx文件将与SAS相关联，Excel将不能打开编辑，要解除这种关联需要在程序最后加上“libname results clear;”（逻辑库中的results将消失）。

**2. libname语句的可选参数**

（1）GETNAMES=YES|NO

是否从工作表的第一行读取数据集的列变量名？

YES——是；

NO——否；

（2）MIXED=YES|NO

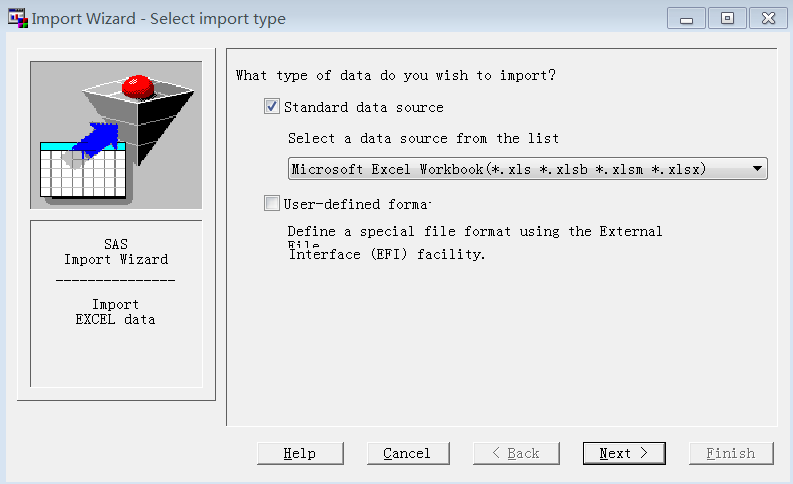
读取字符和数值混合的数据表时，是否将所有数据转化为字符？

YES——是；

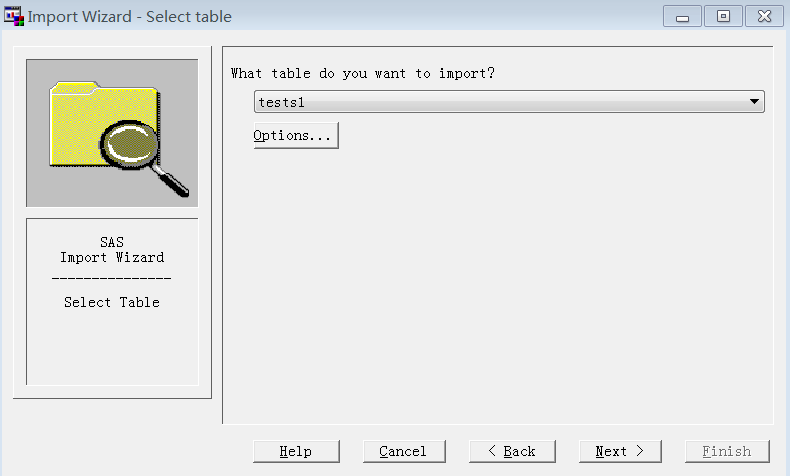
NO——否；

**三、导入向导（Import Wizard）**

1. 点击【文件】——【导入数据】

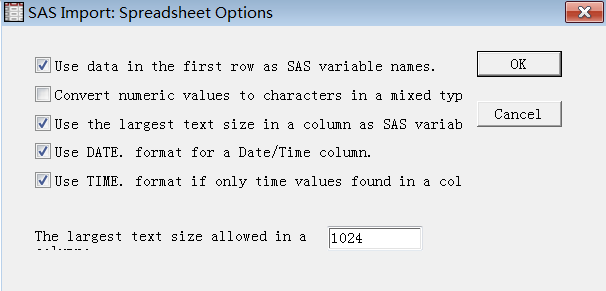


2. 点击Next，点Browse（浏览），打开要导入的Excel文件（exercise.xlsx），点OK.

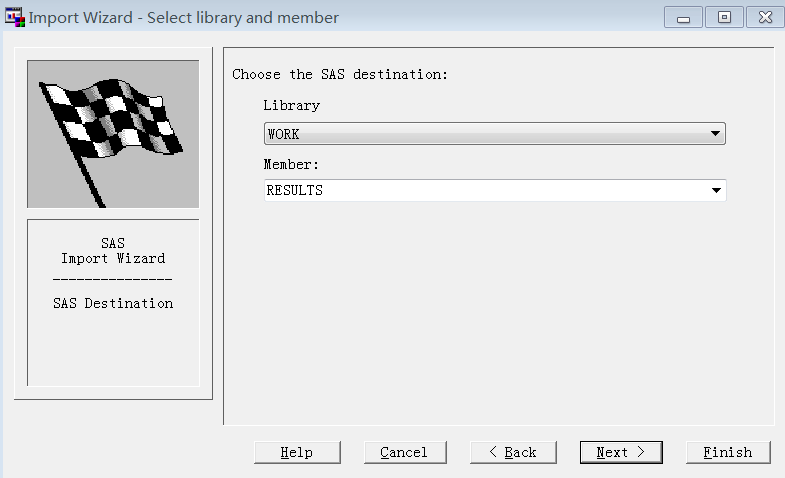


3. 用下拉菜单选择要导入的数据表（tests1），点Options，选择需要的选项，点OK

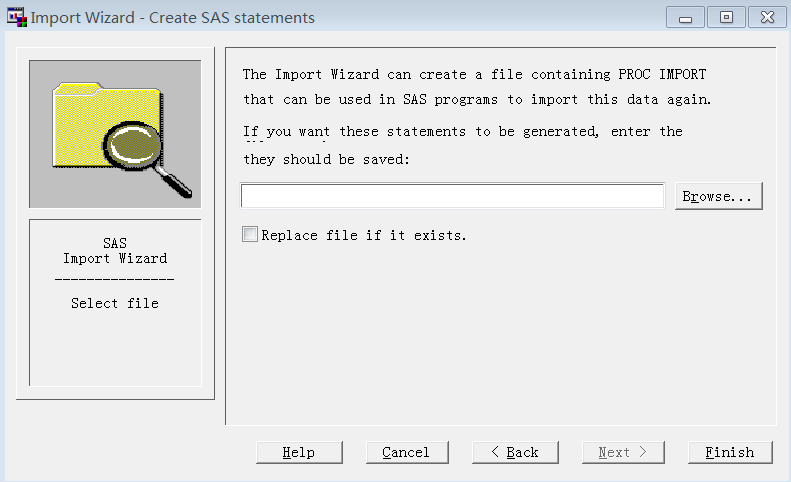
注意：第一行是否从工作表的第一行读取数据集的列变量名。



4.设置输出数据集的逻辑库和数据集名称



5. 生成导入该数据文件的代码，点Browse设置保存路径和文件名，之后点Finish



生成代码如下：

PROC IMPORT OUT= WORK.RESULTS

DATAFILE= "D:\我的文档\My SAS Files\9.3\exercise.xlsx"

DBMS=EXCEL REPLACE;

RANGE="tests1";

GETNAMES=YES;

MIXED=NO;

SCANTEXT=YES;

USEDATE=YES;

SCANTIME=YES;

RUN;

（二）将数据集导出到Excel文件

可以用三种方法：

1. export语句（类似前面的import语句）；

② 数据导出向导（Export Wizard）；

③libname语句

前两种方法有的列没有导出成功。下面只介绍第三种方法：

**例3** 导出数据集Sasuser.Admit和Sasuser.Admit2，分别存入Excel文件“D:\我的文档\My SAS Files\9.3\exercise2.xlsx”的工作表tests1和tests2.

代码：

libname myxlsx EXCEL 'D:\我的文档\My SAS Files\9.3\exercise2.xlsx';

**data** myxlsx.tests1 (dblabel=YES);

set Sasuser.Admit;

**run**;

**data** myxlsx.tests2 (dblabel=YES);

set Sasuser.Admit2;

**run**;

libname myxlsx clear;

**程序说明：**

（1）libname指定引用名myxlsx，引用“导出Excel文件的路径和文件名”；

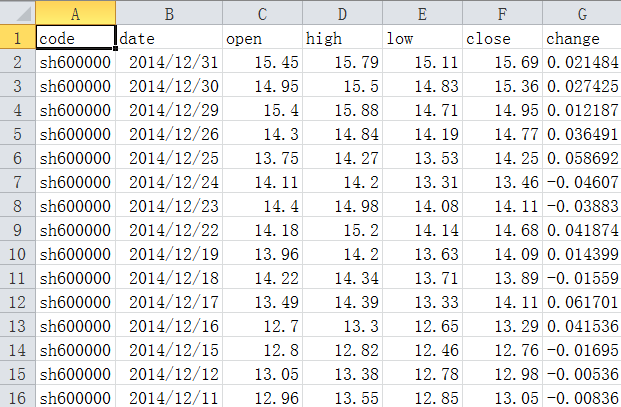
（2） 第一部分data将数据集Sasuser.Admit存入工作表tests1， dblabel=YES表示输出列变量名作为Excel数据的列标签名；

（3）“libname myxlsx clear;”解除文件exercise2.xlsx与SAS的关联。

**（三）导入CSV文件**

CSV文件是用逗号作为分隔符的文本文档，默认用EXCEL打开，显示效果是表格数据，实际上不是表格数据。所以，读入CSV文件时，适用的是固定分隔符数据文件的读入语法。

**例4** 股票数据（C:\MyRawData\sh600000.csv）的一部分如下：



读入数据选取2014年的数据，只保留date、open、high、low、close列，并对date变量按时间正序排序。

代码：

**proc** **import** datafile = 'C:\MyRawData\sh600000.csv' DBMS=CSV OUT = sh600000 REPLACE;

getnames = YES;

datarow = **2**;

**run**;

**data** sh600000\_2014;

set sh600000;

keep date open high low close;

where **'1Jan2014'D** <= date <= **'31Dec2014'D**;

**proc** **sort** data = sh600000\_2014;

by date;

**run**;

**proc** **print** data = sh600000\_2014;

title 'sh600000股票2014年的数据';

**run**;

运行结果：（部分）

