



# SAS PROGRAM

- 1.数据的导入导出
- 2.描述性统计
- 3.随机数
- 4.统计分布
- 5.宏
- 6.假设检验（参数检验、非参数检验）
- 7.相关分析（Pearson、Spearman、Kendall-Tau）
- 8.方差分析
- 9.作图
- 10.回归分析
- 11.聚类分析
- 12.SQL
- 13.数据清洗（缺失值、重复值）
- 14.数字游戏、算法
- 15.IML
- 16.First、Last指针
- 17.数据集合并

## 1.数据的导入导出

☐ 数据导入

```
/直接导入
data score;
input num four six;
cards;
1 98 90
2 85 82
3 89 88
;
run;
```

```

/代码导入
PROC IMPORT OUT= WORK.air
  DATAFILE= "D:\2018统计计算试卷.xls"
  DBMS=EXCEL REPLACE;
  RANGE="air$";
  GETNAMES=YES;
RUN;

/带分隔符的数据的导入/
proc import out = temp
  file = "c:\sasdata\temp.txt"
  dbms = dlm replace;
  delimiter = '$';
  getnames = yes;
  guessingrows = 30;
run;
/带空格的数据/
proc import out = x;
  datafile = "";
  dbms = tab replace;
  getnames = yes;
  datarow = 2;
run;

```

## ☐ 数据导出

```

将数据集导出到txt文件中。
data _null_;
  set sashelp.class;
  file 'c:\sasdata\a.txt';
  put name sex age;
run;

/利用ods
ods listing file = 'c:\sasdata\sasprint.txt';
proc means data =pilots;
  var salary;
  class Jobcode;
run;
ods listing;

/利用export
proc export data = work.xshort
  outfile = "";
  dbms = tab replace;
run;

```

## ☐ 库的建立

```
libname xh1854305 "c:\sasdata";
```

## 2.描述性统计

### ☐ 相关系数、散点图

```
proc corr data = score;  
    var six with four;  
run;  
/或许不用加with/  
proc corr data = score plots = matrix();  
    var four six;  
run;
```

### ☐ 均值、标准差、偏度、峰度、变异系数、中位数、均值的95%置信区间、四分位间距、标准误。

```
proc means data = score mean std skewness kurtosis cv等等;  
    var four six;  
run;  
mean--均值  
std--标准差  
skewness--偏度  
kurtosis--峰度  
median--中位数  
q1--四分位下限  
q3--四分位上限  
stderr--标准误  
cv--变异系数  
clm alpha=0.05--均值的95%置信区间
```

### ☐ 盒形图、茎叶图

```
/盒形图/  
proc univariate data = score plot;  
    var four six;  
run;  
/茎叶图/  
ods graphics off;  
ods select Plots SSPlots;  
proc univariate data = score plots;  
    var four six;  
run;
```

### ☐ 直方图、分布图、密度估计（参数、非参数）

```
proc sgplot data = sasuser.cars  
    histogram midprice;  
    density midprice;
```

```
density midprice / type = kernel;
run;
```

## □ tabulate过程

```
proc tabulate data = class;
  var height weight;
  table height * mean weight * mean; /能不能写成(height weight) * mean/
  where sex = 'M';
  where same and age <= 13;
  title 'blabla';
run;
```

/例子，输出报表反应男女生数学成绩总和/

```
proc tabulate data = score;
  class = sex;
  var = math;
  table sex,math;
run;
```

/例子，输出报表反应男女生数学语文成绩的平均值与方差/

```
proc tabulate data = score;
  class = sex;
  var = math chinese;
  table sex,(math chinese) * (mean var);
run;
```

/例子，使用统计量N与PCTN来计算各个性别的人数和百分比/

```
proc tabulate data = score;
  class = sex;
  table sex * (N PCTN);
run;
```

/对不同的region和subsidiary进行汇总/

```
proc tabulate data = sashelp.shoes;
  class region subsidiary;
  var sales;
  table region * subsidiary,sales;
run;
```

/对不同的product进行汇总/

```
proc tabulate data = sashelp.shoes;
  class product;
  var sales;
  table product * region,sales;
run;
```

/对不同的产品在不同region的销售额进行交叉汇总/

```
proc tabulate data = sashelp.shoes;
  class product region;
```

```
var sales;
table product * region, sales;
run;
```

## □ 增长率的问题

```
data forecast;
  set growth;
  year=1;
  total_employees = total_employees*(1+increase);
  output;
  year=2;
  total_employees = total_employees*(1+increase);
  output;
run;
```

## 3.随机数

### □ 随机数的产生

```
/例子，利用平方取中法求0.7144后的64个0-1上均匀分布的随机数/
data _null_;
  file print;
  array union{101} union1-union101;
  union{1} = 0.7144;
  do i = 1 to 100;
    union{i+1} = int((union{i}**2*100-int(union{i} ** 2 * 10000)) *10000 )
      / 10000;
    put union{i} =
  run;
```

/例子，利用平方取中法0.7239后的96个服从均匀分布的为随机数。

/并由此生成服从泊松分布P(2)的随机数96个。

/生成服从正态分布N(3,15)的随机数8个。

/生成服从指数分布E(0.2)的随机数96个。

```
data _null_;
  file print;
  array union{97} union1-union97;
  array expo{96} expo1-expo96;
  array norm{8} norm1-norm8;
  union{1} = 0.7396;
  do i = 1 to 8;
    norm{i} = poisson(2,i-1);
  end;
  do i = 1 to 96;
    union{i+1} = int( (union{i}**2*100 - int(union{i}**2*100))*10000 )/10000;
    expo{i} = -log(union{i})/0.2;
    put union{i}= expo{i}= @@;
  select;
    when (union{i}<=norm{1}) put "poisson = 0";
    when (union{i}<=norm{2}) put "poisson = 1";
```

```

        when (union{i}<=norm{3}) put "poisson = 2";
        when (union{i}<=norm{4}) put "poisson = 3";
        when (union{i}<=norm{5}) put "poisson = 4";
        when (union{i}<=norm{5}) put "poisson = 5";
        when (union{i}<=norm{5}) put "poisson = 6";
        when (union{i}<=norm{5}) put "poisson = 7";
        otherwise put "poisson = 8";
    end;
end;
do i = 1 to 8;
    no = 0;
    do j = 1 to 6;
        no = union{j+(i-1)*12}-union(j+(i-1)*12+6);
    end;
    norm{i} = no * sqrt(15) + 3;
    put norm{i}= ;
end;
run;

```

☐ 抽取随机数（从1到100中随机不放回的抽取20个数）

```

data original;
    do i = 1 to 100;
        output;
    end;
run;

proc surveyselect data = original out = rn method = srs samplesize = 20;
run;

```

## 4.统计分布

☐ T分布分位表

```

proc iml;
    b = do(1,20,1);
    a = {0.025,0.05,0.1,0.25,0.5,0.75,0.9,0.95,0.975};
    aa = shape(a,20,9);
    bb = shape(1,20,9);
    do i = 2 to 20;
        bb[i,] = i * bb[1,]
    end;
    t_inv = tinv(aa,bb);
    rown = char(b);
    coln = char(a);
    print t_inv[rowname = rown colname=coln];
quit;

```

☐ 正态分布分位表

```

proc iml;
  a = do(0,3,0.01);
  b = shape(a,30,10);
  row = do(0,2.9,0.1);
  col = do(0,0.9,0.01);
  normal = probnorm(b);
  rown=char(row);
  coln=char(col);
  print normal[rowname=rown colname=coln];
quit;

```

## □ 分布检验

```

/表中数字是否服从泊松分布

data poisson;/导入数据/
run;

/计算频数,将结果输出为poi_freq/
proc freq data = poisson noprint;
  tables number / out = poi_freq;
run;

/合并过小频数类别/
data poi_nfreq;
  set poi_freq;
  if number < 9 then output;
  if number = 9 then do;
    count = 6;
    output;
  end;
run;

/求均值/
proc sql noprint;
  select mean(number) into: mean from poisson;
quit;

/计算概率表/
data poi_freq;
  set poi_nfreq;
  if number = 1 then prob = poisson(&mean.,number);
  else if number = 9 then prob = 1 - poisson(&mean.,number-1);
  else prob = poisson(&mean.,number) - poisson(&mean.,number-1);
run;

/将prob合并成一个字符串prob/
proc sql noprint;
  select prob into:prob separated by ' ' from poi_freq;
quit;

/卡方检验/
proc freq data = poi_freq;
  table number / chisq nocum testp=(&prob.);

```

```

weight count;
run;
quit;

```

## 5.宏

```

/花店的题/
data flower;
    input ID $ date mmddyy10. kind $ num;
cards;
240W 02-07-2003 Ginger 120
240W 02-07-2003 Protea 180
356W 02-08-2003 Heliconia
60 356W 02-08-2003 Anthurium 300
188R 02-11-2003 Ginger 24
188R 02-11-2003 Anthurium 24
240W 02-12-2003 Heliconia 48
240W 02-12-2003 Protea 48
356W 02-12-2003 Ginger 240
;
run;

/用宏找出任意种类的花
%macro select(kind);
proc sql;
title '&kind blabla';
select * from flower where kind = '&kind';
quit;
%mend;
%select(Ginger);

/用宏和sql找出每种花订单最大的客户。
proc sql;
    select id, variety, quantity into:maxid1-:maxid4,:flower1-:flower4,:number1-:number4
    from flower group by variety having quantity = max(quantity);
quit;
%macro maxorder(a);
proc sql;
    title1"";
    title2"";
    select * from flower where id = "&&maxid&a";
quit;
title1;
title2;
%mend maxorder;
%macro calc;
    %do i = 1 %to 4;
        %maxorder(&i);
    %end;
%mend calc;
%calc;

/生成周x的报告
%macro salesrep;

```



```

proc sql;
  title 'Monday Report:Current Flower Sales';
  select * from flower;
quit;
title;
%mend salesrep;

%macro meansrep;
  proc means data = flower;
    var quantity;
    class variety;
    title 'Thursday Report: Summary of Flower Sales';
  run;
  title;
%mend meansrep;

%macro test;
  %let x = %sysfunc(date());
  %let a = %sysfunc((weekday(&x)));
  %if &a = 2 %then %salesrep;
  %if &a = 5 %then %meansrep;
%mend test;

%test;

/用宏判断字符型or数值型
%macro test(in);
data _null_;
  temp = 1;
  temp = "&in";
  if _error_ = 1 then put "Char!";
  else put "Numeric!";
run;
%mend test;

```

## □ 上机1-5题

```

%macro unique(var);
proc sql;
  create table &var as
  select distinct &var from temp;
quit;
%mend;
proc contents data = temp out = attribute(keep=name) noprint;
run;
proc sql noprint;
  select name into :varname separated by ' ' from attribute;
quit;

data _null_;
  n = countc("&varname.", ' ');

```

```

do i = 1 to n;
  var = scan("&varname.",i,' ');
  call execute('%unique' || var || ';');
end;
run;

```

## 6.假设检验

### ☐ 卡方独立性检验(freq)

```

H0:两种疗法效果相同 H1:两种疗法效果不同
data cure;
  do i = 1 to 2;
    if i = 1 then cure = "valid";
    else cure = "invalid";
    do j = 1 to 2;
      if j = 1 then method = "jia";
      else method = "yi";
      input w @@;
      output;
    end;
  end;
cards;
25 29 6 3
;
run;
proc freq data = cure;
  table method * cure / chisq nopercnt nocol norow;
  weight w;
run;

```

### ☐ t检验(ttest)

```

/利用ttest/
proc ttest data = normal h0=5;
  var normal;
run;

/自己编写程序/
proc means data = normal;
  var normal;
  output out = nor_stat;
run;

data _null_;
  set nor_stat;
  if _stat_ = 'N' then call symput('n',normal);
  if _stat_ = 'MEAN' then call symput('mean',normal);
run;

data _null_;

```

```

file print;
u = (&mean - 5)*sqrt(&n) / 1;
pvalue = 2 * (1-probnorm(abs(u)));
put 'u-value=' u 'p-value=' p;
if pvalue < 0.05 then put 'Reject the null hypothesis';
else put 'Can not reject the null hypothesis';
run;

```

## ☐ Wilcoxon-Mann-Whitney检验

```

data weight(drop = i);
do i = 1 to 8;
    group = "jia";
    input weight @@;
    output;
end;
do i = 1 to 13;
    group = "yi";
    input weight @@;
    output;
end;
cards;
blabla
;
proc NPAR1WAY wilcoxon;
    class group;
    var weight;
run;

```

## ☐ Wilcoxon符号秩检验

```

proc univariate data = s;
    var = diff;
run;

/符号秩的结果就是wilcoxon符号秩检验

```

## ☐ 符号秩检验 (univariate)

```

/中位数是否为960的检验/
data salary;
    input salary @@;
    msalary = salary - 960;
cards;
blabla
;
proc univariate;
    var msalary;
run;

```

## ☐ 趋势检验(可以用符号检验来做)

```
proc univariate;
  var dif_weig;
run;
```

## ☐ 随机游程检验

```
proc iml;
x={0 0 0 1 1 0 0 1 1 0 1 0 1};
n = ncol(x);
s1 = 1;
n1 = 0;
n2 = 0;
do i = 1 to n-1;
  /计算0与1的个数/
  if x[i]=0 then n1 = n1 + 1;
  else n2 = n2 + 1;
  /判断前后是否相等/
  if x[i] - x[i+1] ^= 0 then s1 = s1 + 1;
end;
n1 = n1 + 1;
c1 = 2 * n1 * n2 / (n1 + n2)*(1+probit(0.025)/sqrt(n1+n2));
c2 = 2 * n1 * n2 / (n1 + n2)*(1+probit(0.025)/sqrt(n1+n2));
if s1 > c1 & s1 < c2 then print 'Pure random';
if s1 < c1 || s1 > c2 then print 'Not pure random';
quit;

/arima/
proc arima data = s;
  identify var = a stationary = (rw);
run;
```

## ☐ A-B 检验

```
data c6li7;
  input group $ diameter @@;
cards;
blabla
;
run;
proc NPAR1WAY data = c6li7 ab;
  class group;
  var diameter;
run;
```

# 7.相关分析

## ☐ Pearson

```
proc corr data = x pearson;  
  var x1 x2;  
run;
```

## ☐ Spearman

```
proc corr data = x Spearman;  
  var x3 x4;  
run;
```

## ☐ Kendall-Tau

```
proc corr data = x Kendall;  
  var x3 x4;  
run;
```

# 8.方差分析

$H_0: u_1 = u_2 = \dots = u_n$ ;  $H_1: u_1, \dots, u_n$ 不全相等

/ANOVA过程/

```
proc anova data = a;  
  class 变量;  
  model 因变量=效应变量;  
run;
```

/GLM过程/

```
proc glm data = a;  
  class 变量;  
  model y = x1 x2 x1 * x2; /其中x1*x2是交互作用项/  
run;
```

# 9.作图

/行标签/

```
axis1 label = ('行变量的名称')  
  order = ('1980', '1990')  
  minor = none  
  major = none;  
后面在proc gplot里写一行：  
maxis = axis1;
```

```

/列标签/
axis2 label=(angle=90 '列变量的名称')
  minor = none
  offset = (0,0);
后面在proc gplot里写一行：
raxis = axis2;

/柱形图/
proc gchart data = class;
  vbar y / discrete type = freq;
  maxis = axis1;
  raxis = axis2;
run;

/饼图/
proc gchart data = class;
  pie y / discrete type = cpercent;
run;

/堆叠图/
proc gchart data = anxiety;
  vbar subject / sumvar = score subgroup = trial;
run;

/散点图/
proc sgplot data = car;
  scatter x = weight y = horse / group = origin;
  xaxis label = 'blabla';
  yaxis label = 'blabla';
run;

/时序图/
proc sgplot data = stk;
  series y = opr x = date;
  series y = hipr x = date;
  series y = lopr x = date;
  yaxis label = '';
  xaxis label = '';
run;

/连线图/
proc gplot data = air;
  plot y * x / legend = legend1;
  symbol i = join line = 1;
run;

```

## 10.回归分析

### ☐ 线性一元（多元）回归

```

proc reg data = hemogl;
  model hemogl = ca mg fe mn cu / selection = stepwise;
run;

```

```

/进行预报/
/新的数据创建一个数据集/
data a;
    t = blabla;
    output;
run;

/数据集的合并/
data aa;
    set old a;
run;
proc reg data = aa;
    model y = x1-x3 / cli alpha = 0.05;
    id t;
run;

```

## ☐ 非线性回归

```

y=ae**bx
/做个线性变换/
data new;
    set old;
    z = log(y);
run;
proc reg data = new;
    model z = x;
run;

```

## ☐ 非参数拟合

```

proc loess data = sasuser.cars;
    model midprice = perform / clm;
run;

```

# 11.聚类分析

```

/系统聚类/
proc cluster data = ads method = cen;
    id n;/对n进行聚类/
    var ad;/变量/
run;

/动态聚类/
proc fastclust data = ads maxclusters = k;
    var age debt sex;
    id n;
run;

```

```

/变量聚类/
proc varclus data = ad;
    var age debt sex;
run;

```

## 12.SQL

15年期末13题。

```

/查询教师的不重复的department/
proc sql;
select distinct depart from teacher;
quit;

/查询score表中成绩为85，86或88的记录/
proc sql;
select * from score where degree in (85,86,88);
quit;

/查询3-105号课程的平均分/
proc sql;
select mean(degree) as average from score where CNO='3-105';
quit;

/查询score表中至少有5名学生选修的并且以3开头的课程的平均分数/
proc sql;
create table a as select count(*) as s,CNO,mean(degree) from score
group by CNO;
select * from a where s>5;
quit;

/查询最低分大于70，最高分小于90的SNO列/
proc sql;
create table a as
select SNO,min(degree) as min, max(degree) as max from score
group by SNO;
select * from a where min>70 & max<90;
quit;

```

16-4-1

将数据集class按group排序，并将birthday字段的年龄转换成年月日的格式，  
将数学成绩按60分划分，60分以下开根号再乘以10，60分到80分各加10分，81分以上乘以105%，  
若超过100分则按100分计算，结果存为classnew数据集，再将结果输出。

```

proc sql;
    create table classnew as select * from class order by group;
    alter table classnew modify birthday num format = yymmdd8. ,math num format=4.;
    update classnew set math = min(100,1.05*math) where math > 80;
    update classnew set math = math + 10 where 60<=math<=80;
    update classnew set math = sqrt(math)*10 where math < 60;
    select * from classnew;

```



```
quit;
```

16-4-2

先将classnew数据集中的name, group, math, english四个变量存为数据集bclass, 并将group按F和M转换成female male, 然后将classnew中的数学成绩math和英语成绩English去掉, 再将bclass与classnew 数据集合并成一个数据集classt, 并输出结果。

```
proc sql;
  create table bclass as select * from classnew(keep=name group math english);
  alter table bclass modify group char(6);
  update bclass set group = "female" where group in ("F1","F2");
  update bclass set group = "male" where group in ("M1","M2");
  select * from bclass;

  alter table classnew drop math, english;
  select * from classnew;

  create table classt as select * from classnew a join bclass b on a.name=b.name;
  select * from classt;
quit;
```

## 13.数据清洗

### ☐ 统计缺失值

```
/先定义好missing和非missing/
proc format;
  value $miss "" = "missing" other = "nonmissing";
  vlaue nmiss . = "missing" other "nonmissing";
run;
/利用freq过程求出数量与比例/
proc freq data = nlbb;
  tables team name position/missing;
  format team name posion $miss.;
  tables salary/missing;
  format salary nmiss.;
run;
```

## 14.数字游戏、算法

### ☐ 求出1-100内与3无关的数

```
data _null_;
  file print;
  l = -1;
```

```

do i = 1 to 1000;
  if(mod(i,3) ne 0) and find(i,'3') = 0 then do;
    l = l + 1;
    if (mod(l,12)=0) then do;
      put;
      l=0;
    end;
    put @(l*5) i @;
  end;
end;
run;

```

## ☐ 身份证题目

```

%macro IDcheck(id);
data d4;
  array x(18) $;
  array xx(18);
  array mult(17)(7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2);
  do i = 1 to 18;
    x(i) = substr(&id,i,1);
    xx(i) = x(i) * 1;
  end;
  check = '10X98765432';
  sum = 0;
  do i = 1 to 17;
    sum = sum + xx(i)*multiplier(i);
  end;
  remainder = mod(sum,11);
  if(substr(&id,18,1)=substr(check,remainder+1,1)) then put 'Correct!';
  else put 'Wrong!';
%mend IDcheck;

```

## ☐ 阶乘题目

```

data _null_;
  file print;
  s = 1;
  do i = 0 to 25 by 5;
    do j = i to i + 4;
      if j > 0 then s = s * j;
      put @((j-i)*15)s@;
    end;
    put;
  end;
run;

```

## ☐ 求1-1000内的素数

```

data _null_;
  file print;
  put '2'@@;
  t = 1;
  do i = 3 to 1000;
    s = 0;
    do j = 2 to i-1;
      if mod(i,j)=0 then s+1;
    end;
    if s = 0 then do;
      m = 6 * mod(t,10);
      if m / 6 = 0 then put;
      put @ m i@@;
      t+1;
    end;
  end;
run;

```

## □ 打印杨辉三角形

```

data _null_;
  array xx(6,6) x1-x36;
  do i =1 to 6;
    do j = 1 to i;
      if j=1 or i=j then do;
        xx(i,j)=1;
        put @(j*5) xx(i,j)@;
      end;
      else do;
        xx(i,j) = xx(i-1,j-1) + xx(i-1,j);
        put @(j*5) xx(i,j)@;
      end;
    end;
    put;
  end;
run;

```

## 15.IML

```

/用IML求线性方程组的解/
proc iml;
  a = {3 5 4, 5 7 3, 4 4 2};
  b = {-1, 2, 2};
  x = inv(a) * b;
  print x;
quit;

```

## 16.First、 Last

```
利用first last之前要sort一下
proc sort data = sashelp.class out = class;
  by age;
run;
/按不同年龄统计观测数/
data a;
  set class;
  by age;
  if first.age then n = 0;
    n+1;
  keep age n;
  if last.age;
run;
```

## 17.Merge、 Set

```
/set是纵向合并
data s1;
  set s2 s3;
run;

/Merge是横向合并,需要先排序
data s1;
  Merge s2 s3;
  by id;
run;

/以a为主键/
data d1;
  merge a(in=in1) b(in=in2);
  if in1;
  by id;
run;
/取a与b的交集/
data d2;
  merge a(in=in1) b(in=in2);
  if in1 and in2;
run;
```