HW3 (Due: April 25)

1. Assume you have 3 independent normal variables {Y1, Y2, Y3} with mean of {mu1, mu2, mu3} and standard deviation {sigma1, sigma2, sigma3}. Derive the joint density function via 2 ways; one is univariate way and another is multivariate way. Verify if they are equal or not equal. [Remark: Multivariate way can be done using vector and matrix. Remember formula with Determinant? Check its definition. You would be pleasantly surprised you used this in the past!].

See attached paper

1. Show how to convert 1) wide to long form and 2) long to wide form. Provide sample program and output.

TLC data to long format (first 20 rows)

ID group time lead

1 1 P 1 30.8

2 1 P 2 26.9

3 1 P 3 25.8

4 1 P 4 23.8

5 2 A 1 26.5

6 2 A 2 14.8

7 2 A 3 19.5

8 2 A 4 21.0

9 3 A 1 25.8

10 3 A 2 23.0

11 3 A 3 19.1

12 3 A 4 23.2

13 4 P 1 24.7

14 4 P 2 24.5

15 4 P 3 22.0

16 4 P 4 22.5

17 5 A 1 20.4

18 5 A 2 2.8

19 5 A 3 3.2

20 5 A 4 9.4

TLC to wide from long format (first 6 subjects)

ID group Base wk1 wk4 wk6

1 1 P 30.8 26.9 25.8 23.8

2 2 A 26.5 14.8 19.5 21.0

3 3 A 25.8 23.0 19.1 23.2

4 4 P 24.7 24.5 22.0 22.5

5 5 A 20.4 2.8 3.2 9.4

6 6 A 20.4 5.4 4.5 11.9

1. Using TLC data, reproduce Tables 15 and 16 (in S188 and 191).

S188: I was not able to get Chi square test statistic. R gives by default F statistic. Of note is that the p-value for the group effect was border line significant with R, as opposed to SAS, where p-value was < 0.0001. After scourging the forums to find cod to get Chisq like SAS in R, I was not able to find it.

I checked the author’s website for the code. He got the same results, with no Chisq p-value… Maybe he didn’t try hard enough, and was just trying to sell the book for R users under the allure of available R code.

numDF F-value p-value

group 1 4.2267 0.0405

week 3 61.4944 <.0001

group:week 3 35.9293 <.0001

S191:

Coefficients:

Value Std.Error t-value p-value

(Intercept) 26.272 0.7102929 36.98756 0.0000

groupA 0.268 1.0045059 0.26680 0.7898

week1 -1.612 0.7919199 -2.03556 0.0425

week4 -2.202 0.8149021 -2.70217 0.0072

week6 -2.626 0.8885253 -2.95546 0.0033

groupA:week1 -11.406 1.1199438 -10.18444 0.0000

groupA:week4 -8.824 1.1524456 -7.65676 0.0000

groupA:week6 -3.152 1.2565645 -2.50843 0.0125

Here results were virtually the same. Only difference is that R uses t-statistic instead of Z (which seems more reasonable to me, although sample size is very large)

1. Using TLC data, you can compute the ‘unadjusted’ means for 8 subgroup (=2 arms at 4 time points). Show how/if you can derive these means from Table 16 – you can pick 3 means if you wish. What do you infer from this?

Three adjusted means from regression coefficients of table 16 for group A at weeks 1, 4 and 6:

Group A @ week 1= 26.272 + 0.268 – 1.612 – 11.406 = 13.522

Group A @ week 4 = 26.272 + 0.268 - 2.202 - 8.824 = 15.514

Group A @ week 6 = 26.272 + 0.268 - 2.626 - 3.152 = 20.762

These are the same as the unadjusted means. I suppose this is because maximum likelihood/least squares estimation aims at finding parameter estimates that maximize the likelihood of the data for each subpopulation of covariate patterns. Thus, find the parameter estimates that are best suited to obtain the unbiased estimate of Mu|X (the subpopulation mean) for each covariate pattern, which is Ybar|X.

My inference from this is that parameter estimates from the model in table 16 are unbiased estimates of the population parameters, because when combined the estimates produce Ybar|X whose expectation is Mu|X, the subpopulation mean.

1. Using TLC data, run the following codes and comment what these codes (=2 different ways) are actually doing.

**proc** **mixed** noclprint=**10** order=data;

class id group time;

model y = group time group\*time / s chisq;

repeated time / type=un subject=id r;

**run**;

This first peace of code is running the GLS model of lead concentration as a function of group and time and their interaction. Since the data has not been sorted, and the “order=data” command is in the model, it means that the procedure will use the “default baseline” values in the data as reference for the model (specifically group=A and time=4)

**proc** **sort**;

by group descending time;

**proc** **mixed** noclprint=**10** order=data;

class id group time;

model y = group time group\*time / s chisq;

repeated time / type=un subject=id r;

**run**;

This is running the same model as before, but before, in the proc sort statement, the data is sorted by group in a descending manner (first A and then P), and by time in an ascending manner (1,2,3,4). This way, the reference levels used by proc mixed are group P and time 1. This type of order is easier to understand and communicate.

In R we need to do something similar with treatment, as by default it takes it in alphabetical order, using group A as a reference. There’s no need for that with time, as it uses the lowest value as a reference by default.

[Remark: Non-SAS users can still try this by read-in/copy/paste/click! Not sure if STATA or R has a counterpart of this problem...]

Trivia for Curious George (optional):

<http://en.wikipedia.org/wiki/Determinant#History>

"determinants were first used in the Chinese mathematics textbook The Nine Chapters on the Mathematical Art (九章算術, Chinese scholars, around the 3rd century BCE)."

Some many smart people before (and contemporary with) us. As Newton said “we stand on the shoulders of giants”

Good to remember that if a determinant is non-zero than the system of equations has a unique answer!

Determinant and parallelogram

<https://math.okstate.edu/people/binegar/3013-S99/3013-l13.pdf>

This is very interesting…never thought of determinants as areas of the vector space encompassed by the matrix.