SAS IMPLEMENTATIONS

Outline

- LIDAR: univariate spline
- Onion yields: simple interaction model
- Dog example:
  - four treatment groups (fixed effect)
  - dogs nested in group (random effect)
  - covariate with nonparametrically modeled effect

LIDAR

Read data

```
options linesize = 65 ;
data lidar ;
infile ’C:\Documents and Settings\David Ruppert\My Documents
!\talks\SemiShortCourse\SAS\lidar.txt’ ;
input range logratio ;
run ;
title ’LIDAR Penalized Spline’ ;
```

LIDAR

Create spline basis functions

```
proc iml ;
use lidar var{range logratio};
read all ;
nknots = 24 ;
rr = (1:nknots) * (221/(1+nknots)) ;
rr = int(rr) ;
knots = range[rr] ;
plus = J(221,nknots) ;
do k=1 to nknots ;
plus[,k] = (range - knots[k])##2 # (range > knots[k]) ;
end ;
varname = ’range’ || ’logratio’ || (’z1’:’z24’) ;
outmatrix = range||logratio||plus ;
create lidar2 from outmatrix [colname=varname] ;
append from outmatrix;
```

LIDAR

Fit a mixed model

```
proc mixed method=reml ;
model logratio = range|range / outpred=outdata  ;
random z1-z24 / type=toep(1) s ;
run ;
```

LIDAR

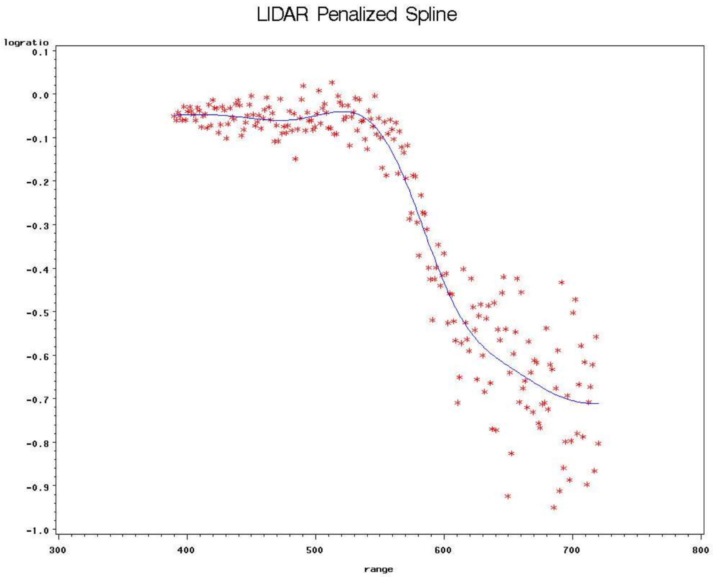
Plot fit

```
proc gplot ;
plot logratio*range=1 pred*range=2/overlay;
symbol1 v=star i=none color=red;
symbol2 v=none i=join color=blue;
run ;
```

Onion Example

Input Data

```
options linesize = 65 ;
data onion ;
infile ’C:\Documents and Settings\David Ruppert\My Documents\talks
!\SemiShortCourse\SAS\onion.txt’ ;
input obs density yield location;
run ;
title ’Onion Yields - Penalized Spline’ ;
```



Onions Example

Create Plus Functions

```
proc iml ;
use onion var{obs density yield location};
read all ;
location = location - 1 ;
nknots = 20 ;
degree = 1 ;
rr = (1:nknots) * (84/(1+nknots)) ;
rr = int(rr) ;
knots = density[rr] ;
plus = J(84,nknots) ;
interact = plus||plus ;
```

Onions Example

Create Plus Functions – cont.

```
do k=1 to nknots ;
plus[,k] = ((density - knots[k])##degree) # (density > knots[k]) ;
interact[,2*k-1] = plus[,k]#(location<.5) ;
interact[,2*k] = plus[,k]#(location> .5) ;
end ;
varname = ’obs’ || ’density’ || ’yield’ || ’location’
!||(’z1’:’z20’) || (’interact1’:’interact40’) ;
outmatrix = obs ||density || yield || location
!|| plus||
interact ;
create lidar2 from outmatrix [colname=varname] ;
append from outmatrix;
```

Onions Example

Fit Interaction Model

```
proc mixed method=REML ;
class location ;
model yield = density density*density location density*location
!location*density*density / outpred=outdata  ;
random z1-z10 / type=toep(1)  ;
random interact1-interact20/ type=toep(1) ;
data outdata ;
set outdata ;
Pred_interaction = pred ;
drop pred ;
run ;
```

Onions Example

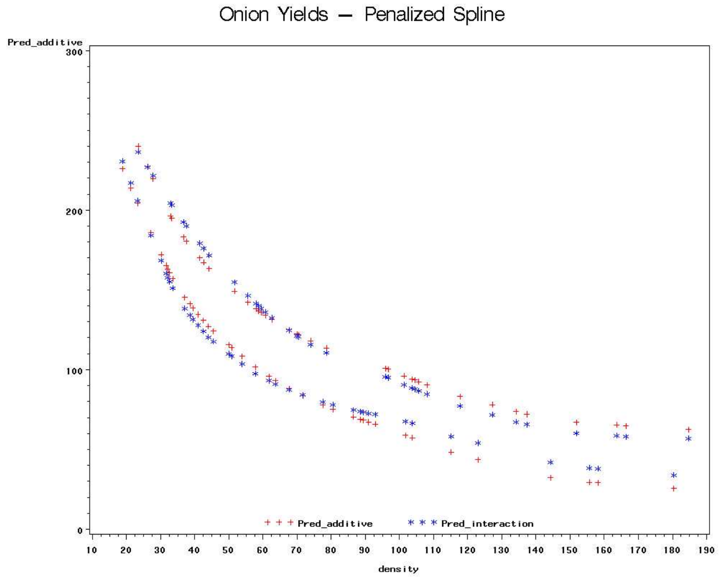
Fit Additive Model

```
proc mixed ;
model yield = density density*density location  / outpred=outdata2  ;
random z1-z10 / type=toep(1)  ;
run ;
data outdata2 ;
set outdata2 ;
Pred_additive = pred ;
drop pred ;
run ;
data outdata ;
merge outdata outdata2 ;
by obs ;
run ;
```

Onions Example

Plots

```
proc gplot ;
legend1 label = none position=inside;
symbol1 v=plus i=none color=red;
symbol2 v=star i=none color=blue;
plot pred_additive*density=1 pred_interaction*density=2
!/overlay legend=legend1;
run ;
proc gplot ;
symbol1 v=plus i=none color=red;
symbol2 v=star i=none color=blue;
plot yield*density=location;
run ;
```



Dogs Example

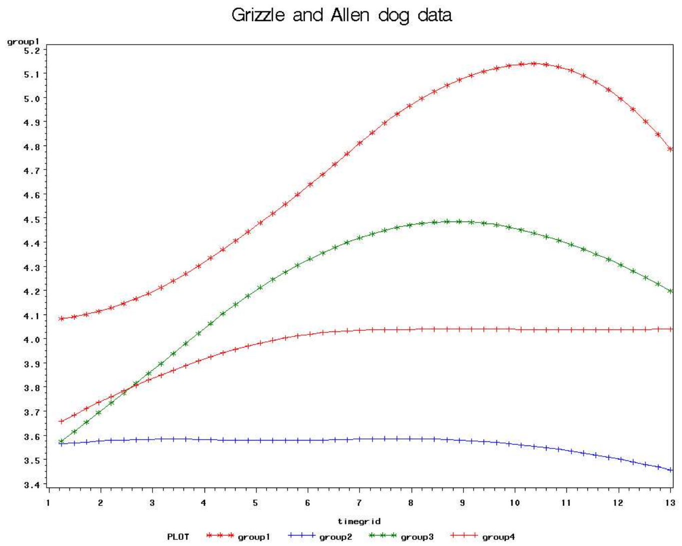
Data Description

- four treatment groups
  - control
  - extrinsic cardiac denervation three weeks prior to coronary occlusion
  - extrinsic cardiac denervation immediately prior to coronary occlusion
  - bilateral thoracic sympathectomy and stellectomy three weeks prior to coronary occlusion
- approximately 9 dogs/group
  - dog is a random effect

Dogs Example

Data Description

- each dog observed at time = 3, 5, 7, 9, 11, and 13 minutes
  - time is a fixed covariate
    - $*$ nonlinear effect
    - $*$ interacts with treatment and dog
- response = coronary sinus potassium



Dogs Example

Input Data

```
options linesize = 65 ;
data dog ;
infile ’C:\Documents and Settings\David Ruppert\My Documents\talks
!\SemiShortCourse\SAS\dogdata.txt’ ;
input group dog time potassium ;
run ;
title ’Grizzle and Allen dog data’ ;
```

Dogs Example

Create Plus Functions

```
proc iml ;
use dog var{group dog time potassium};
read all ;
nknots = 3 ;
knots = {4 7 10} ;
plus = J(252,5*nknots) ;
do k=1 to nknots ;
plus[,k] = (time - knots[k])##2 # (time > knots[k]) ;
plus[,3+k] = plus[,k] # (group=1) ;
plus[,6+k] = plus[,k] # (group=2) ;
plus[,9+k] = plus[,k] # (group=3) ;
plus[,12+k] = plus[,k] # (group=4) ;
end ;
```

Dogs Example

Create Plus Functions

```
varname = ’group’ || ’dog’ || ’time’ || ’potassium’
!|| (’main1’:’main3’) || (’inter1’:’inter12’) ;
outmatrix = group||dog||time||potassium||plus ;
create dog2 from outmatrix [colname=varname] ;
append from outmatrix;
quit ;
```

Dogs Example

PROC MIXED

```
class group dog ;
model potassium = group time time2 time*group time2*group /
!s outpred=outdata  ;
random main1-main3 / type=toep(1) s ;
random int1-int12 / type=toep(1) s ;
random dog / type=toep(1) s ;
random time*dog / type=toep(1) s ;
random time*time*dog /type=toep(1) s ;
ods output  solutionr=r solutionf=f;
run ;
```

Dogs Example

Compute BLUPs

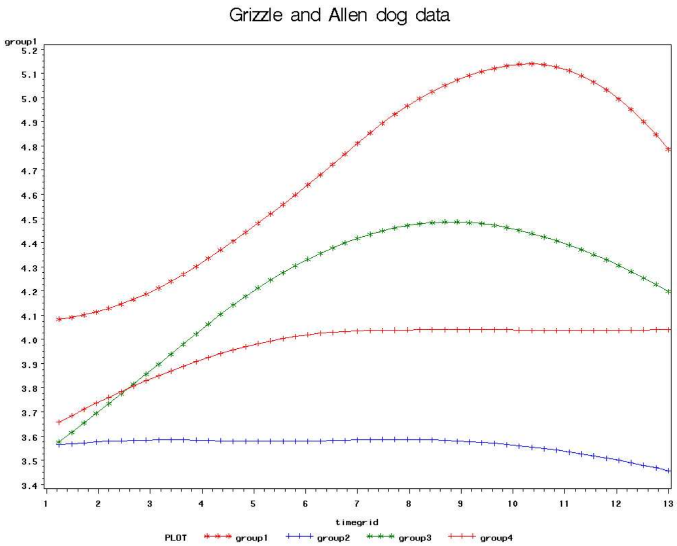
```
proc iml ;
timegrid = 1 + ((1:50))#12/50 ;
timegrid = timegrid‘ ;
nknots = 3 ;
knots = {4 7 10} ;
plus = J(50,nknots) ;
do k=1 to nknots ;
plus[,k] = (timegrid - knots[k])##2 # (timegrid > knots[k]) ;
end ;
use f var{Estimate} ; read all ;estf=Estimate ;
use r var{Estimate} ; read all ;estr = Estimate ;
group1 = estf[1] + estf[2] + (estf[6]+estf[8])#timegrid
!+ (estf[7]+estf[12])#timegrid##2
+ (estr[1]+estr[4])#plus[,1] +(estr[2]+estr[5])#plus[,2]
!+(estr[3]+estr[6])#plus[,3];
```

```
group2 = estf[1] + estf[3] + (estf[6]+estf[9])#timegrid
!+ (estf[7]+estf[13])#timegrid##2
+ (estr[1]+estr[7])#plus[,1] +(estr[2]+estr[8])#plus[,2]
!+ (estr[3]+estr[9])#plus[,3];

group3 = estf[1] + estf[4] + (estf[6]+estf[10])#timegrid
!+ (estf[7]+estf[14])#timegrid##2
+ (estr[1]+estr[10])#plus[,1] +(estr[2]+estr[11])#plus[,2]
! +(estr[3]+estr[12])#plus[,3];

group4 = estf[1] + estf[5] + (estf[6]+estf[11])#timegrid
!+ (estf[7]+estf[15])#timegrid##2
+ (estr[1]+estr[13])#plus[,1] +(estr[2]+estr[14])#plus[,2]
!+(estr[3]+estr[15])#plus[,3];

varname = ’group1’|| ’group2’|| ’group3’ ||’group4’ ||’timegrid’;
outmatrix = group1||group2||group3||group4||timegrid ;
```

```
create dog3 from outmatrix [colname=varname] ;
append from outmatrix ;
quit ;
```

Dogs Example

```
proc gplot data=dog3 ;
plot group1*timegrid group2*timegrid group3*timegrid
!group4*timegrid/ overlay legend ;
symbol1 v=star i=join color="red";
symbol2 v=plus i=join color="blue" ;
symbol3 v=star i=join color="green";
symbol4 v=plus i=join color= "red" ;
run ;
```


Grizzle and Allen dog data
group1
5.2
5.1
5.0
4.9
4.8
4.7
4.6
4.5
4.4
4.3
4.2
4.1
4.0
3.9
3.8
3.7
3.6
3.5
3.4
1
2
3
4
5
6
7
8
9
10
11
12
13
timegrid
PLOT
group1
group2
group3
group4