SLU, Biostokastikum

June 2013

## Repeated measurements

Data come from an experiment studying the level of coronary sinus potassium following coronary occlusion on dogs (Davis 2002 p. 99; Grizzle and Allen, 1969). There were four treatments with nine dogs for each treatment. Measurements were made on each dog 1, 3, 5, 7, 9, 11, 13 minutes after occlusion.

Treat Treatment number (1- untreated, 2 - cardiac denervation 3 weeks prior, 3- cardiac denervation immediately before, 4 - stellectomy 3 weeks prior)

Dog Dog number (1—36)

t Time when measurement was taken (1, 3, 5, 7, 9, 11 or 13 minutes)

y Potassium level

Analyze these data to find possible treatment differences. Note that the data are of a “repeated measures” type, which means that we have to account for correlation between the different observations within the same dog.

The data can be found in the files dogs\_data.sas or dogs.txt.

**Solution in SAS:**

We start with a model that treats time as a categorical variable. An interaction term between treatment and time should be included to allow for different temporal structures within different treatments. The covariance structure between time points is left as free as possible using an unstructured covariance matrix.

ods graphics;

**proc** **mixed** data=repeated plots=(all);

class treat dog ;

model y=treat t treat\*t /ddfm=kr;

repeated /subject=dog type=un;

**run**;

ods graphics off;

We find that the data is read correctly with 36 subjects (=dogs) and 7 measurements for each.

| **Dimensions** | |
| --- | --- |
| **Covariance Parameters** | 28 |
| **Columns in X** | 10 |
| **Columns in Z** | 0 |
| **Subjects** | 36 |
| **Max Obs Per Subject** | 7 |

We check the residuals from the analysis and find that the assumptions for the model are fulfilled.

**How to choose the covariance structure**

The covariance parameters were estimated and are given below. We estimate variances for each time point (e.g. UN(1,1) and UN(2,2) for time points 1 and 2) and covariances between time points. The covariance between time points 1 and 2 is 0.1722 and we can compute the correlation as

| **Covariance Parameter Estimates** | | |
| --- | --- | --- |
| **Cov Parm** | **Subject** | **Estimate** |
| **UN(1,1)** | dog | 0.2261 |
| **UN(2,1)** | dog | 0.1722 |
| **UN(2,2)** | dog | 0.1696 |
| **UN(3,1)** | dog | 0.1696 |
| **UN(3,2)** | dog | 0.1802 |
| **UN(3,3)** | dog | 0.4077 |
| **UN(4,1)** | dog | 0.1945 |
| **UN(4,2)** | dog | 0.1797 |
| **UN(4,3)** | dog | 0.3594 |
| **UN(4,4)** | dog | 0.4795 |
| **UN(5,1)** | dog | 0.2006 |
| **UN(5,2)** | dog | 0.1869 |
| **UN(5,3)** | dog | 0.2663 |
| **UN(5,4)** | dog | 0.4117 |
| **UN(5,5)** | dog | 0.5081 |
| **UN(6,1)** | dog | 0.1958 |
| **UN(6,2)** | dog | 0.1700 |
| **UN(6,3)** | dog | 0.1945 |
| **UN(6,4)** | dog | 0.2940 |
| **UN(6,5)** | dog | 0.4015 |
| **UN(6,6)** | dog | 0.5235 |
| **UN(7,1)** | dog | 0.1855 |
| **UN(7,2)** | dog | 0.1684 |
| **UN(7,3)** | dog | 0.2110 |
| **UN(7,4)** | dog | 0.2512 |
| **UN(7,5)** | dog | 0.3481 |
| **UN(7,6)** | dog | 0.4618 |
| **UN(7,7)** | dog | 0.5205 |

The next step would be to see if we could simplify the model. We can test a other covariance structure with fewer paramaters. AR(1) is the logical choice for time series with equidistant time points.

**proc** **mixed** data=repeated plots=(all);

class treat dog t;

model y=treat t treat\*t /ddfm=kr;

repeated /subject=dog type=ar(**1**);

**run**;

Which gives the results:

Now we need to estimate only one parameter in the covariance structure and we see that the correlation between two consecutive time points is (in average) 0.8:

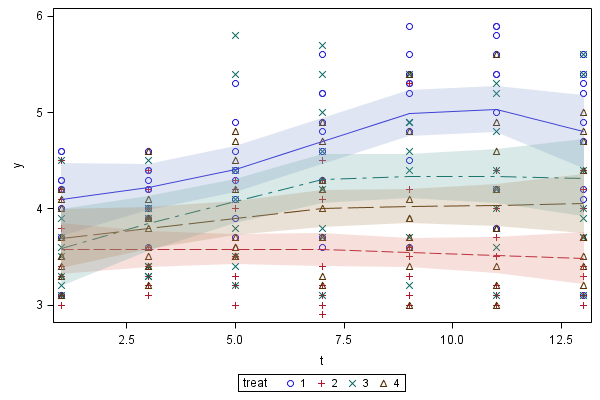
| **Covariance Parameter Estimates** | | |
| --- | --- | --- |
| **Cov Parm** | **Subject** | **Estimate** |
| **AR(1)** | dog | 0.8002 |
| **Residual** |  | 0.3943 |

We can compare the model using UN and AR(1) by AIC:

model using UN: 292.7

model using AR(1): 296.3

which means that the model using AR(1) actually has a worse fit than the model with UN. We can see an explanation for this if we plot the data (find the code in the appendix):



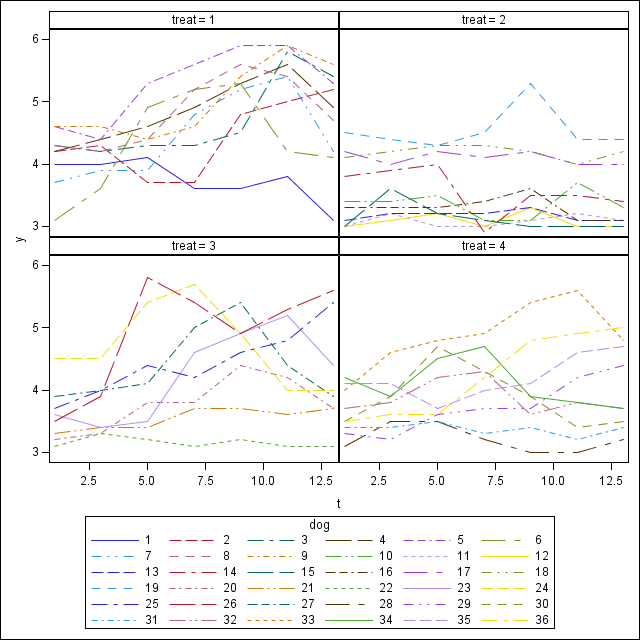
An easier type of plot to make is this

**proc** **sgpanel** data=repeated;

panelby treat;

series x=t y=y /group=dog;

**run**;



We see that the variance in time points 1 and 2 is smaller than the variance in later time points. Differences in variance are accounted for in the UN structure, but not in the AR(1) structure. A good choice of model might be a ARH(1) model that gives us both the advantage of estimating a single correlation coefficient and let the variances in different time points be different. The AIC for the ARH(1) model is the lowest of all three models.

**Time as continuous or class variable**

The time development does not look very linear (check the plots above), at least for one of the treatments. To use a linear time relationship might not be meaningful in this case but a quadratic relationship would be possible. If we want to do so, we exclude t from the class list and create a variable that can represent the quadratic term:

**data** repeated1;

set repeated;

t2=t\*t;

ods graphics;

**proc** **mixed** data=repeated1 plots=(all);

class treat dog ;

model y=treat t t2 /ddfm=kr;

repeated /subject=dog type=arh(**1**);

**run**;

We get very similar fits using time as class variable and using a quadratic fit for time (the quadratic fit is worse, interactions between treatment and time are not significant). We leave this approach and continue with time as class variable.

**Comparing the 4 treatments**

To answer our question about differences between treatments we can compute LSMeans and make pairwise comparisons. We also choose to adjust for multiple testing using Tukeys method.

**proc** **mixed** data=repeated;

class treat dog t;

model y=treat t treat\*t;

repeated /subject=dog type=arh(**1**);

lsmeans treat\*t /pdiff adjust=tukey;

**run**;

The result is very hard to overview and many of the p-values are very high. There are some significant differences, e.g. if we compare treatment 1 time point 9 with treatment 4 time point 1, but this is of course completely irrelevant. We notice that we have too much output and that we overcorrect using SAS Tukey method in this case, since it corrects for nearly 400 comparisons out of which very many are not interesting.

We start by trying to get an overview using letter grouping. This is not implemented in PROC MIXED but we can use a SAS macro that is available on: <http://www.stat.lsu.edu/faculty/geaghan/pdmix800.sas.txt> .

First we need to complete our mixed program with an ods output step to save estimated differences and lsmeans to new dataset:

**proc** **mixed** data=repeated plots=(all);

class treat dog t;

model y=treat t treat\*t;

repeated /subject=dog type=arh(**1**);

lsmeans treat\*t/pdiff adjust=tukey ;

ods output diffs=diffs lsmeans=lsm;

**run**;

Then you can copy the code of the macro into the editor and run it or, better, save it to a file and call the file from SAS by (remember to change the path):

%include 'Z:\my documents\pdmix800\_sas.txt';

Now we can run the macro by:

%***pdmix800***(diffs,lsm,sort=yes);

and get:

| **Obs** | **treat** | **t** | **Estimate** | **Standard Error** | **Letter Group** |
| --- | --- | --- | --- | --- | --- |
| **1** | 1 | 11 | 5.2222 | 0.2367 | A |
| **2** | 1 | 9 | 5.0667 | 0.2423 | AB |
| **3** | 1 | 13 | 4.7222 | 0.2291 | BC |
| **4** | 1 | 7 | 4.6556 | 0.2392 | CD |
| **5** | 3 | 9 | 4.5000 | 0.2570 | BCDE |
| **6** | 3 | 7 | 4.4375 | 0.2537 | BCDE |
| **7** | 1 | 5 | 4.4000 | 0.2330 | CDE |
| **8** | 3 | 11 | 4.3250 | 0.2510 | CDEF |
| **9** | 3 | 13 | 4.2250 | 0.2430 | CDEFG |
| **10** | 3 | 5 | 4.2000 | 0.2472 | CDEFGH |
| **11** | 1 | 3 | 4.1778 | 0.1454 | EF |
| **12** | 1 | 1 | 4.1111 | 0.1578 | EFGI |
| **13** | 4 | 7 | 4.0667 | 0.2392 | DEFGIJ |
| **14** | 4 | 11 | 4.0556 | 0.2367 | DEFGHIJ |
| **15** | 4 | 13 | 4.0444 | 0.2291 | DEFGHIJ |
| **16** | 4 | 5 | 4.0111 | 0.2330 | DEFGIJ |
| **17** | 4 | 9 | 3.9778 | 0.2423 | EFGHIJ |
| **18** | 4 | 3 | 3.7778 | 0.1454 | FGHIJ |
| **19** | 3 | 3 | 3.7250 | 0.1542 | IJK |
| **20** | 2 | 9 | 3.6600 | 0.2299 | FGHIJ |
| **21** | 4 | 1 | 3.6444 | 0.1578 | HK |
| **22** | 2 | 3 | 3.6300 | 0.1379 | JK |
| **23** | 2 | 5 | 3.6200 | 0.2211 | GHIJ |
| **24** | 3 | 1 | 3.6000 | 0.1673 | JK |
| **25** | 2 | 1 | 3.5400 | 0.1497 | JK |
| **26** | 2 | 11 | 3.5000 | 0.2245 | JK |
| **27** | 2 | 7 | 3.4600 | 0.2269 | JK |
| **28** | 2 | 13 | 3.4600 | 0.2174 | JK |

This is of course also hard to overview and of more use if look at main factors but it gives at least some sorting and a feeling about where the significant differences lie.

Another possibility is to extract the interesting comparisons and compare them using a simple Bonferroni adjustment. We start by extracting the comparisons between the different treatments at the same time point. First we save all comparisons in a new file called diffs (you have already done that above, but here is the program once more):

**proc** **mixed** data=repeated;

class treat dog t;

model y=treat t treat\*t;

repeated /subject=dog type=arh(**1**);

lsmeans treat\*t /pdiff ;

ods output diffs=diffs;

**run**;

We select the comparisons that are relevant, in this case if treat has differernt values but t has the same.

**data** diffs1;

set diffs;

if treat ne \_treat & t = \_t;

**run**;

We look at the differences and check if they are significant at a significance level adjusted with Bonferronis method: We make 42 relevant comparisons and compute the new significance level to 0.05/42=0.0012. We find significant difference mainly between treatment 1 and 2 (at time point 7 and later)

**proc** **print** data=diffs1;

**run**;

| **Obs** | **Effect** | **treat** | **t** | **\_treat** | **\_t** | **Estimate** | **StdErr** | **DF** | **tValue** | **Probt** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | treat\*t | 1 | 1 | 2 | 1 | 0.5711 | 0.2175 | 192 | 2.63 | 0.0093 |
| **2** | treat\*t | 1 | 1 | 3 | 1 | 0.5111 | 0.2300 | 192 | 2.22 | 0.0274 |
| **3** | treat\*t | 1 | 1 | 4 | 1 | 0.4667 | 0.2231 | 192 | 2.09 | 0.0378 |
| **4** | treat\*t | 1 | 3 | 2 | 3 | 0.5478 | 0.2004 | 192 | 2.73 | 0.0068 |
| **5** | treat\*t | 1 | 3 | 3 | 3 | 0.4528 | 0.2119 | 192 | 2.14 | 0.0339 |
| **6** | treat\*t | 1 | 3 | 4 | 3 | 0.4000 | 0.2056 | 192 | 1.95 | 0.0531 |
| **7** | treat\*t | 1 | 5 | 2 | 5 | 0.7800 | 0.3212 | 192 | 2.43 | 0.0161 |
| **8** | treat\*t | 1 | 5 | 3 | 5 | 0.2000 | 0.3397 | 192 | 0.59 | 0.5567 |
| **9** | treat\*t | 1 | 5 | 4 | 5 | 0.3889 | 0.3296 | 192 | 1.18 | 0.2394 |
| **10** | treat\*t | 1 | 7 | 2 | 7 | 1.1956 | 0.3297 | 192 | 3.63 | 0.0004 |
| **11** | treat\*t | 1 | 7 | 3 | 7 | 0.2181 | 0.3487 | 192 | 0.63 | 0.5325 |
| **12** | treat\*t | 1 | 7 | 4 | 7 | 0.5889 | 0.3383 | 192 | 1.74 | 0.0833 |
| **13** | treat\*t | 1 | 9 | 2 | 9 | 1.4067 | 0.3340 | 192 | 4.21 | <.0001 |
| **14** | treat\*t | 1 | 9 | 3 | 9 | 0.5667 | 0.3533 | 192 | 1.60 | 0.1103 |
| **15** | treat\*t | 1 | 9 | 4 | 9 | 1.0889 | 0.3427 | 192 | 3.18 | 0.0017 |
| **16** | treat\*t | 1 | 11 | 2 | 11 | 1.7222 | 0.3262 | 192 | 5.28 | <.0001 |
| **17** | treat\*t | 1 | 11 | 3 | 11 | 0.8972 | 0.3450 | 192 | 2.60 | 0.0100 |
| **18** | treat\*t | 1 | 11 | 4 | 11 | 1.1667 | 0.3347 | 192 | 3.49 | 0.0006 |
| **19** | treat\*t | 1 | 13 | 2 | 13 | 1.2622 | 0.3159 | 192 | 4.00 | <.0001 |
| **20** | treat\*t | 1 | 13 | 3 | 13 | 0.4972 | 0.3340 | 192 | 1.49 | 0.1383 |
| **21** | treat\*t | 1 | 13 | 4 | 13 | 0.6778 | 0.3241 | 192 | 2.09 | 0.0378 |
| **22** | treat\*t | 2 | 1 | 3 | 1 | -0.06000 | 0.2245 | 192 | -0.27 | 0.7896 |
| **23** | treat\*t | 2 | 1 | 4 | 1 | -0.1044 | 0.2175 | 192 | -0.48 | 0.6316 |
| **24** | treat\*t | 2 | 3 | 3 | 3 | -0.09500 | 0.2069 | 192 | -0.46 | 0.6466 |
| **25** | treat\*t | 2 | 3 | 4 | 3 | -0.1478 | 0.2004 | 192 | -0.74 | 0.4617 |
| **26** | treat\*t | 2 | 5 | 3 | 5 | -0.5800 | 0.3316 | 192 | -1.75 | 0.0819 |
| **27** | treat\*t | 2 | 5 | 4 | 5 | -0.3911 | 0.3212 | 192 | -1.22 | 0.2249 |
| **28** | treat\*t | 2 | 7 | 3 | 7 | -0.9775 | 0.3404 | 192 | -2.87 | 0.0045 |
| **29** | treat\*t | 2 | 7 | 4 | 7 | -0.6067 | 0.3297 | 192 | -1.84 | 0.0673 |
| **30** | treat\*t | 2 | 9 | 3 | 9 | -0.8400 | 0.3448 | 192 | -2.44 | 0.0158 |
| **31** | treat\*t | 2 | 9 | 4 | 9 | -0.3178 | 0.3340 | 192 | -0.95 | 0.3426 |
| **32** | treat\*t | 2 | 11 | 3 | 11 | -0.8250 | 0.3368 | 192 | -2.45 | 0.0152 |
| **33** | treat\*t | 2 | 11 | 4 | 11 | -0.5556 | 0.3262 | 192 | -1.70 | 0.0902 |
| **34** | treat\*t | 2 | 13 | 3 | 13 | -0.7650 | 0.3261 | 192 | -2.35 | 0.0200 |
| **35** | treat\*t | 2 | 13 | 4 | 13 | -0.5844 | 0.3159 | 192 | -1.85 | 0.0658 |
| **36** | treat\*t | 3 | 1 | 4 | 1 | -0.04444 | 0.2300 | 192 | -0.19 | 0.8470 |
| **37** | treat\*t | 3 | 3 | 4 | 3 | -0.05278 | 0.2119 | 192 | -0.25 | 0.8036 |
| **38** | treat\*t | 3 | 5 | 4 | 5 | 0.1889 | 0.3397 | 192 | 0.56 | 0.5788 |
| **39** | treat\*t | 3 | 7 | 4 | 7 | 0.3708 | 0.3487 | 192 | 1.06 | 0.2889 |
| **40** | treat\*t | 3 | 9 | 4 | 9 | 0.5222 | 0.3533 | 192 | 1.48 | 0.1410 |
| **41** | treat\*t | 3 | 11 | 4 | 11 | 0.2694 | 0.3450 | 192 | 0.78 | 0.4358 |
| **42** | treat\*t | 3 | 13 | 4 | 13 | 0.1806 | 0.3340 | 192 | 0.54 | 0.5895 |

**Appendix:**

Program to create the picture giving a mean function per treatment:

**proc** **sort** data=repeated;

by treat t;

**proc** **loess** data=repeated;

by treat;

model y = t / clm;

score /clm;

ods output ScoreResults=SR;

**run**;

**proc** **sgplot** data=SR;

scatter x=t y=y / group=treat;

series x=t y=p\_y / group=treat;

band x=t lower=lcl\_y upper=ucl\_y / group=treat transparency=**0.75**;

**run**;