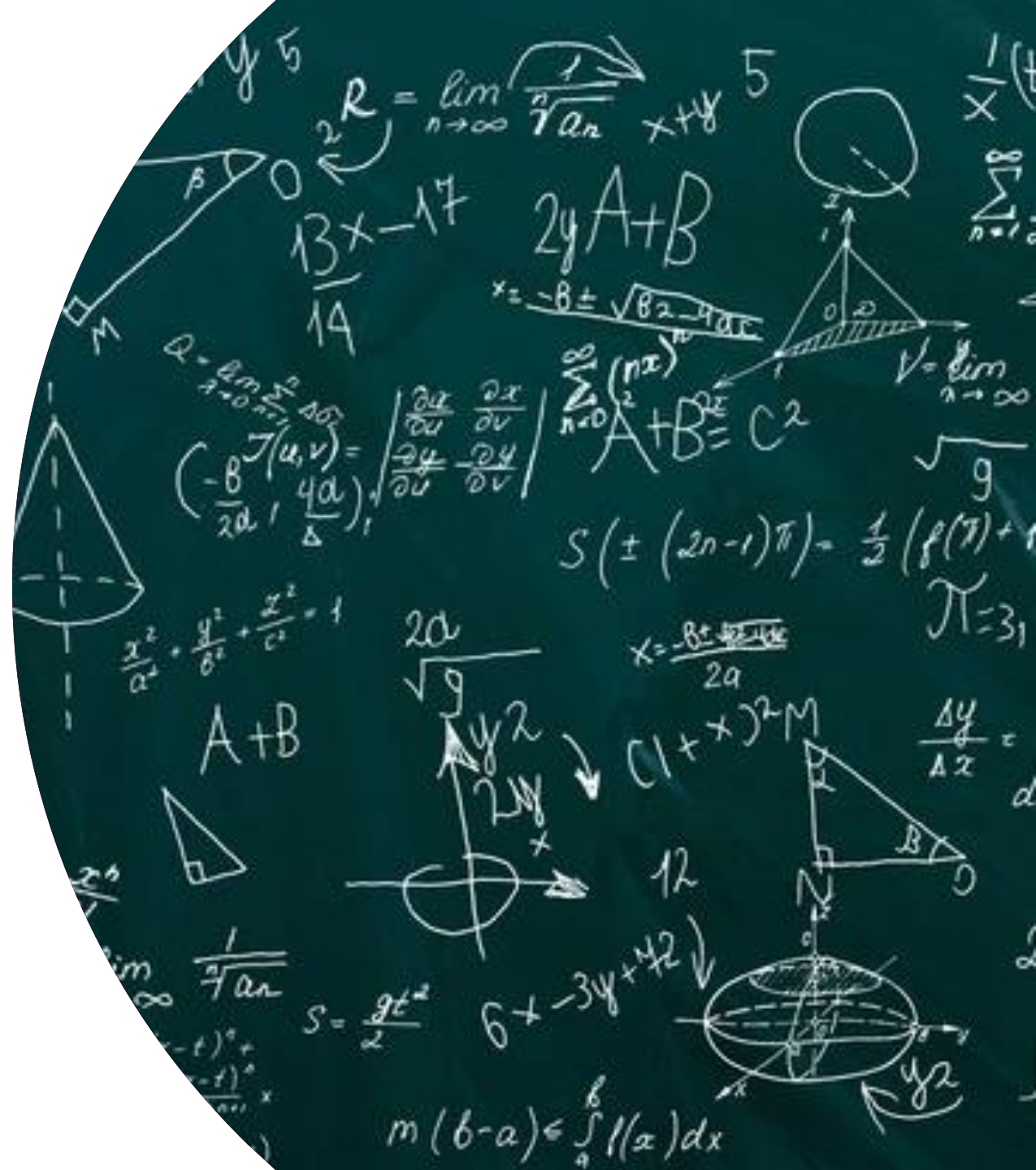


A programmer's introduction to modelling with proc mixed focusing on how to debug and validate, common mistakes and how to interpret the output

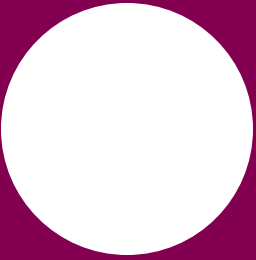
George Gibson



Contents

- What is proc mixed
- Model estimates with brief discussion on 2- and 3-way interactions and model interpretation
- Throughout: Debugging tips and common mistakes





What is proc mixed



“PROC MIXED provides easy accessibility to numerous **mixed linear models** that are useful in many common statistical analyses. In the style of the GLM procedure, PROC MIXED fits the specified **mixed linear model** and produces appropriate statistics.” - SAS

Mixed Linear model is a model that looks at both fixed and random effects



“PROC MIXED provides easy accessibility to numerous mixed linear models that are useful in many common statistical analyses. In the style of the GLM procedure, PROC MIXED fits the specified mixed linear model and produces appropriate statistics.” - SAS

Mixed Linear model is a model that looks at both fixed and random effects



Things we know will have a certain impact:

- Type of soil a seed is planted in
- Type of medication a patient is on (Drug A vs Drug B)

Things that might have a small, unpredictable impact that we want to consider:

- Plant pot location
- Hospital location



Proc Mixed Syntax

- BY
- CLASS
- CODE
- ID
- MODEL
- RANDOM
- REPEATED
- PARMS
- PRIOR
- CONTRAST
- ESTIMATE
- LSMEANS
- LSMESTIMATE
- SLICE
- STORE
- WEIGHT



Proc Mixed Syntax

- BY – Obtain separate analysis based on groups
- CLASS – Define your categorical variables
- MODEL – Define the model itself (including the fixed effects)
- RANDOM – Defines the random effects
- REPEATED – Defines a time variable where data is collected
- ESTIMATE – Create model estimates for specific “scenarios” of interest
- LSMEANS – Creates least squares means of the fixed effect



Proc Mixed Example

```
proc mixed data=input;  
  by imputenm;  
  class usubjid trt01pn avisitn;  
  model chg = trt01pn avisitn trt01pn*avisitn;  
  repeated avisitn / type=un subject=usubjid;  
  lsmeans trt01pn|avisitn/ pdiff cl;  
Run;
```



Proc Mixed Breakdown

```
proc mixed data=input;
```

Telling SAS we want to use proc mixed and that our input dataset is input



Proc Mixed Breakdown

```
proc mixed data=input;
```

Telling SAS we want to use proc mixed and that our input dataset is input

Validation Tip: Dev and Validation's input datasets must be identical, including sorting variables



Proc Mixed Breakdown

```
class usubjid trt01pn avisitn;
```

Our model has three categorical variables



Proc Mixed Breakdown

```
class usubjid trt01pn avisitn;
```

Our model has three categorical variables

Validation Tip: SAS will assume any character variables are categorical, and any numerical variables are continuous



Proc Mixed Breakdown

```
model chg = trt01pn avisitn trt01pn*avisitn;
```

We would like to look at how treatment, visit and the interaction term between treatment and visit affects the change in baseline.

Interaction terms explore the combined effect of two or more variables on our outcome. So, because of the inclusion of the interaction term, we can explore how the effectiveness of the treatment changes over time. It helps us to understand if the treatment has a different effect on patients depending on whether they are early in their treatment journey or further along.



Proc Mixed Breakdown

```
model chg = trt01pn avisitn trt01pn*avisitn;
```

We would like to look at how treatment, visit and the interaction term between treatment and visit affects the change in baseline.

Interaction terms explore the combined effect of two or more variables on our outcome. So, because of the inclusion of the interaction term, we can explore how the effectiveness of the treatment changes over time. It helps us to understand if the treatment has a different effect on patients depending on whether they are early in their treatment journey or further along.

Validation Tip: The order of the variables must be identical, especially when using interaction terms



Proc Mixed Breakdown

```
repeated avisitn / type=un subject=usubjid;
```

AVISITN (ie visits) are repeated, and our analysis should account for the fact that multiple measurements are taken for the same subject (specified by USUBJID). Type=un specifies the variance-covariance matrix to be used in the analysis and should be listed in your SAP



Proc Mixed Breakdown

```
repeated avisitn / type=un subject=usubjid;
```

AVISITN (ie visits) are repeated, and our analysis should account for the fact that multiple measurements are taken for the same subject (specified by USUBJID). Type=un specifies the variance-covariance matrix to be used in the analysis and should be listed in your SAP

Validation Tip: If you get a warning in your log relating to model convergence, query with your statistician



Proc Mixed Breakdown

```
lsmeans trt01pn|avisitn/ pdiff cl;
```

Requesting least squares means for the interaction between treatment and visit. The optional extras are to request the differences between these means (pdiff) and the confidence intervals (cl)



Proc Mixed Breakdown

```
lsmeans trt01pn|avisitn/ pdiff cl;
```

Requesting least squares means for the interaction between treatment and visit. The optional extras are to request the differences between these means (pdiff) and the confidence intervals (cl)

Validation Tip: Lsmeans statement assumes equal weighting on all covariates ie if you have a categorical variable “Race” with two levels, Lsmeans will assume that the split is 50:50. This is rarely true and so that is why estimate statements are used, as you can accurately define the weightings/proportions.

This can be avoided by using a weighted dataset; however this will not be covered in this presentation.

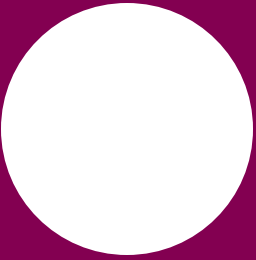


Proc Mixed Breakdown

`Run;`

You have finished writing your model code





Estimates and interactions



Can be used to make estimates for specific scenarios/conditions:

- Change from baseline for Drug 1 at Visit 3
- Change from baseline for Drug 1 over Visit 3
- Change from baseline at Visit 4: Drug 1 vs Drug 2



Syntax

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race height weight;
```

TRT01PN has 4 levels, AVISITN has 5 and race has 2

Estimate 'Treatment 1 at Visit 3'

```
height &average_h.
```

```
weight &average_w.
```

```
race 0.72 0.28
```



Syntax

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race height weight;
```

TRT01PN has 4 levels, AVISITN has 5 and race has 2

Estimate 'Treatment 1 at Visit 3'

```
height &average_h.
```

```
weight &average_w.
```

```
race 0.72 0.28
```

```
trt01pn 1 0 0 0
```

```
avisitn 0 0 1 0 0 / CL;
```

Estimate for the change from baseline at Visit 3 for a subject who is of average height, weight and is on treatment 1



Syntax

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race height weight;
```

TRT01PN has 4 levels, AVISITN has 5 and race has 2

Estimate 'Treatment 1 at Visit 3'

```
height &average_h.  
weight &average_w.  
race 0.72 0.28  
trt01pn 1 0 0 0  
avisitn 0 0 1 0 0 / CL;
```

Validation Tip: These values must be to the same degrees of accuracy in both dev and val. If val has 0.716 and 0.284 then you would get different values

Estimate for the change from baseline at Visit 3 for a subject who is of average height, weight and is on treatment 1



Syntax

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race height weight;
```

TRT01PN has 4 levels, AVISITN has 5 and race has 2

Estimate 'Treatment 1 at Visit 3'

```
height &average_h.  
weight &average_w.  
race 0.72 0.28  
trt01pn 1 0 0 0  
avisitn 0 0 1 0 0 / CL;
```

Validation Tip 2: When deriving these (either programmatically or otherwise), ensure that each subject only has one observation in your dataset ie if you are using proc means ensure the input dataset is one observation per subject

Estimate for the change from baseline at Visit 3 for a subject who is of average height, weight and is on treatment 1



Syntax

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race height weight;
```

TRT01PN has 4 levels, AVISITN has 5 and race has 2

Estimate 'Treatment 1 over Visit 3'

```
height &average_h.
```

```
weight &average_w.
```

```
race 0.72 0.28
```

```
trt01pn 1 0 0 0
```

```
avisitn 0.3333333333 0.3333333333 0.3333333333 0 0 / CL;
```

Estimate for the change from baseline over Visit 3 for a subject who is of average height, weight and is on treatment 1



Syntax

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race height weight;
```

TRT01PN has 4 levels, AVISITN has 5 and race has 2

Estimate 'Treatment 1 over Visit 3'

```
height &average_h.
```

```
weight &average_w.
```

```
race 0.72 0.28
```

```
trt01pn 1 0 0 0
```

```
avisitn 0.3333333333 0.3333333333 0.3333333333 0 0 / CL;
```

Validation Tip: SAS will round these in the background so that the sum will equal 1, you do not need to give one value a weighting of 0.3333333334

Estimate for the change from baseline over Visit 3 for a subject who is of average height, weight and is on treatment 1



Syntax

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race height weight;
```

TRT01PN has 4 levels, AVISITN has 5 and race has 2

Estimate 'Treatment 1 vs Treatment 2 at Visit 4'

```
height &average_h.
```

```
weight &average_w.
```

```
race 0.72 0.28
```

```
trt01pn 1 -1 0 0
```

```
avisitn 0 0 0 1 0 / CL;
```

Estimate for the difference from change from baseline at Visit 4 for subjects who are of average height, weight with one being on treatment 1 and another being on treatment 2



2-Way Interactions

```
class trt01pn avisitn;  
  model chg = trt01pn avisitn trt01pn*avisitn;
```

Estimate 'Treatment 1 at Visit 3'

...

Trt01pn*avisitn 0

X***Y** long “string” where every group of X must sum up to 1 or 0



2-Way Interactions

```
class trt01pn avisitn;  
  model chg = trt01pn avisitn trt01pn*avisitn;
```

Estimate 'Treatment 1 at Visit 3'

...

Trt01pn*avisitn	<u>0 0 0 0 0</u>	<u>0 0 0 0 0</u>	<u>0 0 0 0 0</u>	<u>0 0 0 0 0</u>
	Treatment 1	Treatment 2	Treatment 3	Treatment 4

X***Y** long “string” where every group of X must sum up to 1 or 0

X=“Block” and the number of levels in your variables is the number of blocks you need



2-Way Interactions

```
class trt01pn avisitn;  
  model chg = trt01pn avisitn trt01pn*avisitn;
```

Estimate 'Treatment 1 at Visit 3'

...

Trt01pn*avisitn	0 0 1 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
	Treatment 1	Treatment 2	Treatment 3	Treatment 4

0 Visit 1

0 Visit 2

0 Visit 3

0 Visit 4

0 Visit 5

X*Y long "string" where every group of X must sum up to 1 or 0

X="Block" and the number of levels in your variables is the number of blocks you need

Y="Values" and the number of levels within this variables is the number of values within each block



2-Way Interactions – Validation Tip

Validation tip: The order of the variables in the model statement is key and must be identical (the interaction term order is not important and can differ between dev and val)

```
class trt01pn avisitn;  
  model chg = trt01pn avisitn height weight trt01pn*avisitn;  
Estimate 'Treatment 1 at Visit 3'
```

Trt01pn*avisitn	<u>0 0 1 0 0</u>	<u>0 0 0 0 0</u>	<u>0 0 0 0 0</u>	<u>0 0 0 0 0</u>
	Treatment 1	Treatment 2	Treatment 3	Treatment 4

```
class trt01pn avisitn;  
  model chg = avisitn trt01pn height weight trt01pn*avisitn;  
Estimate 'Treatment 1 at Visit 3' 'Treatment 3 at Visit 1'
```

Trt01pn*avisitn	<u>0 0 1 0</u>	<u>0 0 0 0</u>	<u>0 0 0 0</u>	<u>0 0 0 0</u>	<u>0 0 0 0</u>
	Visit 1	Visit 2	Visit 3	Visit 4	Visit 5



3-Way Interactions

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race trt01pn*avisitn*race;
```

TRT01PN has 4 levels, AVISITN has 5 levels, RACE has 2 levels

Estimate 'Treatment 1 at Visit 3 for race group 1'

...

```
Trt01pn*avisitn*race 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0
```



3-Way Interactions

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race trt01pn*avisitn*race;
```

TRT01PN has 4 levels, AVISITN has 5 levels, RACE has 2 levels

Estimate 'Treatment 1 at Visit 3 for race group 1'

...

```
Trt01pn*avisitn*race 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0
```

X*Y*Z long “string”

X=Number of levels within this variable number of rows you need



3-Way Interactions

```
class trt01pn avisitn race;  
model chg = trt01pn avisitn race trt01pn*avisitn*race;
```

TRT01PN has 4 levels, AVISITN has 5 levels, RACE has 2 levels

Estimate 'Treatment 1 at Visit 3 for race group 1'

...

Trt01pn*avisitn*race	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0

4 rows, one for each treatment

X*Y*Z long "string"

X=Number of levels within this variable number of rows you need



3-Way Interactions

```
class trt01pn avisitn race;  
model chg = trt01pn avisitn race trt01pn*avisitn*race;
```

TRT01PN has 4 levels, AVISITN has 5 levels, RACE has 2 levels

Estimate 'Treatment 1 at Visit 3 for race group 1'

...

Trt01pn*avisitn*race	0 0	0 0	0 0	0 0	0 0
	0 0	0 0	0 0	0 0	0 0
	0 0	0 0	0 0	0 0	0 0
	0 0	0 0	0 0	0 0	0 0

4 rows, one for each treatment

5 blocks, one for each visit

X*Y*Z long "string"

X=Number of levels within this variable number of rows you need

Y=Number of levels within this variable is the number of blocks within each row



3-Way Interactions

```
class trt01pn avisitn race;  
model chg = trt01pn avisitn race trt01pn*avisitn*race;
```

TRT01PN has 4 levels, AVISITN has 5 levels, RACE has 2 levels

Estimate 'Treatment 1 at Visit 3 for race group 1'

...

2 values, one for each race level

Trt01pn*avisitn*race	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

4 rows, one for each treatment

5 blocks, one for each visit

X*Y*Z long "string"

X=Number of levels within this variable number of rows you need

Y=Number of levels within this variable is the number of blocks within each row

Z=Number of levels within this variable is the number of values in each block



3-Way Interactions

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race trt01pn*avisitn*race;
```

TRT01PN has 4 levels, AVISITN has 5 levels, RACE has 2 levels

Estimate 'Treatment 1 at Visit 3 for race group 1'

...

Trt01pn*avisitn*race	0 0	0 0	1 0	0 0	0 0
	0 0	0 0	0 0	0 0	0 0
	0 0	0 0	0 0	0 0	0 0
	0 0	0 0	0 0	0 0	0 0

X*Y*Z long “string”

X=Number of levels within this variable number of rows you need

Y=Number of levels within this variable is the number of blocks within each row

Z=Number of levels within this variable is the number of values in each block



3-Way Interactions

```
class trt01pn avisitn race;  
model chg = trt01pn avisitn race trt01pn*avisitn*race;
```

Estimate 'Treatment 1 over Visit 3 for race group 1'

...

Trt01pn*avisitn*race	0.333333	0	0.333333	0	0.333333	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0

X*Y*Z long “string”

X=Number of levels within this variable number of rows you need

Y=Number of levels within this variable is the number of blocks within each row

Z=Number of levels within this variable is the number of values in each block



3-Way Interactions

```
class trt01pn avisitn race;  
  model chg = trt01pn avisitn race trt01pn*avisitn*race;
```

Estimate 'Treatment 1 vs Treatment 2 at Visit 3 for race group 1'

...

Trt01pn*avisitn*race	0 0	0 0	1 0	0 0	0 0
	0 0	0 0	-1 0	0 0	0 0
	0 0	0 0	0 0	0 0	0 0
	0 0	0 0	0 0	0 0	0 0

X*Y*Z long “string”

X=Number of levels within this variable number of rows you need

Y=Number of levels within this variable is the number of blocks within each row

Z=Number of levels within this variable is the number of values in each block



Summary

- Proc mixed is an incredibly powerful tool
- While mistakes can happen, they are often simple and things consider
 - Datasets not being sorted
 - Model statements having conflicting order
 - Weighting having different degrees of accuracy
- When making estimates, keep your code clean and organised



Q&A



Confidentiality Notice

This file is private and may contain confidential and proprietary information. If you have received this file in error, please notify us and remove it from your system and note that you must not copy, distribute or take any action in reliance on it. Any unauthorized use or disclosure of the contents of this file is not permitted and may be unlawful. AstraZeneca PLC, 1 Francis Crick Avenue, Cambridge Biomedical Campus, Cambridge, CB2 0AA, UK, T: +44(0)203 749 5000, www.astrazeneca.com

