

数学与应用数学专业
SAS 软件与金融数据库课程设计



题目 设计程序用线性插值法填充缺失数据

班级 115010401

姓名 赵赞豪、肖玉、秦江（排序不分先后）

学号 11501040136/11501040132/11501040135

日期 2016 年 12 月 27 日

1. 课程设计评价参考标准及得分（占 70%）

序号	指标	分值	得分
1	内容体现应用数学与金融、经济的结合，体现应用价值和现实意义	30	
2	综合应用数学专业知识解决实际问题的能力	30	
3	与学分相适应的工作量和难度，有一定的创新	20	
4	报告撰写质量：图标美观，参考文献，格式合适等	20	

2. 答辩情况：（占 30%）

--

论文成绩 _____

任课教师签名 _____

目 录

一. 引言	4
二. 程序代码.....	4
三.程序代码分析.....	10
1.建立样本数据库	10
2.拆分样本数据集	10
3.运用宏变量添加缺省值	11
4.用 SAS/IML 过程创建时间数据集.....	12
5.合并时间数据集和被拆分后的数据集	13
6.建立辅助标记数据集	13
7.建立标记数据集	14
8.用 SAS/IML 进行线性插值（核心程序）	14
9.对数据集 FINALDATA 进行加工处理.....	16
10.输出传送系统控制输出	17
四.结论	17
五.参考文献	18

设计程序用线性插值法填充缺失数据

摘要：用线性插值法填充缺失数据的实质是把与插值点靠近的两个数据点用直线连接，然后取平均值，从而可以得到插值点的数据。解决该问题的数学方法是线性插值模型。该模型优点是可以简单快速地对缺失的数据进行插值，缺点是精确度不够，产生的误差较大。总而言之，该模型可以快速地实现对缺失数据的插值处理。

关键词：线性插值法；直线连接；平均值；插值处理。

一. 引言

1.问题的提出：该问题是在债券工作人员不知道一个月某几天的回购利率，知道一个月的某些回购利率。这个时候，债券工作人员想知道一个月所有的回购利率。此时，提出用 SAS 软件编程，利用线性插值的方法对缺失数据进行插值处理。

2.选题背景及意义：背景是在某些债券公司里，某些回购利率没有及时记录，但是工作人员却需要的情况。意义是，如果编程实现对回购利率的近似插值，那么将会提高债券工作人员的决策安全性，同时给投资人一个具体的数据参考，降低投资人的风险。

3.文献综述：本次课程设计参考的图书为《SAS 编程技术教程（第二版）》，本书结构合理，分块设计风格很好，是一本很好的授课和自学教材。

4.研究方法：线性插值法

5.论文结构安排：本篇论文的结构安排是标题并列式安排，程序段是总分安排。

二. 程序代码

```

/* 创建样本数据集 */
data test;
informat Date yymmdd10. code$8. Ir;
label date='日期|date' code='基准利率代码|code' ir='基准利率|ir';
input Date: Code $ Ir@@; format Date yymmdd10. code $8. Ir 6.4;
cards;
2004-04-29 R3M 0.0286 2004-04-30 R1M 0.0265
2004-04-30 R2M 0.0279 2004-05-08 R1M 0.0265
2004-05-09 R1M 0.0264 2004-05-10 R1M 0.0264
2004-05-10 R2M 0.0264 2004-05-10 R3M 0.0290
2004-05-11 R1M 0.0264 2004-05-11 R2M 0.0272
2004-05-12 R1M 0.0260 2004-05-12 R2M 0.0268
2004-05-13 R1M 0.0255 2004-05-14 R1M 0.0260
2004-05-14 R2M 0.0265 2004-05-17 R1M 0.0256
2004-05-17 R2M 0.0265 2004-05-18 R1M 0.0256
2004-05-18 R3M 0.0285 2004-05-19 R1M 0.0256
2004-05-19 R2M 0.0260 2004-05-19 R3M 0.0285
2004-05-20 R1M 0.0265 2004-05-20 R2M 0.0270
2004-05-20 R3M 0.0265 2004-05-21 R1M 0.0260
2004-05-24 R1M 0.0210 2004-05-24 R2M 0.0270
2004-05-24 R3M 0.0285 2004-05-25 R1M 0.0225
2004-05-25 R2M 0.0270 2004-05-26 R1M 0.0263
2004-05-26 R3M 0.0283 2004-05-27 R1M 0.0280
2004-05-27 R2M 0.0270 2004-05-27 R3M 0.0285
2004-05-28 R1M 0.0268 2004-05-31 R1M 0.0288
2004-05-31 R2M 0.0320 2004-06-01 R1M 0.0310
2004-06-01 R2M 0.0319 2004-06-01 R3M 0.0335
;
run;
/*拆分数据集*/
data R1M;
set test;
where code='R1M';
rename ir=r1m; /*对基准利率重命名*/
label ir='1个月回购利率|r1m'; /*对基准利率重置标签*/
run;
data R2M;
set test;
where code='R2M';
rename ir=r2m; /*对基准利率重命名*/
label ir='2个月回购利率|r2m'; /*对基准利率重置标签*/
run;
data R3M;
set test;

```

```

where code='R3M';
rename ir=r3m; /*对基准利率重命名*/
label ir='3个月回购利率|r3m'; /*对基准利率重置标签*/
run;
/*用宏添加缺省值*/
%macro create(varn,view);
data b;
%do i=1 %to &varn; /*生成变量*/
x&i=.; /*设定变量值为缺省值*/
%if _n_=1 %then output b; /*将数据输出至数据集b中*/
%end;
run;
%do j=1 %to &view; /*生成观测*/
proc append base=a data=b; /*创建数据集a, 并且将数据集b的观测添加到数据集a (从数据集a的最后一个观测开始添加) */
run;
%end;
%mend create;
%create(1,200); run; /*利用宏参数创建一个含有1个变量, 200个观测的数据集b, 并将数据集b的观测添加至数据集a中*/
proc iml; /*IML实现题设所需时间 (作为以下步骤合并基准) 的相关创建*/
use a; /*调用数据集a*/
edit a; /*编辑数据集a*/
read all into aa; /*将数据集a中的数值型观测值读入矩阵aa*/
list all; /*在SAS界面中显示*/
do i='01jan04'd to '31may04'd; /*定义循环初始值和终值*/
aa[i-'01jan04'd+1,1]=i; /*从第一个缺省值开始将日期逐渐录入 (从2014年1月1日到2014年5月31日) */
end;
create a1 var{'date'}; /*创建数据集a1, 设置变量date*/
append from aa; /*将矩阵aa中的值填充到数据集a1中*/
quit;
data a2; /*保留时间数据集a1中的非缺省值数据观测*/
set a1;
where date is not missing;
format date yymmdd10.;
run;
data c(drop=code); /*剔除基准利率代码, 并且合并a2, r1m, r2m, r3m到数据集c中*/
merge a2 r1m r2m r3m;
by date;
run;
data d; /*建立月份数据集d, 作为以后判断的依据*/
set c;
month=month(date); /*从SAS日期值得到对应的月份*/

```

```

run;
proc sort data=d out=d1; /*对数据集d排序，排序后的数据集输出到数据集d1中*/
by month; /*按月份排序*/
run;
data d;
set d1;
by month;
if last.month then id=1; /*对每月最后一天的日期标记为1，由于时间观测有六月一号，
在此特别声明在六月一号处也添加标记1，以便后续步骤*/
run;
proc iml; /*用IML进行线插值*/
use d; /*调用数据集d*/
edit d; /*编辑数据集d*/
read all into m; /*将数据集d的观测读入矩阵m中*/
print m; /*显示矩阵m*/
row=nrow(m); /*得到行数*/
col=ncol(m); /*得到列数*/
find all where(id=1) into t; /*将数据集d中每月最后一天的位置输出到位置矩阵t中
*/
print row col t; /*显示行数，列数，以及位置矩阵*/
/*****
*****/
/***** 填充过程1——填充每月月底值 *****/
/*****
*****/
/*****
*****/
do i=1 to nrow(t); /* do1,定位6个每一月最后一天，第六月除外，下同*/
do j=2 to ncol(m)-2; /* do2 ,定位6个月份的最后一天的r1m,r2m,r3m,以便判断
缺省值并进行填充*/
if m[t[i,1],j]=. then do; /* do3, 如果6个月份最后一天的r1m,r2m,r3m其中任
一是缺省值，执行下列操作*/
do c1=t[i,1]-1 to 1 by -1 until(m[c1,j]^=.); /* do4, r1m,r2m,r3m若任一
为缺省值，分别独立在每一月份的范围向上寻找非缺省值，
因为本次向上查找为第一个月份，所以到第一个观测值才停止查找，并且按一个步长逆序查找
*/
if m[c1,j]^=. then do; /* do5, 如果6个月份最后一天的r1m,r2m,r3m均为非缺省值，
执行下列操作*/
do c2=t[i,1]+1 to t[i+1,1] until(m[c2,j]^=.); /* do6 , r1m,r2m,r3m分别
独立向下一个月份寻找，直到找到非缺省值时，该步骤停止*/
if m[c2,j]^=. then do; /* do7, 如果在下一个月份的r1m,r2m,r3m按列分别找到非缺
省值，执行下一步骤 */
m[t[i,1],j]=(m[c1,j]+m[c2,j])/2; /*r1m,r2m,r3m分别取独立向上一个月份的范围，和
向下一个月份的范围查找得到的非缺省值，进行取平均值，
从而填充6个月份每一个月底那天的r1m,r2m,r3m的缺省值（不为缺省值的数据保持原样）*/

```

```

end; /* end7 */
else m[t[i,1],j]=.;/*r1m,r2m,r3m分别独立向下查找，若向下一个月份分别查找，没有找到非缺省值，则该月最后一天的r1m,r2m,r3m就为缺省值*/
end; /* end6 */
end; /* end5 */
else m[t[i,1],j]=.;/*r1m,r2m,r3m分别独立向上查找，若向上在该月范围分别查找，没有找到非缺省值，则该月最后一天的r1m,r2m,r3m就为缺省值*/
end; /* end4 */
end; /* end3 */
end; /* end2 */
end; /* end1 */
print m;/*显示6个月每一个月的最后一天的填充值*/
/*****
*****/
/*****          填充过程2——填充每月其它缺失值
*****/
/*****
*****/
do i=nrow(t) to 1 by -1;/*倒序填充每个月的其它缺失值*/
  do j=2 to ncol(m)-2;/*定位6个月份的除最后一天的其它天数的r1m,r2m,r3m,以便判断缺省值并进行填充*/
    if m[t[i,1],j]^=. then/*如果填充其它天数的基准值（每个月最后一天的数据）为非缺省值，执行下一步操作*/
      do k=t[i,1] to t[i-1,1]+1 by -1;/*设置循环条件，使得后续操作可以逆序执行*/
        if m[k,j]=. then m[k,j]=m[t[i,1],j];/*如果填充其它天数的基准值（每个月最后一天的数据）为缺省值，那么，该月的这列其它天数的填充值为缺省值*/
      end;
    end;
  end;
print m;/*再一次显示矩阵m,此时矩阵m已经经过填充数据处理，等待用户执行下一步操作*/
create finaldata from m;
append from m;/*创建新的数据集finaldata,将矩阵m中的数据（包扣缺省值）全部录入数据集finaldata中*/
quit;
data finaldata;/*加工数据集finaldata*/
set finaldata;
format col1 yymmdd10. col2 6.4 col3 6.4 col4 6.4;
keep col1 col2 col3 col4;
label col1='日期|date'
      col2='一个月回购利率|r1m'
      col3='两个月回购利率|r2m'
      col4='三个月回购利率|r3m';/*对变量的标签进行重命名*/
rename col1=date

```



```

col2=r1m
col3=r2m
col4=r3m; /*对变量进行重命名*/
run;
/*****
*****/
ods pdf file='F:\SASdesign\题9结果.pdf'; /*输出传送系统，控制结果的输出为pdf
文件*/
proc print data=finaldata label;
run;
ods pdf close;
/*****
*****/
ods html file='F:\SASdesign\题9结果.htm'; /*输出传送系统，控制结果的输出为html
文件*/
proc print data=finaldata label;
run;
ods html close;
/*****
*****/
ods rtf file='F:\SASdesign\题9结果.rtf'; /*输出传送系统，控制结果的输出为rtf
文件*/
proc print data=finaldata label;
run;
ods rtf close;

```

三.程序代码分析

1.建立样本数据库

对于样本数据库的建立，我们采取以数据步的方式。如下图所示：

```
/* 创建样本数据集 */
data test;
  informat Date yymmdd10. code$8. Ir;
  label date='日期|date' code='基准利率代码|code' ir='基准利率|ir';
  input Date: Code $ Ir@@; format Date yymmdd10. code $8. Ir 6.4;
  cards;
2004-04-29 R3M 0.0286 2004-04-30 R1M 0.0265
2004-04-30 R2M 0.0279 2004-05-08 R1M 0.0265
2004-05-09 R1M 0.0264 2004-05-10 R1M 0.0264
2004-05-10 R2M 0.0264 2004-05-10 R3M 0.0290
2004-05-11 R1M 0.0264 2004-05-11 R2M 0.0272
2004-05-12 R1M 0.0260 2004-05-12 R2M 0.0268
2004-05-13 R1M 0.0255 2004-05-14 R1M 0.0260
2004-05-14 R2M 0.0265 2004-05-17 R1M 0.0256
2004-05-17 R2M 0.0265 2004-05-18 R1M 0.0256
2004-05-18 R3M 0.0285 2004-05-19 R1M 0.0256
2004-05-19 R2M 0.0260 2004-05-19 R3M 0.0285
2004-05-20 R1M 0.0265 2004-05-20 R2M 0.0270
2004-05-20 R3M 0.0265 2004-05-21 R1M 0.0260
2004-05-24 R1M 0.0210 2004-05-24 R2M 0.0270
2004-05-24 R3M 0.0285 2004-05-25 R1M 0.0225
2004-05-25 R2M 0.0270 2004-05-26 R1M 0.0263
2004-05-26 R3M 0.0283 2004-05-27 R1M 0.0280
2004-05-27 R2M 0.0270 2004-05-27 R3M 0.0285
2004-05-28 R1M 0.0268 2004-05-31 R1M 0.0288
2004-05-31 R2M 0.0320 2004-06-01 R1M 0.0310
2004-06-01 R2M 0.0319 2004-06-01 R3M 0.0335
;
run;
```

图 3-1

2.拆分样本数据集

本步骤是对样本数据集的一种基于标准利率代码的拆分，即对于不同的标准利率代码，分别拆分到三个数据集中，使得被拆分后的三个数据集中的标准利率代码相同。具体程序如下图所示：

```

/*拆分数数据集*/
data R1M;
set test;
where code='R1M';
rename ir=r1m; /*对基准利率重命名*/
label ir='1个月回购利率|r1m'; /*对基准利率重置标签*/
run;

data R2M;
set test;
where code='R2M';
rename ir=r2m; /*对基准利率重命名*/
label ir='2个月回购利率|r2m'; /*对基准利率重置标签*/
run;

data R3M;
set test;
where code='R3M';
rename ir=r3m; /*对基准利率重命名*/
label ir='3个月回购利率|r3m'; /*对基准利率重置标签*/
run;

```

图 3-2

3.运用宏变量添加缺省值

本课程设计运用宏来添加缺省值。首先我们建立一个空的数据集 b，然后用循环生成变量，紧接着将数据输出到数据集 b 中，然后我们再次运用循环生成观测。紧接着创建数据集 a，将数据集 b 中的数据添加到数据集 a 中。循环交互式添加缺省值，程序如下图：

```

/*用宏添加缺省值*/
❏ %macro create(varn,view);
  data b;
  %do i=1 %to &varn; /*生成变量*/
  x&i=.; /*设定变量值为缺省值*/
  %if _n_=1 %then output b; /*将数据输出至数据集b中*/
  %end;
  run;
  %do j=1 %to &view; /*生成观测*/
  proc append base=a data=b;
  /*创建数据集a, 并且将数据集b的观测添加到数据集a
  (从数据集a的最后一个观测开始添加) */
  run;
  %end;
  %mend create;
  %create(1,200); run;
/*利用宏参数创建一个含有1个变量, 200个观测的数据集b,
并将数据集b的观测添加至数据集a中*/

```

图 3-3

4.用 SAS/IML 过程创建时间数据集

用 SAS/IML 过程, 调用数据集 a, 将数据集 a 内所有数值型观测全部输出到矩阵 aa 中, 然后用 do 循环语句将日期从第一个缺省值开始依次录入矩阵 aa 中。紧接着创建数据集 a1, 设置变量 date。然后将矩阵 aa 中的数据填充到数据集 a1 中。

由于数据集 a1 中含有缺省值数据, 那么必须对它进行剔除缺省值处理才可以得到。所以紧接着建立数据集 a2, 将 set 算符将 a1 中的数据导入到 a2 中, 再选取非缺省值的观测保留在数据集 a2 中。具体程序如下图:

```

proc iml; /*IML实现题设所需时间（作为以下步骤合并基准）的相关创建*/
use a; /*调用数据集a*/
edit a; /*编辑数据集a*/
read all into aa; /*将数据集a中的数值型观测值读入矩阵aa*/
list all; /*在SAS界面中显示*/
do i='01jan04'd to '31may04'd; /*定义循环初始值和终值*/
aa[i-'01jan04'd+1,1]=i;
/*从第一个缺省值开始将日期逐渐录入（从2014年1月1日到2014年5月31日）*/
end;
create a1 var{'date'}; /*创建数据集a1, 设置变量date*/
append from aa; /*将矩阵aa中的值填充到数据集a1中*/
quit;

```

图 3-4

5.合并时间数据集和被拆分后的数据集

对于第二步创建的三个数据集 r1m,r2m,r3m,以及第四步创建的时间数据集 a2，完成一项剔除基准利率代码 code 的合并。与此同时，将合并后的数据集放入数据集 c 中。

```

data c(drop=code);
/*剔除基准利率代码，并且合并a2，r1m,r2m,r3m到数据集c中*/
merge a2 r1m r2m r3m;
by date;
run;

```

图 3-5

6.建立辅助标记数据集

建立数据集 d，导入数据集 c，在数据集 c 的基础上增加一个变量 month，用 month 函数从观测的 SAS 日期之中取得对应的月份。然后对数据及 d 按照变量 month 进行排序，将排序后的结果输出到已建立的数据集 d1 中。以便于标记数据集的建立。具体程序如下图：

```

data d;
  /*建立月份数据集d, 作为以后判断的依据*/
  set c;
  month=month(date);
  /*从SAS日期值得到对应的月份*/
run;
proc sort data=d out=d1;
  /*对数据集d排序, 排序后的数据集输出到数据集d1中*/
  by month;
  /*按月份排序*/
run;

```

图 3-6

7.建立标记数据集

建立数据集 d, 将数据集 d1 的数据导入 d 中, 经过排序, 然后用 if 语句判定 month 变量值是否为最后一个。若为最后一个, 则标记为 1。此时数据集 d 中有 id 变量, 除每个月最后一天为标记 1 外, 其他位置为缺省值。具体程序如下:

```

data d;
  set d1;
  by month;
  if last.month then id=1;
  /*对每月最后一天的日期标记为1,
  由于时间观测有六月一号, 在此特别声明在六月一号处也添加标记1, 以便后续步骤*/
run;

```

图 3-7

8.用 SAS/IML 进行线性插值（核心程序）

第一步: 调用标记数据集 d, 将数据集 d 的观测读入到矩阵 m 中, 用 nrow 函数和 ncol 函数求得矩阵 m 的行数与列数, 同时用 find 语句查找到有标记 1 的位置 (即带有标记的观测行的行数) 并且输入到矩阵 t 中。然后显示矩阵 m 的行数与列数, 以及位置矩阵 t。具体程序如

下图：

```
proc iml; /*用IML进行线性插值*/
use d; /*调用数据集d*/
edit d; /*编辑数据集d*/
read all into m; /*将数据集d的观测读入矩阵m中*/
print m; /*显示矩阵m*/
row=nrow(m); /*得到行数*/
col=ncol(m); /*得到列数*/
find all where(id=1) into t; /*将数据集d中每月最后一天的位置输出到位置矩阵t中*/
print row col t; /*显示行数，列数，以及位置矩阵*/
```

图 3-81

第二步：进行填充过程一，填充每一月的月底值。

具体程序如下图（附有详细解释说明）所示：

```
/******
*****          填充过程1——填充每月月底值          *****
*****
do i=1 to nrow(t); /* do1, 定位6个每一月最后一天，第六月除外，下同*/
  do j=2 to ncol(m)-2; /* do2, 定位6个月份的最后一天的r1m, r2m, r3m, 以便判断缺省值并进行填充*/
    if m[t[i,1],j]=. then do; /* do3, 如果6个月份最后一天的r1m, r2m, r3m其中任一为缺省值，执行下列操作*/
      do c1=t[i,1]-1 to 1 by -1 until(m[c1,j] ^=.); /* do4, r1m, r2m, r3m若任一为缺省值，分别独立在每一月份的范围向上寻找非缺省值，
      因为本次向上查找为第一个月份，所以到第一个观测值才停止查找，并且按一个步长逆序查找*/
      if m[c1,j] ^=. then do; /* do5, 如果6个月份最后一天的r1m, r2m, r3m均为非缺省值，执行下列操作*/
        do c2=t[i,1]+1 to t[i+1,1] until(m[c2,j] ^=.); /* do6, r1m, r2m, r3m分别独立向下一个月份寻找，直到找到非缺省值时，该步骤停止*/
        if m[c2,j] ^=. then do; /* do7, 如果在下一个月份的r1m, r2m, r3m按列分别找到非缺省值，执行下一步骤 */
          m[t[i,1],j]=(m[c1,j]+m[c2,j])/2; /*r1m, r2m, r3m分别取独立向上一个月的范围，和向下一个月份的范围查找得到的非缺省值，进行取平均值，
          从而填充6个月份每一个月月底那天的r1m, r2m, r3m的缺省值（不为缺省值的数据保持原样）*/
        end; /* end7 */
      else m[t[i,1],j]=.; /*r1m, r2m, r3m分别独立向下查找，若向下一个月份分别查找，没有找到非缺省值，则该月最后一天的r1m, r2m, r3m就为缺省值*/
      end; /* end6 */
    end; /* end5 */
  else m[t[i,1],j]=.; /*r1m, r2m, r3m分别独立向上查找，若向上在该月范围分别查找，没有找到非缺省值，则该月最后一天的r1m, r2m, r3m就为缺省值*/
  end; /* end4 */
end; /* end3 */
end; /* end2 */
end; /* end1 */
print m; /*显示6个月每一月的最后一天的填充值*/
```

图 3-82

第三步：进行填充过程二，填充每一个月除月底之外的其他值。

具体程序如下图（附有详细解释说明）所示：

```

/*****
*****          填充过程2——填充每月其它缺失值          *****
*****/
do i=nrow(t) to 1 by -1; /*倒序填充每个月的其它缺失值*/
  do j=2 to ncol(m)-2; /*定位6个月份的除最后一天的其它天数的r1m, r2m, r3m, 以便判断缺省值并进行填充*/
    if m[t[i,1],j] ^= . then /*如果填充其它天数的基准值（每个月最后一天的数据）为非缺省值，执行下一步操作*/
      do k=t[i,1] to t[i-1,1]+1 by -1; /*设置循环条件，使得后续操作可以逆序执行*/
        if m[k,j] = . then m[k,j]=m[t[i,1],j]; /*如果填充其它天数的基准值（每个月最后一天的数据）为缺省值，那么，该月的这列其它天数的填充值为缺省值*/
      end;
    end;
  end;
end;

```

图 3-83

第四步：再一次显示矩阵 `m`，创建数据集 `finaldata`（其 `size` 和矩阵 `m` 相同），并且将矩阵 `m` 中的数据输出到数据集 `finaldata` 中，具体程序如下图：

```

print m; /*再一次显示矩阵m, 此时矩阵m已经经过填充数据处理，等待用户执行下一步操作*/
create finaldata from m;
append from m; /*创建新的数据集finaldata, 将矩阵m中的数据（包括缺省值）全部录入数据集finaldata中*/
quit;

```

图 3-84

9.对数据集 FINALDATA 进行加工处理

导入数据集 `finaldata`，保留前四个变量，然后对其变量的标签进行重命名。紧接着对变量进行重命名。最终得到我们想要的数据集 `finaldata`。具体程序如下图所示：


```

data finaldata; /*加工数据集finaldata*/
set finaldata;
format col1 yymmdd10. col2 6.4 col3 6.4 col4 6.4;
keep col1 col2 col3 col4;
label col1='日期|date'
      col2='一个月回购利率|r1m'
      col3='两个月回购利率|r2m'
      col4='三个月回购利率|r3m' /*对变量的标签进行重命名*/
rename col1=date
       col2=r1m
       col3=r2m
       col4=r3m; /*对变量进行重命名*/
run;

```

图 3-85

10. 输出传送系统控制输出

对于最后得到的数据集的输出，我们采用输出传送系统控制输出。在此次课程设计中，我们采用三种不同的输出文件输出。具体程序如图所示：

```

/*****
ods pdf file='F:\SASdesign\题9结果.pdf'; /*输出传送系统，控制结果的输出为pdf文件*/
proc print data=finaldata label;
run;
ods pdf close;
/*****
ods html file='F:\SASdesign\题9结果.htm'; /*输出传送系统，控制结果的输出为html文件*/
proc print data=finaldata label;
run;
ods html close;
/*****
ods rtf file='F:\SASdesign\题9结果.rtf'; /*输出传送系统，控制结果的输出为rtf文件*/
proc print data=finaldata label;
run;
ods rtf close;

```

图 3-86

四. 结论

用线性插值的方法可以快速插入多个缺失值，但是会有一定的误差。

五.参考文献

参考文献:

[1] 朱世武.SAS 编程技术教程（第 2 版）[M].北京:清华大学出版社,2013:7-481.

[附录一](#): PDF 文件

[附录二](#): HTML 文件

[附录三](#): DOC 文件, RTF 格式