44

Some of My Favorite Statistical Subroutines

44.2 Smoothplots (Mean and Median) of Chapter 5—X1 versus X2

```
%let Y=X1;
%let X=X2;
%let slice_X=10;

title ‘ ’;
data IN;
input X1 X2;
cards;
13 14
17 19
54 43
23 88
11 77
09 33
32 53
10 12
51 52
13 14
17 19
43 10
88 98
```

```
77 25
33 76
53 41
12 15
52 53
83 76
43 41
13 14
17 19
32 53
10 12
51 52
13 14
17 19
43 10
88 98
77 25
33 76
53 41
12 15
52 53
83 76
43 41
;
run;

data smooth;
set IN;
wt=1;
run;

PROC PRINT; run;

data score;
set smooth;
keep wt &Y &X;
run;
```

```
data notdot;
set score;
if &X ne .;

PROC MEANS data=notdot sum noprint; var wt;
output out=samsize (keep=samsize) sum=samsize;
run;

data scoresam (drop=samsize);
set samsize score;
retain n;
if _n_=1 then n=samsize;
if _n_=1 then delete;
run;

PROC SORT data=scoresam; by descending &X;
run;

data score_X;
set scoresam;
if &X ne . then cum_n+wt;
if &X = . then slice_X =.;
else slice_X=floor(cum_n*&slice_X/(n+1));
drop cum_n n;
run;

PROC SUMMARY data=score_X nway;
class slice_X;
var &X &Y;
output out=smout_&X mean = sm_&X sm_&Y/noinherit;
run;

title ‘Mean Smoothplot - X1 vs. X2’;
PROC PRINT data=smout_&X;
run;
```

```
PROC PLOT data=smout_&X HPCT=80 VPCT=80;
plot sm_&Y*sm_&X;
run;

title ‘ ’;
PROC SUMMARY data=score_X nway;
class slice_X ;
var &X &Y;
output out=smout_&X median = sm_&X sm_&Y/noinherit;
run;

PROC PRINT data=smout_&X;
title ‘Median Smoothplot - X1 vs. X2’;
run;

PROC PLOT data=smout_&X HPCT=80 VPCT=80;
plot sm_&Y*sm_&X;
run;
quit;
```

44.3 Smoothplots of Chapter 10—Logit and Probability

```
%let Y=Y;
%let X=X2;
%let slice_X=10;

data IN;
input X1 X2;
cards;
13 14
17 19
54 43
23 88
11 77
```

```
09 33
32 53
10 12
51 52
13 14
17 19
43 10
88 98
77 25
33 76
53 41
12 15
52 53
83 76
43 41
13 14
17 19
32 53
10 12
51 52
13 14
17 19
43 10
88 98
77 25
33 76
53 41
12 15
52 53
83 76
43 41
;
run;

data IN;
set IN;
Y=RAND('BERNOULLI',1/3);
```

```
data smooth;
set IN;
wt=1;
run;

PROC PRINT;
run;

data score;
set smooth;
keep wt &Y &X;
run;

data notdot;
set score;
if &X ne.;

PROC MEANS data=notdot sum noprint; var wt;
output out=samsize (keep=samsize) sum=samsize;
run;

data scoresam (drop=samsize);
set samsize score;
retain n;
if _n_=1 then n=samsize;
if _n_=1 then delete;
run;

PROC SORT data=scoresam; by descending &X;
run;

data score_X;
set scoresam;
if &X ne . then cum_n+wt;
if &X = . then slice_X =.;
else slice_X=floor(cum_n*&slice_X/(n+1));
drop cum_n n;
run;
```

```
PROC SUMMARY data=score_X nway;
class slice_X;
var &X &Y;
output out=smout_&X mean= sm_&X sm_&Y/noinherit;
run;

PROC PRINT data=smout_&X;
run;

data sliced_X;
set smout_&X;
Logit_&Y=log( sm_Y/(1-sm_Y));
if sm_&Y=1 then Logit_Y= 7;
if sm_&Y=0 then Logit_Y=-7;
Prob_&Y= exp(Logit_Y)/((1+exp(Logit_Y)));
run;

PROC PRINT data=sliced_X;
run;

PROC PLOT data=sliced_X HPCT=80 VPCT=80;
plot Logit_&Y*sm_&X /vaxis=-7 to +7 by 1;
format logit_&Y 6.2;
title ‘Smooth Logit Plot‘;
run;

PROC PLOT data=sliced_X HPCT=80 VPCT=80;
plot Prob_Y*sm_&X /vaxis=0 to 1 by 0.25;
format Prob_&Y 6.2;
title ‘Smooth Probability Plot‘;
run;
quit;
```

44.4 Average Correlation of Chapter 16—Among Var1 Var2 Var3 (Table 44.1)

```
%let varlist =
Var1 Var2 Var3;
title2 " AVG_CORR of &varlist ";

%let numvars=3;

data dat1;
input Var1 Var2 Var3 :4.0;
cards;
1234 2345 3456
5678 4567 8798
1256 0978 4567
;
run;

PROC CORR data=dat1 out=out;
var &varlist;
run;

data out1;
set out;
if _type_='MEAN' or _type_='STD' or _type_='N' then delete;
drop _type_;
array vars (&numvars)
&varlist;

array pos (&numvars) x1 - x&numvars;
do i= 1 to &numvars;
pos(i)=abs(vars(i));
end;
drop
&varlist i;
run;
```

```
data out2;
set out1;
array poss (&numvars) x1- x&numvars;
do i= 1 to &numvars;
if poss(i) =1 then poss(i)=.;
drop i;
end;
run;

PROC PRINT;
run;

PROC MEANS data=out2 sum;
output out=out3 sum=;


PROC PRINT;
run;

data out4;
set out3;
sum_=sum(of x1-x&numvars);
sum_div2= sum_/2;
bot= ((_freq_*_freq_)-_freq_)/2;
AVG_CORR= sum_div2/bot;
run;

data avg_corr;
set out4;
keep avg_corr;
PROC PRINT;
run;
```

44.5
Bootstrapped Decile Analysis of Chapter 29—Using Data from Table 23.4 (Table 44.2)

```
options source nonotes;
options nomprint nomlogic nosymbolgen;

Y=RESPONSE;
%let data_in=IN; /* add wt=1 to dataset IN */
%let depvar=Y;
%let indvars=_X11 - _X13 _X19 _X21;

%let samsize_bs=16003;
%let n_sampl_bs=50;

PROC SURVEYSELECT data=&data_in method=urs out=sample
   n=&samsize_bs rep=&n_sampl_bs outhits;
run;

%macro loop;
%do rep=1 %to &n_sampl_bs;

data Replicate&Rep;
set sample;
if Replicate=&Rep;
run;

%let dsn=Replicate&Rep;
ods exclude ODDSRATIOS;
PROC LOGISTIC data=&dsn nosimple noprint des outest=coef;
model &depvar = &indvars;
run;

PROC SCORE data=&dsn predict type=parms score=coef
out=score;
var &indvars;
run;
```

```
data score;
set score;
estimate=&depvar.2;

data notdot;
set score;
if estimate ne .;

PROC MEANS data=notdot sum noprint; var wt;
output out=samsize (keep=samsize) sum=samsize;
run;

data scoresam (drop=samsize);
set samsize score;
retain n;
if _n_=1 then n=samsize;
if _n_=1 then delete;
run;

PROC SORT data=scoresam; by descending estimate; run;

data score;
set scoresam;
if estimate ne . then cum_n+wt;
if estimate = . then dec=.;
else dec=floor(cum_n*10/(n+1));
run;

PROC SUMMARY data=score missing;
class dec;
var &depvar wt;
output out=sum_dec sum=sum_can sum_wt;

data sum_dec;
set sum_dec;
avg_can=sum_can/sum_wt;
run;
```

```
data avg_rr;
set sum_dec;
if dec=.;
keep avg_can;
run;

data sum_dec1;
set sum_dec;
if dec=. or dec=10 then delete;
cum_n +sum_wt;
r =sum_can;
cum_r +sum_can;
cum_rr=(cum_r/cum_n)*100;
avg_cann=avg_can*100;
run;

data avg_rr;
set sum_dec1;
if dec=9;
keep avg_can;
avg_can=cum_rr/100;
run;

%let scoresam=&Rep;
data scoresam&Rep;
set avg_rr sum_dec1;
retain n;
if _n_=1 then n=avg_can;
if _n_=1 then delete;
lift&Rep = (cum_rr/n);
if dec ne .;
keep dec lift&Rep;
run;

PROC SORT data=scoresam&Rep; by dec;
run;
%end;
```

```
data combine;
merge %do i=1 %to &n_sampl_bs;
scoresam&i
%end;;
by dec;
run;

data bs_lift_SE;
set combine;
bs_est=mean(of lift:);
bs_std=std(of lift:);
bs_SE=1.28*bs_std;
keep dec bs_est bs_SE;
run;

ods exclude ODDSRATIOS;
PROC LOGISTIC data=&data_in nosimple noprint des outest=coef;
model &depvar = &indvars;
run;

PROC SCORE data=&data_in predict type=parms score=coef
out=score;
var &indvars;
run;

data score;
set score;
estimate=&depvar.2;

data notdot;
set score;
if estimate ne .;

PROC MEANS data=notdot sum noprint; var wt;
output out=samsize (keep=samsize) sum=samsize;
run;

data scoresam (drop=samsize);
set samsize score;
```

```
retain n;
if _n_=1 then n=samsize;
if _n_=1 then delete;
run;

PROC SORT data=scoresam; by descending estimate;
run;

data score;
set scoresam;
if estimate ne . then cum_n+wt;
if estimate = . then dec=.;
else dec=floor(cum_n*10/(n+1));
run;

PROC SUMMARY data=score missing;
class dec;
var &depvar wt;
output out=sum_dec sum=sum_can sum_wt;
run;

data sum_dec;
set sum_dec;
avg_can=sum_can/sum_wt;
run;

data avg_rr;
set sum_dec;
if dec=.;
keep avg_can;
run;

data sum_dec1;
set sum_dec;
if dec=. or dec=10 then delete;
cum_n +sum_wt;
r =sum_can;
cum_r +sum_can;
```

```
cum_rr=(cum_r/cum_n)*100;
avg_cann=avg_can*100; run;

data avg_rr;
set sum_dec1;
if dec=9;
keep avg_can;
avg_can=cum_rr/100; run;

data scoresam;
set avg_rr sum_dec1;
retain n;
if _n_=1 then n=avg_can;
if _n_=1 then delete;
lift=(cum_rr/n);
if dec ne .;
_2SAM_EST=2*lift;
keep dec _2SAM_EST lift;
run;

data boot;
merge
bs_lift_SE scoresam;
lift_bs=_2SAM_EST-bs_est;
keep dec _2SAM_EST bs_est lift_bs bs_SE;
run;

%end;
%mend;
dm 'clear log';
%loop

ods exclude ODDSRATIOS;
PROC LOGISTIC data=&data_in nosimple noprint des outest=coef;
model &depvar = &indvars;
freq wt;
run;
```

```
PROC SCORE data=&data_in predict type=parms score=coef
out=score;
var &indvars;
run;

data score;
set score;
estimate=&depvar.2;
label
estimate='estimate';
run;

data notdot;
set score;
if estimate ne .;

PROC MEANS data=notdot sum noprint; var wt;
output out=samsize (keep=samsize) sum=samsize;
run;

data scoresam (drop=samsize);
set samsize score;
retain n;
if _n_=1 then n=samsize;
if _n_=1 then delete;
run;

PROC SORT data=scoresam; by descending estimate;
run;

data score;
set scoresam;
if estimate ne . then cum_n+wt;
if estimate = . then dec=.;
else dec=floor(cum_n*10/(n+1));
run;
```

```
PROC SUMMARY data=score missing; class dec;
var &depvar wt;
output out=sum_dec sum=sum_can sum_wt;

data sum_dec;
set sum_dec;
avg_can=sum_can/sum_wt;
run;

data avg_rr;
set sum_dec;
if dec=.;
keep avg_can;
run;

data sum_dec1;
set sum_dec;
if dec=. or dec=10 then delete;
cum_n +sum_wt;
r =sum_can;
cum_r +sum_can;
cum_rr=(cum_r/cum_n)*100;
avg_cann=avg_can*100;
run;

data avg_rr;
set sum_dec1;
if dec=9;
keep avg_can;
avg_can=cum_rr/100;
run;

data scoresam;
set avg_rr sum_dec1;
retain n;
if _n_=1 then n=avg_can;
if _n_=1 then delete;
lift=(cum_rr/n);
```

```
if dec=0 then decc=' top ';
if dec=1 then decc=' 2 ';
if dec=2 then decc=' 3 ';
if dec=3 then decc=' 4 ';
if dec=4 then decc=' 5 ';
if dec=5 then decc=' 6 ';
if dec=6 then decc=' 7 ';
if dec=7 then decc=' 8 ';
if dec=8 then decc=' 9 ';
if dec=9 then decc='bottom';
if dec ne .;
run;

PROC SORT data= scoresam; by dec;
PROC SORT data= boot; by dec;
run;

data scoresam_bs;
merge
scoresam boot; by dec;
run;

options label;
title1' ';
title2" samsize_bs=&samsize_bs, n_sampl_bs=&n_sampl_bs ";

PROC PRINT data=scoresam_bs d split='*' noobs;
var decc sum_wt r avg_cann cum_rr lift lift_bs bs_SE;
label decc='DECILE'
sum_wt ='NUMBER OF*INDIVIDUALS'
r ='NUMBER OF*RESPONDERS'
cum_r ='CUM No. CUSTOMERS w/* RESPONDERS'
avg_cann ='RESPONSE *RATE (%)'
cum_rr ='CUM RESPONSE * RATE (%)'
lift ='C U M*Single-Sample*LIFT (%)'
lift_bs ='C U M*BOOTSTRAP*LIFT (%)'
bs_SE='BOOTSTRAP*MARGIN of*ERROR (80%)';
sum sum_wt r;
```

```
format sum_wt r cum_n cum_r comma8.0;
format avg_cann cum_rr 6.2;
format lift lift_bs 3.0;
format bs_SE 5.1;
run;
```

44.6 H-Spread Common Region of Chapter 42

```
%let spread=50;
title “H-spread&spread”;

data IN;
call streaminit(12345);

do id=1 to 120;
X1 = RAND(‘NORMAL’,10, 1);
X2 = RAND(‘NORMAL’,10, 1.5);
X3 = RAND(‘NORMAL’,10, 2);
output;
end;
run;

PROC RANK data=IN groups=100 out=OUT;
var X1 X2 X3;
ranks X1r X2r X3r;
run;

PROC PRINT data=OUT;
run;

PROC RANK data=IN groups=100 out=OUT;
var X1-X3;
ranks X1r X2r X3r;
run;
```

```
data H_spread&spread._X1;
set out;
rhp=(100-&spread)/2;
if x1r=> (rhp-1) and x1r<=(99-rhp);
keep id x1 x1r;
run;

data H_spread&spread._X2;
set out;
rhp=(100-&spread)/2;
if x2r=> (rhp-1) and x2r<=(99-rhp);
keep id x2 x2r;
run;

data H_spread&spread._X3;
set out;
rhp=(100-&spread)/2;
if x3r=> (rhp-1) and x3r<=(99-rhp);
keep id x3 x3r;
run;

PROC SORT data=H_spread&spread._X1; by id;
PROC SORT data=H_spread&spread._X2; by id;
PROC SORT data=H_spread&spread._X3; by id;
run;

data H_spread&spread._X1X2X3;
merge
H_spread&spread._X1 (in=var_x1)
H_spread&spread._X2 (in=var_x2)
H_spread&spread._X3 (in=var_x3);
by id;
if var_x1=1 and var_x2=1 and var_x3=1;
run;


PROC MEANS data=H_spread&spread._X1X2X3 mean n ;
var X1-X3;
run;
```

```
data H_spread&spread._X1;
set H_spread&spread._X1X2X3;
var='x1';
x=x1;
keep id x var;

data H_spread&spread._X2;
set H_spread&spread._X1X2X3;
var='x2';
x=x2;
keep id x var;

data H_spread&spread._X3;
set H_spread&spread._X1X2X3;
var='x3';
x=x3;
keep id x var;

data H_spread&spread._X1X2X3;
set H_spread&spread._X1 H_spread&spread._X2 H_spread&spread._X3;

PROC PRINT;
run;

PROC SORT; by var;
PROC PRINT data= H_spread&spread._X1X2X3;
run;

PROC BOXPLOT data=H_spread&spread._X1X2X3;
plot x*var;
run;
```

44.7 Favorite—Proc Corr with Option Rank, Vertical Output (Table 44.3)

```
data dat1;
input X1 - X4: 3.0 TARGET 1.0;
cards;
123 234 345 456 1
. 756 . 654 0
234 843 654 867 1
123 234 345 456 0
654 856 534 654 1
234 543 854 867 1
123 834 845 456 0
654 756 534 654 0
234 543 654 867 0
;
run;

PROC PRINT;
run;

data dat1;
set dat1;
wt=1;
run;

PROC CORR data=dat1 rank fisher;
ods output fisherpearsoncorr=out;
var x1-x4;
with target;
freq wt;
run;

ods listing;
PROC PRINT data=out;
run;
```

```
data out1;
set out;
abs_corr = abs(corr);
CorrCoef_with_TARGET=corr;
Predictor=var;
N=nobs;
keep var corr n pvalue abs_corr CorrCoef_with_TARGET Predictor;
run;

PROC SORT data=out1; by descending abs_corr;
run;

data out2;
set out1;
if abs_corr ge .0;
if CorrCoef_with_TARGET = . then delete;

data out2;
set out2;
Rank=_n_;
run;

PROC PRINT data=out2 noobs;
var Rank Predictor CorrCoef_with_TARGET n pvalue;
run;
```

44.8 Favorite—Decile Analysis—Response (Table 44.4)

```
%let data_in=IN;
%let depvar=Y;
%let indvars=X1 X2 X3;

data &data_in;
input &depvar &indvars wt;
cards;
```

1	63.28405135	–62.89590924	0.31725	1
1	–7.965165127	9.077917498	0.29397	1
1	–40.8721149	41.85990786	0.40705	1
1	108.8084024	–107.6672824	0.25316	1
1	3.071713061	–2.215322147	0.40705	1
1	44.96645653	–44.18664467	0.25316	1
1	2.328170141	–1.89973146	0.24562	1
1	89.08870743	–88.21705972	0.42732	1
1	30.1080088	–29.0253107	0.24562	1
1	–11.14966201	11.97082199	0.25316	1
1	24.6912264	–23.85538734	0.25316	1
1	33.46889223	–32.68556731	0.40705	1
1	51.82377813	–51.4138173	0.40705	1
1	70.28970224	–69.42221865	0.24562	1
1	–95.85890655	97.00002655	0.40705	1
1	77.53692092	–77.19292134	0.26126	1
0	3.309578275	–3.261180349	0.24562	1
0	10.12748375	–9.549172853	0.25316	1
0	–12.88207239	13.97592671	0.29397	1
0	–17.32877567	18.18516658	0.31111	1
0	–70.59773747	71.24695425	0.31111	1
0	43.27915239	–42.13803238	0.24562	1
0	–7.880514668	8.995154718	0.25316	1
0	40.93399103	–40.09173673	0.25316	1
0	81.07550795	–80.35859121	0.24562	1
0	–7.965165127	9.063100546	0.24562	1
0	36.93492473	–35.95553062	0.28211	1
0	23.23610469	–22.80766601	0.25339	1
0	0	1.141120008	0.24562	1
0	0	0.939629385	0.25316	1
0	81.17218438	–80.76633346	0.24562	1
0	21.67949378	–20.97110166	0.24562	1
0	61.36545177	–60.91557128	0.25316	1
0	61.36545177	–60.95549093	0.31725	1
0	77.90838509	–77.58149603	0.28481	1
0	77.90838509	–77.60466917	0.24562	1

```
0       77.90838509      –77.08023514      0.32738      1
0       16.48495995      –15.88724129      0.40705      1
0       39.74610442      –38.99089853      0.24562      1
0       30.7499237       –29.94045894      0.24562      1
;
run;

PROC LOGISTIC data=&data_in nosimple des outest=coef;
model &depvar = &indvars;
freq wt;
run;

PROC SCORE data=&data_in predict type=parms score=coef
out=score;
var &indvars;
run;

data score;
set score;
estimate=&depvar.2;
run;

data notdot;
set score;
if estimate ne.;

PROC MEANS data=notdot sum noprint; var wt;
output out=samsize (keep=samsize) sum=samsize;
run;

data scoresam (drop=samsize);
set samsize score;
retain n;
if _n_=1 then n=samsize;
if _n_=1 then delete;
run;

PROC SORT data=scoresam; by descending estimate;
run;
```

```
data score;
set scoresam;
if estimate ne . then cum_n+wt;
if estimate = . then dec=.;
else dec=floor(cum_n*10/(n+1));
run;

PROC SUMMARY data=score missing;
class dec;
var &depvar wt;
output out=sum_dec sum=sum_can sum_wt;

data sum_dec;
set sum_dec;
avg_can=sum_can/sum_wt;
run;

data avg_rr;
set sum_dec;
if dec=.;
keep avg_can;
run;

data sum_dec1;
set sum_dec;
if dec=. or dec=10 then delete;
cum_n +sum_wt;
r =sum_can;
cum_r +sum_can;
cum_rr=(cum_r/cum_n)*100;
avg_cann=avg_can*100;
run;

data avg_rr;
set sum_dec1;
if dec=9;
keep avg_can;
avg_can=cum_rr/100;
```

```
run;

data scoresam;
set avg_rr sum_dec1;
retain n;
if _n_=1 then n=avg_can;
if _n_=1 then delete;
lift=(cum_rr/n);
if dec=0 then decc=' top ';
if dec=1 then decc=' 2 ';
if dec=2 then decc=' 3 ';
if dec=3 then decc=' 4 ';
if dec=4 then decc=' 5 ';
if dec=5 then decc=' 6 ';
if dec=6 then decc=' 7 ';
if dec=7 then decc=' 8 ';
if dec=8 then decc=' 9 ';
if dec=9 then decc='bottom';
if dec ne .;
run;

title1 ‘ ’;
title2‘ Decile Analysis based on ’;
title3“ &depvar (RESPONSE) regressed on &indvars ”;

PROC PRINT data=scoresam d split=‘*’ noobs;
var decc sum_wt r avg_cann cum_rr lift;
label decc='DECILE’
   sum_wt ='NUMBER OF*INDIVIDUALS’
   r =‘NUMBER OF*RESPONDERS’
   cum_r =‘CUM No. CUSTOMERS w/* RESPONSES’
   avg_cann =‘RESPONSE *RATE (%)’
   cum_rr =‘CUM RESPONSE * RATE (%)’
   lift =‘ C U M *LIFT (%)’;
sum sum_wt r;
format sum_wt r cum_n cum_r comma10.;
format avg_cann cum_rr 5.2;
format lift 3.0; run;
```

44.9 Favorite—Decile Analysis—Profit (Table 44.5)

```
%let data_in=IN;
%let depvar=Y;
%let indvars=X1 X2;

data &data_in;
input &indvars &depvar wt;
cards;
1 0.64417 14212.99 1
0 0.05839 908.11 1
0 0.06754 538.77 1
1 0.21690 1548.25 1
1 0.50600 11701.12 1
0 0.02847 13.70 1
0 0.26161 1575.19 1
0 0.29051 1602.65 1
0 0.04119 528.26 1
0 0.05310 618.37 1
1 0.44417 12312.99 1
0 0.06839 978.11 1
0 0.07754 738.77 1
1 0.31690 1348.25 1
1 0.51600 11901.12 1
0 0.04847 17.70 1
0 0.28161 1595.19 1
0 0.31051 1662.65 1
0 0.05119 578.26 1
0 0.06310 698.37 1
;
run;

PROC REG data=&data_in outest=coeff;
estimate: model Y = &indvars;
run;
```

```
PROC SCORE data=&data_in
out=score (keep= wt estimate Y )
predict SCORE=coeff TYPE=PARMS;
var &indvars;
run;

data notdot;
set score;
if estimate ne.;

PROC MEANS data=notdot sum noprint; var wt;
output out=samsize (keep=samsize) sum=samsize;
run;

data scoresam (drop=samsize);
set samsize score;
retain n;
if _n_=1 then n=samsize;
if _n_=1 then delete;
run;

PROC SORT data=scoresam; by descending estimate;
run;

data score;
set scoresam;
if estimate ne . then cum_n+wt;
if estimate = . then dec=.;
else dec=floor(cum_n*10/(n+1));
run;

PROC SUMMARY Data=Score missing;
class dec;
var Y wt;
output out=sum_dec sum=sum_Y sum_wt;
```

```
data sum_dec;
set sum_dec;
avg_Y=sum_Y/sum_wt;
run;

data avg_fix;
set sum_dec;
if dec ne .;
keep sum_Y sum_wt;

PROC SUMMARY data=avg_fix;
var sum_Y sum_wt;
output out=fix_dec sum=num_Y tot_cus;
run;

data avg_ss;
set fix_dec;
avg_Y=num_Y/tot_cus;
keep avg_Y;
run;

data sum_dec1;
set sum_dec;
if dec=. or dec=10 then delete;
cum_n +sum_wt;
s =sum_Y;
cum_s +sum_Y;
cum_ss=(cum_s/cum_n);
avg_Ys=avg_Y;
run;

data scoresam;
set avg_ss sum_dec1;
retain n;
if _n_=1 then n=avg_Y;
if _n_=1 then delete;
```

```
lift=(cum_ss/n)*100;
if dec=0 then decc=‘ top’;
if dec=1 then decc=‘ 2 ’;
if dec=2 then decc=‘ 3 ’;
if dec=3 then decc=‘ 4 ’;
if dec=4 then decc=‘ 5 ’;
if dec=5 then decc=‘ 6 ’;
if dec=6 then decc=‘ 7 ’;
if dec=7 then decc=‘ 8 ’;
if dec=8 then decc=‘ 9 ’;
if dec=9 then decc=‘bottom’;
if dec ne .;
run;

title1 ‘ ’;
title2‘ Decile Analysis based on ’;
title3“ &depvar (PROFIT) regressed on &indvars ”;

PROC PRINT data=scoresam d split='*' noobs;
var decc sum_wt s avg_Ys cum_ss lift;
label decc='DECILE'
sum_wt ='NUMBER OF*CUSTOMERS'
s ='TOTAL*PROFIT'
cum_s ='CUM No. CUSTOMERS w/* PROFIT'
avg_Ys =' DECILE* MEAN PROFIT'
cum_ss ='DECILE* CUM PROFIT'
lift =' C U M * LIFT ';
sum sum_wt s;
format s dollar14.2;
format sum_wt cum_n cum_s comma10.;
format avg_Ys cum_ss dollar10.2;
format lift 3.0;
run;
```

44.10 Favorite—Smoothing Time-Series Data (Running Medians of Three) (Table 44.6)

```
%let ln=10;

data IN;
input TS $2. X1 – X&ln;
cards;
TS 23 45 36 57 65 19 29 44 33 56
;
run;

PROC PRINT;
title2’ dataset IN ‘;
run;

PROC TRANSPOSE data=IN out=tposed;
id TS;
var X1-X&ln;
run;

PROC PRINT data=tposed;
title2 ‘ dataset tposed ‘;
run;

data tposed;
set tposed;
Time+1;
run;

PROC PRINT;
title2 ‘ dataset tposed with Time’;
run;

PROC PRINT data=in;
run;
```

```
PROC PLOT data=tposed vpercent=90 hpercent=90;
plot TS *time/ haxis=1 to &ln by 1 vaxis=20 to 70 by 5;
title3’ Original Time-Series Plot’; run;

data Medians_of_3;
set IN;
array d{*} X1-X&ln;
array med [&ln] med1 - med&ln;
array smmed[&ln] smmed1 - smmed&ln;
do i=1 to dim(d);
med{i}=.;
if i <=dim(d)-2 then do;
med{i} = median(d{i},d{i+1},d{i+2});
smmed(i+1) = med(i);
end;
end;
smmed(1) = median( d(2), d(3), (3*d(2) -2*d(3)) );
smmed(dim(d))= median( d(dim(d)), d(dim(d)-1), (3*d(dim(d)-1) -2*d(dim(d))) );
drop i;
run;

PROC PRINT;
var smmed1-smmed&ln;
title2 ' Smooth Sequence of Xs '; run;

PROC TRANSPOSE data=Medians_of_3 out=tposed;
id TS;
var smmed1 - smmed&ln;
run;

PROC PRINT data=tposed;
title2 ' dataset sm tposed '; run;

data tposed;
set tposed;
Time+1;
run;
```

```
PROC PRINT;
title2 ' dataset sm tposed with Time';
run;

PROC PLOT data=tposed vpercent=90 hpercent=90;
plot TS *time/ haxis=1 to &ln by 1 vaxis=20 to 70 by 5;
title3’ Smooth Time-Series Plot ‘;
run;

%let ln=10;

data IN;
input TS $2. X1 - X&ln;
cards;
TS 36 45 45 57 57 29 29 33 33 44
;
run;

PROC PRINT;
title2’ dataset IN ‘;
run;

data Medians_of_3;
set IN;
array d{*} X1-X&ln;
array med [&ln] med1 - med&ln;
array smmed[&ln] smmed1 - smmed&ln;
do i=1 to dim(d);
med{i}=.;
if i <=dim(d)-2 then do;
med{i} = median(d{i},d{i+1},d{i+2});
smmed(i+1) = med(i);
end;
end;
smmed(1) = median( d(2), d(3), (3*d(2) -2*d(3)) );
smmed(dim(d))= median( d(dim(d)), d(dim(d)-1), (3*d(dim(d)-1) -2*d(dim(d))) );
drop i j;
run;
```

```
PROC PRINT;
var smmed1-smmed&ln;
title2 ‘ Smooth Sequence of Xs ‘;
run;

PROC TRANSPOSE data=Medians_of_3 out=tposed;
id TS;
var smmed1 - smmed&ln;
run;

PROC PRINT data=tposed;
title2 ‘ dataset    sm tposed ‘;
run;

data tposed;
set tposed;
Time+1;
run;

PROC PRINT;
title2 ‘ dataset sm tposed with Time’;
run;

PROC PLOT data=tposed vpercent=90 hpercent=90;
plot TS *time/ haxis=1 to &ln by 1 vaxis=20 to 70 by 5;
title3’ Double Smooth Time-Series Plot ‘;
run;
quit;
```

44.11

Favorite—First Cut Is the Deepest—Among Variables with Large Skew Values

```
PROC MEANS data=IN skew;
var X1-X24;
output out=skews skew=;
run;

PROC PRINT data=skews;
run;

PROC TRANSPOSE data=skews
out=Tskews (rename=(Col1=Skew _NAME_=Variable));
var X1-X24;
run;


PROC SORT data=Tskews; by descending Skew;
run;

PROC PRINT data=Tskews;
run;
```