

# À propos de la méthode de Bonferroni

## Résumé

*L'objet de cette vignette est de rappeler la méthode de Bonferroni et de préciser les commandes `means` et `lsmeans` de la procédure GLM de SAS, dans le cadre de la modélisation linéaire gaussienne. En particulier, bien que très proches, ces deux commandes ne fonctionnent pas de la même façon pour la mise en œuvre de la méthode de Bonferroni et présentent, l'une comme l'autre, des bizarreries que nous indiquons.*

*Retour au [plan du cours](#)*

## 1 Rappels sur la méthode de Bonferroni

Dans le cadre du modèle linéaire gaussien, le test usuel, permettant de tester toute hypothèse nulle linéaire en les paramètres, est le test de Fisher ; ce test se ramène à un test de Student, plus simple, lorsque l'hypothèse nulle ne porte que sur un seul paramètre.

Dans la pratique, un problème qui se pose souvent avec ces tests est celui des tests multiples. L'idée est la suivante : si l'on travaille avec un niveau de test (une erreur de première espèce) de 5%, et si l'on fait un seul test, alors, lorsque la valeur de la statistique de test dépasse la valeur limite (fournie par les tables ou par les logiciels), ce qui conduit à rejeter l'hypothèse nulle  $H_0$ , il y a un risque (une probabilité) de 0,05, pour que l'on rejette à tort cette hypothèse ; ce risque est donc contrôlé et faible. Par contre, si l'on répète ce même test pour divers paramètres avec les mêmes données (dans le cadre de la même expérience), le risque de rejeter à tort augmente d'autant et devient grand si l'on fait de nombreux tests. De plus, ce risque n'est plus contrôlé (à la limite, si l'on fait 100 tests différents avec les mêmes données et si l'on rejette 5 fois  $H_0$ , il y a de fortes chances pour que ces rejets soient faits à tort).

En particulier, dans le cadre d'une ANOVA (analyse de variance, ou plan factoriel), une fois un modèle choisi, il est fréquent qu'on cherche à savoir si les différences des moyennes de la variable réponse entre les niveaux d'un facteur pris en compte dans le modèle sont significatives ou non. Plus ces niveaux sont

nombreux, plus il faut faire de tests pour répondre à cette question, et plus le risque de conclure à tort à la significativité des différences est grand.

Pour se prémunir contre l'augmentation du risque de première espèce dans les tests multiples, il existe différentes méthodes, la plus célèbre étant celle dite de Bonferroni. Cette dernière consiste à majorer le risque réel pour l'ensemble des tests (impossible à déterminer) par le niveau de test choisi (en général 5%), en divisant pour cela le niveau de chacun des tests effectués par le nombre total de tests. Ainsi, si on dispose d'un facteur à 4 niveaux et si on veut tester toutes les différences deux à deux entre les niveaux, il y a 6 tests à réaliser, de sorte que chacun est fait avec un niveau de 5/6, soit 0,83%, ce qui est faible. Pour 5 niveaux, on doit réaliser chaque test au niveau de 0,1% et ainsi de suite. La méthode est trop conservatrice (elle a tendance à ne rejeter que très rarement  $H_0$ , à cause du niveau très faible de chaque test) et s'avère peu intéressante dès que le nombre de niveaux est supérieur ou égal à 5 : en gros, la correction est alors telle que le remède est pire que le mal...

On notera que des méthodes nouvelles pour diminuer le risque de rejeter à tort  $H_0$ , autrement dit pour se prémunir des faux positifs, ont été mises au point assez récemment. Elles sont nettement plus performantes que la méthode de Bonferroni et nous ne recommandons d'utiliser cette dernière, le cas échéant, que pour des facteurs à 3 ou 4 niveaux.

## 2 Les commandes `means` et `lsmeans` de la procédure GLM de SAS

### 2.1 Principe général

Lorsqu'on réalise une ANOVA avec la procédure GLM de SAS, on peut utiliser les deux commandes `means` et `lsmeans` pour calculer les moyennes partielles de la variable réponse  $Y$  au sein des niveaux des facteurs entrant dans le modèle considéré et pour tester la significativité des différences entre ces moyennes partielles.

Dans un premier temps, nous allons utiliser le célèbre exemple des rendements de vaches laitières pour illustrer l'usage de ces deux commandes (se reporter au chapitre 3 pour la présentation de ces données). Nous considérons que le modèle le mieux adapté à ces données est le modèle complet (effets de

chacun des deux facteurs et des interactions) et nous exécutons le programme SAS suivant :

```
data vach;
infile 'vach.don';
input f1 f2 y;
run;
* ----- ;
proc glm data=vach;
class f1 f2;
model y = f1 f2 f1*f2 / ss3;
means f2;
lsmeans f2;
run;
```

```
f1      1      0.4000000      0.4000000      0.15      0.7029
f2      3     290.2000000     96.7333333     35.83     <.0001
f1*f2    3     41.0000000     13.6666667      5.06     0.0056
```

```
*****
Level of f2      N      Mean      Std Dev
1              10      9.1000000     1.37032032
2              10     11.0000000     1.94365063
3              10     11.5000000     1.43372088
4              10     16.4000000     2.54732976
```

```
*****
Least Squares Means
f2      y LSMEAN
1          9.1000000
2         11.0000000
3         11.5000000
4         16.4000000
```

En voici les principaux résultats :

```
The GLM Procedure

Class Level Information

Class      Levels      Values
f1          2          1 2
f2          4          1 2 3 4

Number of Observations Read      40
```

Dependent Variable: y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	331.6000000	47.3714286	17.54	<.0001
Error	32	86.4000000	2.7000000		
Corrected Total	39	418.0000000			
R-Square	Coeff Var	Root MSE	y Mean		
0.793301	13.69306	1.643168	12.00000		

Source	DF	Type III SS	Mean Square	F Value	Pr > F
--------	----	-------------	-------------	---------	--------

Les deux dernières parties des résultats ci-dessus sont celles correspondant respectivement aux commandes `means` et `lsmeans`. On pourra vérifier que les moyennes fournies sont tout simplement celles des 10 observations correspondant à chaque niveau du second facteur. On notera que `means` fournit en plus l'estimation usuelle de l'écart-type des 10 observations de chaque niveau de `f2` (avec un dénominateur en  $n - 1$ ).

## 2.2 Tests des différences et méthode de Bonferroni

Les deux commandes considérées permettent aussi de mettre en œuvre les tests de comparaison, avec ou sans la correction de Bonferroni. Commençons par regarder ce que fait la commande `means` pour tester les différences entre niveaux du facteur `f2` et comment elle met en œuvre cette correction.

```
proc glm data=vach;
class f1 f2;
model y = f1 | f2 / ss3;
means f2 / t;
means f2 / bon;
run;
```

Voici les résultats.

t Tests (LSD) for y

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

```
Alpha                      0.05
Error Degrees of Freedom    32
Error Mean Square           2.7
Critical Value of t         2.03693
Least Significant Difference  1.4968
```

Means with the same letter are not significantly different.

t Grouping	Mean	N	f2
A	16.4000	10	4
B	11.5000	10	3
B	11.0000	10	2
C	9.1000	10	1

\*\*\*\*\*

Bonferroni (Dunn) t Tests for y

NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

```
Alpha                      0.05
Error Degrees of Freedom    32
Error Mean Square           2.7
Critical Value of t         2.81234
Minimum Significant Difference  2.0666
```

Means with the same letter are not significantly different.

Bon Grouping	Mean	N	f2
A	16.4000	10	4
B	11.5000	10	3
B	11.0000	10	2
C	9.1000	10	1

L'option t de la commande means réalise les tests de Student (*t-tests*) de comparaison des moyennes de Y pour les différents niveaux de f2 (chaque test

est fait avec un niveau de 5%) et présente les résultats sous forme de regroupement ou non de ces niveaux. Le principe est exactement le même avec l'option bon, mais chaque test est maintenant fait avec un niveau six fois plus petit et les résultats sont, bien sûr, différents.

Regardons maintenant ce que fait la commande lsmeans dans les mêmes conditions.

```
proc glm data=vach;
class f1 f2;
model y = f1 | f2 / ss3;
lsmeans f2 / pdiff;
lsmeans f2 / tdiff;
lsmeans f2 / adj=bon tdiff;
run;
```

Voici les résultats.

Least Squares Means

f2	y LSMEAN	LSMEAN Number
1	9.1000000	1
2	11.0000000	2
3	11.5000000	3
4	16.4000000	4

Least Squares Means for effect f2  
Pr > |t| for H0: LSMean(i)=LSMean(j)

Dependent Variable: y

i/j	1	2	3	4
1		0.0145	0.0026	<.0001
2	0.0145		0.5011	<.0001
3	0.0026	0.5011		<.0001
4	<.0001	<.0001	<.0001	

NOTE: To ensure overall protection level, only probabilities associated with pre-planned comparisons should be used.

\*\*\*\*\*

Least Squares Means

f2	y LSMEAN	LSMEAN Number
----	----------	---------------

```

      1      9.1000000      1
      2     11.0000000      2
      3     11.5000000      3
      4     16.4000000      4

Least Squares Means for Effect f2
t for H0: LSMean(i)=LSMean(j) / Pr > |t|

```

```

Dependent Variable: y

i/j      1      2      3      4
1      -2.58557  -3.26599  -9.93404
      0.0145      0.0026  <.0001
2      2.585573  -0.68041  -7.34847
      0.0145      0.5011  <.0001
3      3.265986  0.680414  -6.66806
      0.0026      0.5011  <.0001
4      9.934042  7.348469  6.668055
      <.0001      <.0001  <.0001

```

NOTE: To ensure overall protection level, only probabilities associated with pre-planned comparisons should be used.

```

*****

Least Squares Means
Adjustment for Multiple Comparisons: Bonferroni

```

```

      f2      y LSMEAN      LSMEAN
      Number
1      9.1000000      1
2     11.0000000      2
3     11.5000000      3
4     16.4000000      4

```

```

Least Squares Means for Effect f2
t for H0: LSMean(i)=LSMean(j) / Pr > |t|

```

```

Dependent Variable: y

i/j      1      2      3      4
1      -2.58557  -3.26599  -9.93404
      0.0869      0.0156  <.0001
2      2.585573  -0.68041  -7.34847
      0.0869      1.0000  <.0001
3      3.265986  0.680414  -6.66806
      0.0156      1.0000  <.0001
4      9.934042  7.348469  6.668055
      <.0001      <.0001  <.0001

```

Les deux premiers tableaux sont comparables, le premier ne donnant que les *p-values* des tests de Student de comparaison des moyennes deux à deux, le second donnant en plus les statistiques de tests (il s'agit ici de tests multiples, non corrigés). Le troisième et dernier tableau reprend les mêmes statistiques de tests que le second, mais il ne donne pas les mêmes *p-values*, car il fait la correction de Bonferroni et utilise un niveau de test de 0,83%. De façon plus précise, il multiplie les précédentes *p-values* par 6 (nombre de différences testées), ce qui est assez discutable (c'est approximatif).

Bien entendu les deux présentations faites par `means` et par `lsmeans` sont équivalentes, bien que différentes ; ceci est vrai que ce soit sans la correction ou avec la correction de Bonferroni.

## 2.3 Cas particulier du modèle additif : premières bizarreries

Bien que les effets d'interactions soient significatifs dans ce modèle et que les effets principaux du facteur `f1` ne le soient pas, considérons néanmoins, pour illustrer notre propos, le modèle additif avec les deux facteurs et sans les interactions. Dans ce modèle, intéressons nous aux tests de comparaison des niveaux du second facteur, avec la correction de Bonferroni, et comparons encore les sorties des commandes `means` et `lsmeans`.

Voici le programme SAS :

```

proc glm data=vach;
class f1 f2;
model y = f1 f2 / ss3;
means f2;
lsmeans f2;
run;
* ----- ;
proc glm data=vach;
class f1 f2;
model y = f1 f2 / ss3 solution;
means f2 / bon;
lsmeans f2 / adj=bon tdiff;
run;

```

Et en voici les principales sorties ;

Dependent Variable: y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	290.6000000	72.6500000	19.96	<.0001
Error	35	127.4000000	3.6400000		
Corrected Total	39	418.0000000			

R-Square      Coeff Var      Root MSE      y Mean  
0.695215      15.89899      1.907878      12.00000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
f1	1	0.4000000	0.4000000	0.11	0.7422
f2	3	290.2000000	96.7333333	26.58	<.0001

\*\*\*\*\*

Level of f2	N	Mean	Std Dev
1	10	9.1000000	1.37032032
2	10	11.0000000	1.94365063
3	10	11.5000000	1.43372088
4	10	16.4000000	2.54732976

\*\*\*\*\*

Least Squares Means

f2	y LSMEAN
1	9.1000000
2	11.0000000
3	11.5000000
4	16.4000000

\*\*\*\*\*

Parameter	Estimate	Standard Error	t Value	Pr >  t
Intercept	16.30000000 B	0.67453688	24.16	<.0001
f1 1	0.20000000 B	0.60332413	0.33	0.7422
f1 2	0.00000000 B	.	.	.
f2 1	-7.30000000 B	0.85322916	-8.56	<.0001
f2 2	-5.40000000 B	0.85322916	-6.33	<.0001
f2 3	-4.90000000 B	0.85322916	-5.74	<.0001

f2      4      0.00000000 B      .      .      .

\*\*\*\*\*

Bonferroni (Dunn) t Tests for y

NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

Alpha      0.05  
Error Degrees of Freedom      35  
Error Mean Square      3.64  
Critical Value of t      2.79660  
Minimum Significant Difference      2.3861

Means with the same letter are not significantly different.

Bon Grouping	Mean	N	f2
A	16.4000	10	4
B	11.5000	10	3
B			
C B	11.0000	10	2
C			
C	9.1000	10	1

\*\*\*\*\*

Least Squares Means  
Adjustment for Multiple Comparisons: Bonferroni

f2	y LSMEAN	LSMEAN Number
1	9.1000000	1
2	11.0000000	2
3	11.5000000	3
4	16.4000000	4

Least Squares Means for Effect f2  
t for H0: LSMEAN(i)=LSMEAN(j) / Pr > |t|

Dependent Variable: y

i/j	1	2	3	4
1		-2.22683	-2.81284	-8.55573
		0.1949	0.0480	<.0001
2	2.226834		-0.58601	-6.3289
	0.1949		1.0000	<.0001
3	2.812843	0.586009		-5.74289
	0.0480	1.0000		<.0001

4	8.555732	6.328898	5.742889
	<.0001	<.0001	<.0001

Laissons de côté les résultats concernant le modèle, qui n'est manifestement pas adapté aux données, et regardons les résultats des deux commandes `means` et `lsmeans`.

En utilisation basique, sans test de comparaison, ces deux commandes redonnent exactement les mêmes résultats qu'avec le modèle complet, autrement dit on obtient, avec chacune des deux commandes, les moyennes de la variable réponse pour l'ensemble des observations réalisées dans chaque niveau du second facteur et, uniquement avec la première commande, les écarts-types des mêmes observations. On peut déjà s'étonner de ces résultats inchangés, compte tenu que les estimations des paramètres ont, elles, changé.

Regardons maintenant les résultats lorsqu'on demande, en plus, les tests de comparaison, avec la correction de Bonferroni. Avec `means`, les moyennes de la variable réponse dans les 4 niveaux de `f2` sont toujours les mêmes, mais les valeurs critiques et la différence significative minimum ont changé par rapport au modèle complet, ce qui est normal puisque le modèle a changé. Avec `lsmeans`, les moyennes de la variable réponse dans les 4 niveaux de `f2` sont encore les mêmes, mais différences et *p-values* ont également changé, pour tenir compte du changement de modèle. On pourra vérifier que ces différences sont celles obtenues en divisant l'écart entre les estimations des paramètres des effets principaux de `f2` par leur erreur-type, ce qui est logique. On pourra encore vérifier que les résultats des deux commandes sont ici parfaitement cohérents (voir la différence significative minimum dans `means` et la *p-value* du test entre les niveaux 1 et 3 dans `lsmeans`, pour une différence très proche de la valeur en question).

En conclusion, les tests de comparaison des niveaux du facteur `f2`, avec la correction de Bonferroni, sont corrects avec les deux commandes, mais les valeurs moyennes, en particulier celles appelées valeurs ajustées dans `lsmeans`, sont bizarres, dans la mesure où elles ne correspondent pas aux estimations fournies par le modèle. On pourra en effet vérifier que, dans le modèle additif, les estimations des niveaux de `f2` sont respectivement 9.0 10.9 11.4 16.3 (ajouter l'estimation de l'effet général à celle de chaque niveau du facteur). Pour terminer, on notera que les différences deux à deux entre ces deux séries d'estimations sont les mêmes, de sorte que cette bizarrerie n'a pas de consé-

quence sur les tests de comparaison.

## 2.4 Cas particulier d'un plan incomplet : nouvelles bizarreries

Pour illustrer les particularités des commandes `means` et `lsmeans` dans le cas d'un plan incomplet, nous allons considérer le cas d'un plan en blocs, incomplet et équilibré, avec un seul facteur. L'exemple traité est celui de 5 traitements (le facteur) testés sur 10 patients (les blocs). Les données et leur description sont fournies dans la section 4.

Les effets d'interactions ne pouvant pas être pris en compte dans ce cas (pas assez d'observations) et ne présentant, au demeurant, que peu d'intérêt, c'est le modèle additif qui est considéré ici, et il s'avère que ce modèle est très significatif, possède un très bon coefficient  $R^2$  (0,97) et que chacun des 2 facteurs (bloc et traitement) est très significatif. Ce modèle est donc bien adapté aux données (de plus, on peut vérifier que le graphique des résidus ne met en évidence aucune particularité).

Pour étudier les différences entre les moyennes de  $Y$  aux différents niveaux du facteur traitement, regardons ce que donne la méthode de Bonferroni avec les deux commandes `means` et `lsmeans`. Exécutons le programme SAS suivant :

```
data traitmt;
infile 'traitmt.don';
input bloc trait y;
run;
* ----- ;
proc glm data=traitmt;
class bloc trait;
model y = bloc trait / ss3 solution;
means trait / bon;
lsmeans trait / adj=bon tdiff;
run;
```

En voici les résultats :

Dependent Variable: y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	13	2091.200000	160.861538	34.88	<.0001
Error	16	73.800000	4.612500		
Corrected Total	29	2165.000000			
R-Square	Coeff Var	Root MSE	y Mean		
0.965912	4.295346	2.147673	50.00000		

Bon Grouping	Mean	N	trait
A	51.000	6	5
A			
A	50.500	6	2
A			
A	50.000	6	3
A			
A	49.500	6	4
A			
A	49.000	6	1

Source	DF	Type III SS	Mean Square	F Value	Pr > F
bloc	9	2076.200000	230.688889	50.01	<.0001
trait	4	345.200000	86.300000	18.71	<.0001

Least Squares Means  
Adjustment for Multiple Comparisons: Bonferroni

Parameter	Estimate	Standard Error	t Value	Pr >  t
Intercept	55.06666667 B	1.46714008	37.53	<.0001
bloc 1	4.80000000 B	1.86681547	2.57	0.0205
bloc 2	-2.46666667 B	1.86681547	-1.32	0.2050
bloc 3	1.33333333 B	1.81107703	0.74	0.4723
bloc 4	-3.00000000 B	1.81107703	-1.66	0.1171
bloc 5	-0.46666667 B	1.81107703	-0.26	0.7999
bloc 6	6.46666667 B	1.81107703	3.57	0.0026
bloc 7	-2.66666667 B	1.86681547	-1.43	0.1724
bloc 8	10.26666667 B	1.81107703	5.67	<.0001
bloc 9	-24.93333333 B	1.81107703	-13.77	<.0001
bloc 10	0.00000000 B	.	.	.
trait 1	-9.80000000 B	1.35830777	-7.21	<.0001
trait 2	-6.80000000 B	1.35830777	-5.01	0.0001
trait 3	-1.00000000 B	1.35830777	-0.74	0.4723
trait 4	-2.40000000 B	1.35830777	-1.77	0.0963
trait 5	0.00000000 B	.	.	.

trait	y LSMEAN	LSMEAN Number
1	44.2000000	1
2	47.2000000	2
3	53.0000000	3
4	51.6000000	4
5	54.0000000	5

Least Squares Means for Effect trait  
t for H0: LSMean(i)=LSMean(j) / Pr > |t|

Dependent Variable: y

i/j	1	2	3	4	5
1		-2.20863	-6.47865	-5.44796	-7.21486
2	2.208631		<.0001	0.0005	<.0001
3	0.4213	0.4213		-3.23932	-5.00623
4	6.47865	4.270019	0.0059		0.0013
5	<.0001	0.0059	0.0059	1.030694	-0.73621
	5.447955	3.239325	-1.03069		1.0000
	0.0005	0.0514	1.0000		-1.7669
	7.21486	5.006229	0.73621	1.766904	
	<.0001	0.0013	1.0000	0.9631	

\*\*\*\*\*  
Bonferroni (Dunn) t Tests for y

NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

Alpha	0.05
Error Degrees of Freedom	16
Error Mean Square	4.6125
Critical Value of t	3.25199
Minimum Significant Difference	4.0323

Means with the same letter are not significantly different.

Cette fois, les résultats de la commande means sont incorrects, les moyennes calculées pour chaque niveau du facteur traitement étant les moyennes de toutes les observations faites avec ce traitement, sans tenir compte de l'effet bloc. Néanmoins, la valeur critique du t et la différence significative minimum sont, pour leur part, correctes.

Si l'on regarde maintenant les résultats fournis par la commande lsmeans,

ils sont corrects en ce qui concerne les tests (réalisés avec la correction de Bonferroni). Mais, les moyennes fournies dans la colonne “y LSMEAN” sont bizarres, pour ne pas dire incompréhensibles... En effet, on s’attend à trouver ici les estimations des valeurs moyennes de  $Y$ , pour chaque traitement, fournies par le modèle considéré ; autrement dit, la somme de l’effet général, ou *intercept*, et des effets principaux des niveaux du traitement. Or ce n’est pas ce que donne SAS, l’*intercept* de 55.07 étant remplacé par 54, valeur a priori inexplicable. En fait, les “pseudo-moyennes” données ici par SAS sont déterminées à partir des estimations des effets principaux du facteur traitement dans le modèle considéré, en leur ajoutant une quantité  $y_0$  telle que la moyenne des valeurs ainsi obtenues soit égale à la moyenne générale de  $Y$  : si l’on note  $\hat{a}_j$  les estimations SAS des 5 effets principaux et  $\bar{y}$  la moyenne générale de  $Y$  sur l’ensemble des 30 observations réalisées, il vient :

$$y_0 = \bar{y} - \frac{\sum_{j=1}^5 \hat{a}_j}{5}.$$

En conclusion, déjà que l’usage de la correction de Bonferroni n’est pas vraiment judicieuse dans bien des cas, la façon dont procède SAS, avec les commandes `means` et `lsmeans` de la procédure GLM, est assez déconcertante : il convient donc de manipuler ces commandes avec beaucoup de précautions.

### 3 Usage de `lsmeans` pour les graphiques d’interactions

Nous venons de voir que la commande `lsmeans` présente certaines bizarreries. Il serait dommage de ne pas signaler ici qu’elle présente par ailleurs l’avantage de permettre la réalisation des graphiques d’interactions dans les modèles d’ANOVA à deux facteurs croisés, ou plus.

Pour illustrer cette possibilité, nous reprenons l’exemple du rendement des vaches laitières pour lequel nous allons réaliser le premier graphique d’interactions (les deux se font de la même manière).

Voici le programme SAS qui permet de faire ce graphique :

```
proc glm data=vach noprint;
class f1 f2;
```

```
model y = f1 | f2 / ss3;
lsmeans f1 | f2 / out=graph;
run;
* ----- ;
proc print data=graph;
run;
* ----- ;
proc gplot data=graph;
axis1 label=('premier facteur') order=(1 to 2 by 1)
      minor=none length=6cm;
axis2 label=('moyenne' justify=right 'des y')
      order=(6 to 22 by 2) minor=none length=6cm;
symbol1 i=join v=dot cv=black;
symbol2 i=join v=triangle cv=black;
symbol3 i=join v=circle cv=black;
symbol4 i=join v=# cv=black;
symbol5 i=join v=% cv=black;
plot lsmean*f1=f2 / haxis=axis1 vaxis=axis2;
run;
options reset=all;
quit;
```

Afin de comprendre comment se passent les choses, voici le contenu du fichier “graph” :

Obs	_NAME_	f1	f2	LSMEAN	STDERR
1	y	1	.	12.1	0.36742
2	y	2	.	11.9	0.36742
3	y	.	1	9.1	0.51962
4	y	.	2	11.0	0.51962
5	y	.	3	11.5	0.51962
6	y	.	4	16.4	0.51962
7	y	1	1	9.4	0.73485
8	y	1	2	12.0	0.73485
9	y	1	3	12.2	0.73485
10	y	1	4	14.8	0.73485
11	y	2	1	8.8	0.73485
12	y	2	2	10.0	0.73485
13	y	2	3	10.8	0.73485
14	y	2	4	18.0	0.73485

On notera que les valeurs nécessairement manquantes des colonnes `f1` et `f2` de ce fichier expliquent le message d’erreur (non fatale) obtenu systématiquement :



ment dans la fenêtre “log” de SAS au moment de la réalisation du graphique. Il ne faut donc pas tenir compte de ce message.

Le graphique obtenu est donné par la Figure A.1.

## 4 Les données “traitements”

On a souhaité expérimenter 5 traitements sur 10 patients, mais il n’est pas possible d’administrer plus de 3 traitements distincts à un patient donné, quel qu’il soit. On a donc réalisé un plan en blocs, incomplet, équilibré, en administrant 3 traitements à chaque patient, les 10 patients constituant ainsi 10 blocs. Le fichier “traitmt.don” contient :

- en première colonne, le numéro du patient, donc du bloc (codé de 1 à 10) ;
- en deuxième colonne, le numéro du traitement (codé de 1 à 5) ;
- en troisième colonne, la mesure biologique réalisée à l’issue de l’expérience.

Voici le fichier “traitmt.don” :

```
1 1 52
1 2 50.5
1 3 59.5
2 2 46
2 3 52
2 5 52
3 1 45.5
3 3 56
3 4 54.5
4 2 47
4 4 50
4 5 50
5 1 48
5 3 53
5 5 52
6 1 49
6 2 54
6 5 65
7 2 45
7 3 51
7 4 51
8 1 55.5
8 2 60.5
8 4 61
9 3 28.5
9 4 28.5
9 5 30
10 1 44
10 4 52
10 5 57
```

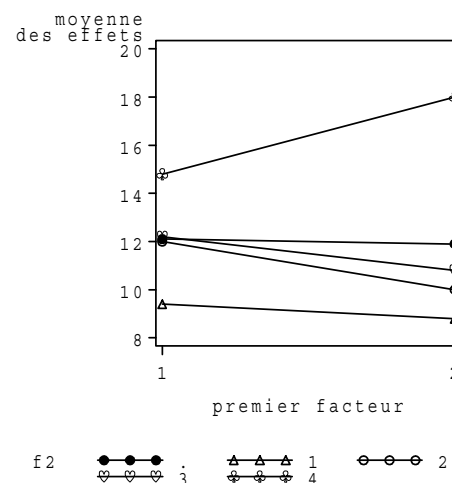


FIGURE 1 – Graphique des interactions