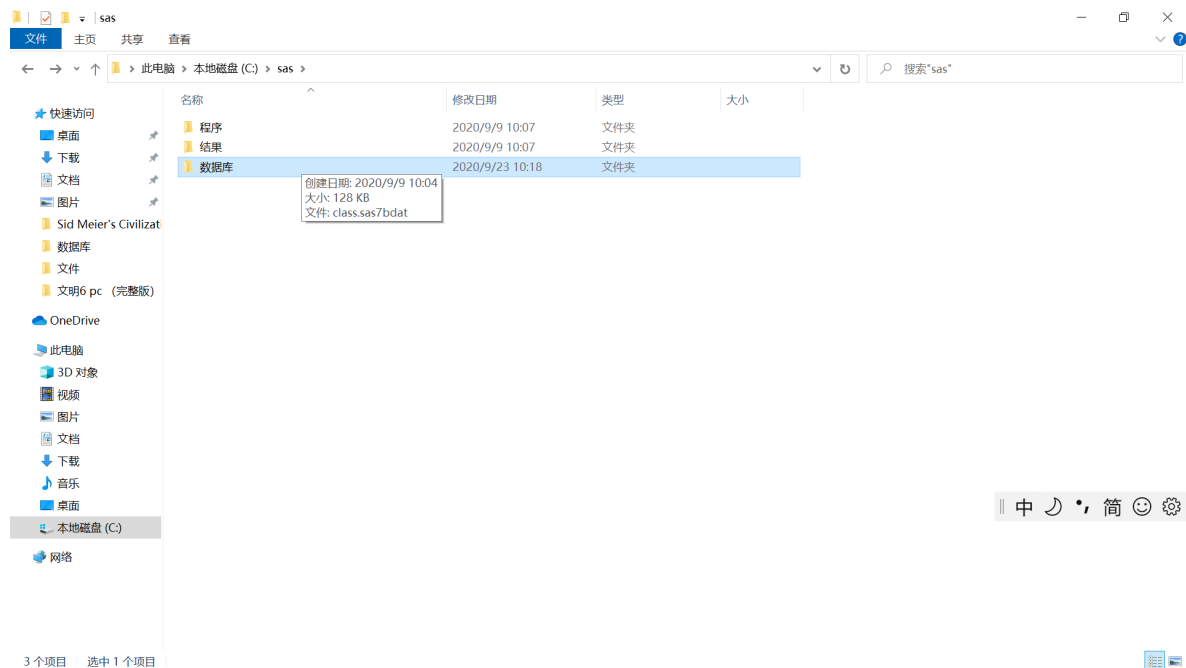


实验1 SAS基础(一)

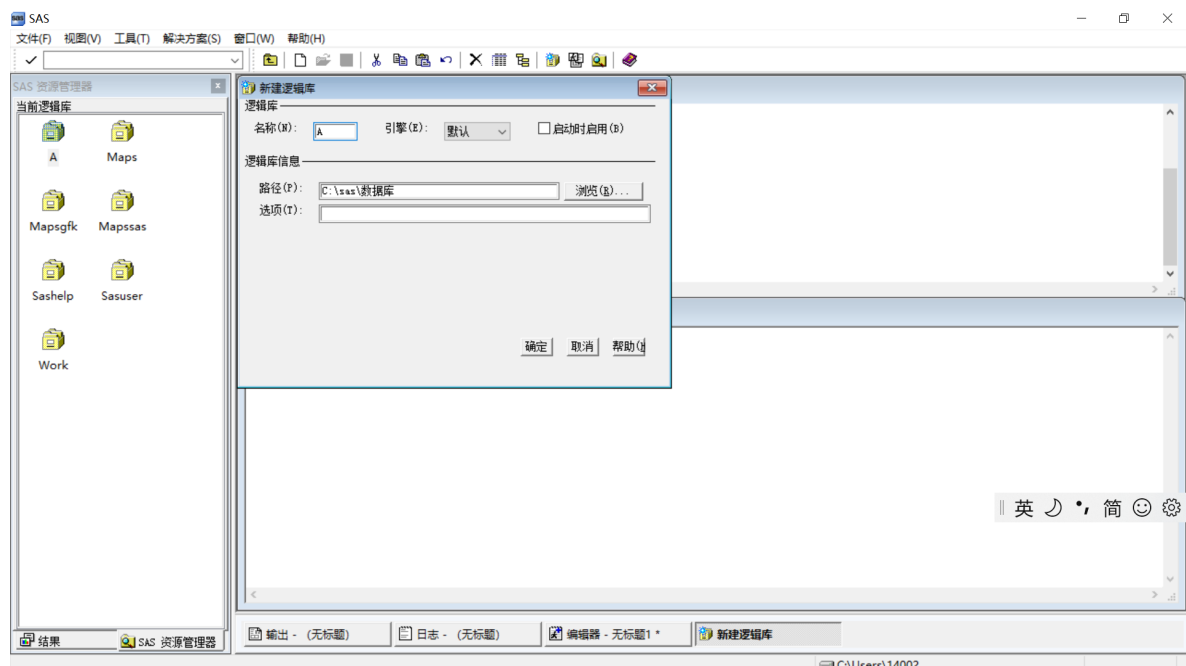
承子杰 1807402074

2.SAS库的建立与使用(自定),并将sashelp库下的数据集class存放到自己库中。

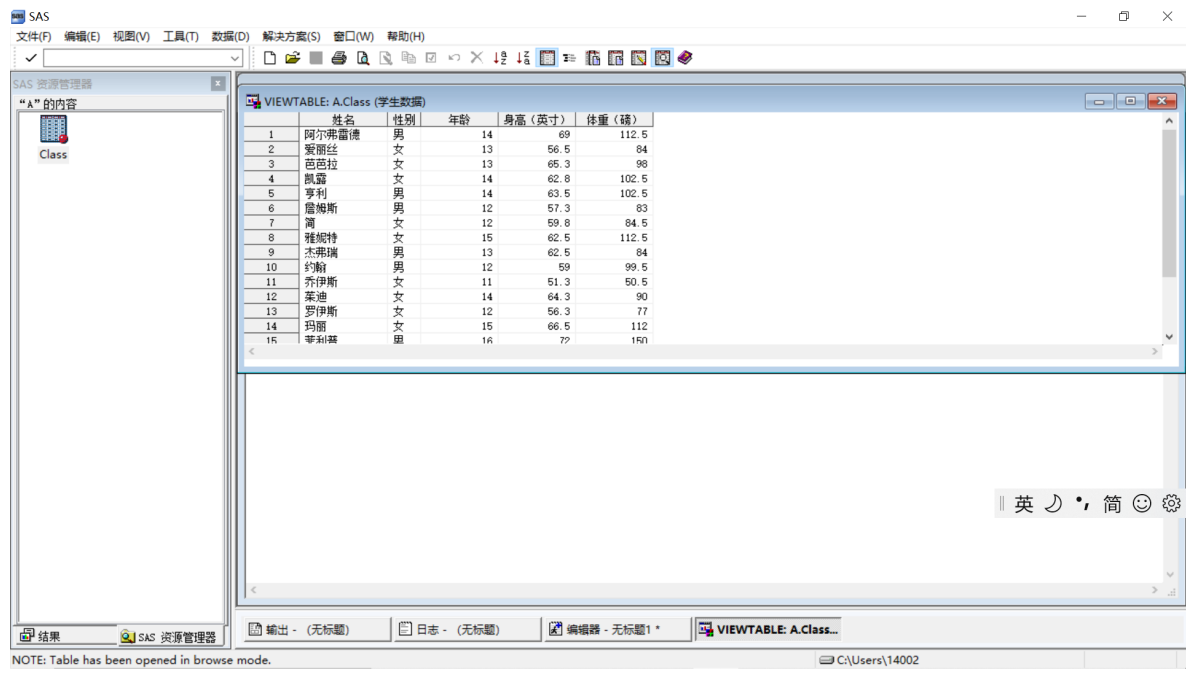
步骤: STEP1 先在C盘中建立一个文件夹



STEP2 然后再SAS界面中创建一个自己的数据库，引用路径为上面建立数据库的文件位置，我在这里给数据库起名为A。



STEP3 找到sashelp.class中的文件，右击复制，再粘贴到逻辑库A中



3.产生正态分布随机数，计算正态分布上0.05，0.025，0.01分位数

分析：可以调用rannor (seed, x) 函数来生成正态分布随机数，对于正态分布的分位数可以采用PROBIT(p)函数来调用。

代码与结果：生成正态分布随机数代码如下：

```

1  data a,b;
2  seed=2;
3  do i=1 to 10;
4  call rannor(seed,x);
5  *采用种子为2，生存10个随机数
6  output ;
7  end;
8  run;
9
10 proc print data=a;
11 run;

```

结果如下：

SAS 系统

观测	seed	i	x
1	2019015659	1	1.31183
2	1114108698	2	-0.66405
3	2015530869	3	0.54104
4	136114913	4	2.04198
5	285901162	5	1.43923
6	102523109	6	0.90473
7	288555560	7	0.55341
8	1276438356	8	-0.35157
9	815741553	9	-0.80223
10	972842384	10	-0.29515

生成各分位数的代码如下：

```
1 data a;  
2 input x;  
3 x_p=PROBIT(x);  
4 *正态分布分位数函数  
5 datalines;  
6 0.05  
7 0.025  
8 0.01  
9 run;  
10  
11 proc print data=a;  
12 run;
```

结果如下：

VIEWTABLE: Work.A		
	x	x_p
1	0.05	-1.644853627
2	0.025	-1.959963985
3	0.01	-2.326347874

总结： 正态分布的0.05分位数是 -1.64，0.025的分位数为-1.96，0.01的分位数为 -2.33

4.产生标准正态分布表（计算正态分布的概率）

分析：对于正态分布的概率，可以采用PROBNORM (x) 函数 ($P(X \leq x)$)

代码与结果：

```
1 data a;  
2 do x=-4 to 4;  
3 p_x=PROBNORM(x);  
4 *正态分布概率计算函数  
5 output;  
6 end;  
7 run;  
8  
9 proc print data=a;  
10 run;
```

结果如下：

SAS 系统		
观测	x	p_x
1	-4	0.00003
2	-3	0.00135
3	-2	0.02275
4	-1	0.15866
5	0	0.50000
6	1	0.84134
7	2	0.97725
8	3	0.99865
9	4	0.99997

总结：考虑到正态分布的 3σ 原则，故我们只生成了-4到4的正态分布概率。

5. SAS数据集的整理（增加、修改变量；计数变量）：新建数据集，对sashelp.class中的变量进行如下操作：（1）使变量weight值加10。（2）增加变量 **$bmi = weight/height^2$** 。（3）统计数据集中观测个数（用累计变量的形式），计算总体重（用累加变量的形式）

分析：在2中建立的逻辑库A中新建立一个数据集result，拷贝A.class中的数据。

对于 (1) , 只需直接引用 $weight$ 变量, 执行 $weight = weight + 10$ 语句。

对于 (2) , 只需引入一个新变量 bmi , 执行 $bmi = weight / (height * * 2)$ 语句。

对于 (3) , 只需引入一个累计变量 i , 执行 $i + 1$ 语句。

对于 (4) , 只需引入一个总体重变量 Sum_Weight , 执行 $Sum_Weight + weight$ 语句。

代码与结果:

```
1 data a.result;  
2 set a.class;  
3 * a.result 拷贝 a.class 中的数据  
4 weight=weight+10;  
5 *引用 weight 变量, 并加10  
6 bmi=weight/(height**2);  
7 *创建 bmi 变量, bmi = weight / (height ** 2)  
8 format bmi 5.3;  
9 *控制 bmi 格式 5个位宽, 小数点后3位, 整数1位  
10 i+1;  
11 *累加变量i, 记录观测数量  
12 Sum_weight+weight;  
13 *体重累加求和  
14 run;  
15  
16 proc print data=a.result;  
17 *输出结果  
18 run;
```

结果如下图:

SAS 系统

观测	Name	Sex	Age	Height	Weight	bmi	i	Sum_Weight
1	阿尔弗雷德	男	14	69.0	122.5	0.026	1	122.5
2	爱丽丝	女	13	56.5	94.0	0.029	2	216.5
3	芭芭拉	女	13	65.3	108.0	0.025	3	324.5
4	凯露	女	14	62.8	112.5	0.029	4	437.0
5	亨利	男	14	63.5	112.5	0.028	5	549.5
6	詹姆斯	男	12	57.3	93.0	0.028	6	642.5
7	简	女	12	59.8	94.5	0.026	7	737.0
8	雅妮特	女	15	62.5	122.5	0.031	8	859.5
9	杰弗瑞	男	13	62.5	94.0	0.024	9	953.5
10	约翰	男	12	59.0	109.5	0.031	10	1063.0
11	乔伊斯	女	11	51.3	60.5	0.023	11	1123.5
12	茱迪	女	14	64.3	100.0	0.024	12	1223.5
13	罗伊斯	女	12	56.3	87.0	0.027	13	1310.5
14	玛丽	女	15	66.5	122.0	0.028	14	1432.5
15	菲利普	男	16	72.0	160.0	0.031	15	1592.5
16	罗伯特	男	12	64.8	138.0	0.033	16	1730.5
17	罗纳德	男	15	67.0	143.0	0.032	17	1873.5
18	托马斯	男	11	57.5	95.0	0.029	18	1968.5
19	威廉	男	15	66.5	122.0	0.028	19	2090.5

总结:考虑到 *bmi* 为小数, 故在此做了格式设定, 保留了三位小数。

6. 自己编程举例, 使得程序中包括不同形式的条件语句、三种循环语句

分析: 仍旧新建数据集, 采用 `sashelp.class` 中的数据进行实验。

对于 *IF* 语句, 我们进行如下操作, 将男性的观测分别输出到数据集 *M* 中。

对于 *IF-THEN* 语句, 我们做如下操作, 男生的身高加10英尺, 女生的体重减10磅

对于 *SELECT-WHEN-OTHERWISE* 语句, 我们做如下操作, 年龄小于13岁的 *bmi* 加 0.01, 年龄大于14岁的 *bmi* 减0.01

最后对 *DO-END*, *DO-WHILE-END* 和 *DO-UNTIL-END* 单独进行试验, 通过三种方法分别对100以内的整数求和。

代码与结果:

```
1 data a.m;  
2 set a.class;  
3 if(Sex='男');
```

```

4  *删选出性别为男的记录
5  run;
6
7  proc print data=a.m;
8  *打印结果
9  run;
10
11 data a.result;
12 set a.class;
13 if(Sex='男') then
14     height=height+10;
15     *性别为男的身高加10
16 else
17     weight=weight-10;
18     *性别为女的体重-10
19 bmi=weight/(height**2);
20 format bmi 5.3;
21 select(age);
22     when(11,12) bmi=bmi+0.01;
23     *年龄为11, 12岁的bmi+0.01
24     when(15,16) bmi=bmi-0.01;
25     *年龄为15, 16岁的bmi-0.01
26     otherwise bmi=bmi;
27     *其余的bmi不变
28 end;
29
30 proc print data=a.result;
31 *打印结果
32 run;

```

结果:

男性观测如下:

SAS 系统

观测	Name	Sex	Age	Height	Weight
1	阿尔弗雷德	男	14	69.0	112.5
2	亨利	男	14	63.5	102.5
3	詹姆斯	男	12	57.3	83.0
4	杰弗瑞	男	13	62.5	84.0
5	约翰	男	12	59.0	99.5
6	菲利普	男	16	72.0	150.0
7	罗伯特	男	12	64.8	128.0
8	罗纳德	男	15	67.0	133.0
9	托马斯	男	11	57.5	85.0
10	威廉	男	15	66.5	112.0

Result数据集操作后结果如下：

SAS 系统

观测	Name	Sex	Age	Height	Weight	bmi
1	阿尔弗雷德	男	14	79.0	112.5	0.018
2	爱丽丝	女	13	56.5	74.0	0.023
3	芭芭拉	女	13	65.3	88.0	0.021
4	凯露	女	14	62.8	92.5	0.023
5	亨利	男	14	73.5	102.5	0.019
6	詹姆斯	男	12	67.3	83.0	0.028
7	简	女	12	59.8	74.5	0.031
8	雅妮特	女	15	62.5	102.5	0.016
9	杰弗瑞	男	13	72.5	84.0	0.016
10	约翰	男	12	69.0	99.5	0.031
11	乔伊斯	女	11	51.3	40.5	0.025
12	茱迪	女	14	64.3	80.0	0.019
13	罗伊斯	女	12	56.3	67.0	0.031
14	玛丽	女	15	66.5	102.0	0.013
15	菲利普	男	16	82.0	150.0	0.012
16	罗伯特	男	12	74.8	128.0	0.033
17	罗纳德	男	15	77.0	133.0	0.012
18	托马斯	男	11	67.5	85.0	0.029
19	威廉	男	15	76.5	112.0	0.009

求和代码如下：

```
1 data Sum;
2 sum1=0;
3 sum2=0;
4 sum3=0;
5 *初始化总和变量
6 do i=1 to 100;
7     sum1=sum1+i;
8 end;
9 j=0;
10 do while(j<=100);
11     sum2=sum2+j;
12     j=j+1;
13 end;
14 k=0;
15 do until(k>100);
16     sum3=sum3+k;
17     k=k+1;
18 end;
19 drop i j k;
20 *删除 i j k变量
21 run;
22
23 proc print data=Sum;
```



```
24 *输出结果
25 run;
```

结果如下：

SAS 系统

观测	sum1	sum2	sum3
1	5050	5050	5050