Auxiliary Lecture 5: Time Series Data - An Example

LSA 2013, LI539
Mixed Effect Models

T. Florian Jaeger

Brain and Cognitive Sciences
University of Rochester

July 17, 2013

Data

- Our example data comes from a 2x2 design, where both factors and their interaction are **within-subjects** and **within-items**.

```
d = read.csv(file = "eye-tracking-sample.csv")
str(d)
```

```
> 'data.frame':   35236 obs. of  11 variables:
 $ Subj                   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Item                   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Sample                 : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Time                   : num  0.004 0.008 0.012 0.016 0.02 0.024 ...
 $ cTime                  : num  -0.198 -0.194 -0.19 -0.186 -0.182 ...
 $ Bin                    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ CondWordFrequency      : Factor w/ 2 levels "high","low": 1 1 1 1 1 1 ...
 $ cCondWordFrequencyHigh: num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
 $ CondCompetitors        : Factor w/ 2 levels "one","two": 2 2 2 2 2 ...
 $ cCondCompetitionHigh   : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
 $ LooksToTarget          : int  0 0 1 1 0 0 1 0 1 1 ...
```

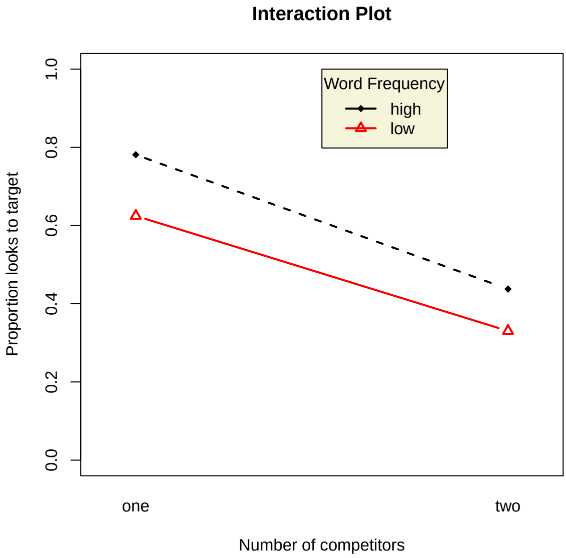
Repeated measures ANOVA for 2x2 within-factors

- Let's start with the F1 analysis (by-participant). For now we collapse over time.
- For quick help for ANOVA in R, see `http://www.statmethods.net/stats/anova.html`

```
d.agg = aggregate(d[,c('LooksToTarget')], by= list(
      Subj = d$Subj,
      CondWordFrequency = d$CondWordFrequency,
      CondCompetitors = d$CondCompetitors
   ), FUN = mean
)
# This variables stores average proportions of looks to target
# (in the original data LooksToTarget was 1 for every sample
# for which the fixation was on the target and zero otherwise
names(d.agg)[length(names(d.agg))] = "LooksToTarget"
str(d.agg)
```

```
> 'data.frame':   64 obs. of  4 variables:
 $ Subj             : int  1 2 3 4 5 6 7 8 9 10 ...
 $ CondWordFrequency: Factor w/ 2 levels "high","low": 1 1 1 1 1 ...
 $ CondCompetitors  : Factor w/ 2 levels "one","two": 1 1 1 1 1 ...
 $ LooksToTarget    : num  0.815 0.784 0.772 0.806 0.561 ...
```

Visualizing the data



F1 and F2 Results

```
m.F1 = aov(LooksToTarget ~ CondWordFrequency * CondCompetitors +
    Error(Subj/(CondWordFrequency * CondCompetitors)),
    data = d.agg)
summary(m.F1)
```

- ... and similarly for F2 (by aggregating by item) ...

```
[F1:]
                                  Df  Sum Sq Mean Sq F value    Pr(>F)
CondWordFrequency                  1 0.08428 0.08428  3.9443 0.0519329 .
CondCompetitors                    1 0.36136 0.36136 16.9112 0.0001295 ***
CondWordFrequency:CondCompetitors  1 0.01749 0.01749  0.8187 0.3694427
Residuals                         56 1.19661 0.02137

[F2]
                                   Df Sum Sq Mean Sq F value    Pr(>F)
CondWordFrequency                   1 0.2626 0.26261 10.9929  0.001004 **
CondCompetitors                     1 2.1119 2.11193 88.4064 < 2.2e-16 ***
CondWordFrequency:CondCompetitors   1 0.0415 0.04150  1.7373  0.188284
Residuals                         376 8.9822 0.02389
```

Mixed Linear Model

- Here, we are aggregating by subject and item at the same time.

```
d.agg = aggregate(d[,c('LooksToTarget')], by= list(
        Subj = d$Subj,
        Item = d$Item,
        CondWordFrequency = d$CondWordFrequency,
        CondCompetitors = d$CondCompetitors
    ), FUN = mean
)
# This variables stores average proportions of looks to target
names(d.agg)[length(names(d.agg))] = "LooksToTarget"
```

Coding the two Factors

- Here, I am first dummy coding and then centering the predictors.
- That's essentially the same as contrast/sum-coding the predictors, which is sometimes also called ANOVA coding.

```
d.agg$cCondWordFrequencyHigh =
    myCenter(ifelse(d.agg$CondWordFrequency == "high", 1, 0))
d.agg$cCondCompetitorsTwo =
    myCenter(ifelse(d.agg$CondCompetitors == "two", 1, 0))
```

Why center?

- Higher-order terms (e.g. interactions or higher order terms of polynomials) are likely to be collinear with the lower order effects.

Using ANOVA-coding

- For a highly balanced data set like the current one (check it for yourself), I could just sum-code the predictors.

```
contrasts(d.agg$CondWordFrequency) = contr.sum(2)
contrasts(d.agg$CondCompetitors) = contr.sum(2)
```

NB: R will assign the value 1 to the alphabetically first level of the factor and -1 to the second level

NB: Under this coding the distance between the two conditions is *two* units.

```
contrasts(d.agg$CondWordFrequency)
```

```
     [,1]
high    1
low    -1
```

- For lots of information on predictor coding, see Maureen Gillespie's tutorial:
  `http://wiki.bcs.rochester.edu:2525/HlpLab/StatsCourses ?action=AttachFile&do=get&target=gillespie-tutorial.pdf`

Mixed Linear Model

- We start with a model with a **full random effect structure** yielding:

```
Formula: LooksToTarget ~ cCondWordFrequencyHigh * cCondCompetitorsTwo +
    (1 + cCondWordFrequencyHigh * cCondCompetitorsTwo | Subj) +
    (1 + cCondWordFrequencyHigh * cCondCompetitorsTwo | Item)
   Data: d.agg
   AIC    BIC logLik deviance REMLdev
 -1016 -917.6  533.2    -1094   -1066
Random effects:
 Groups Name                        Variance  Std.Dev. Corr
 Item   (Intercept)                 0.004607 0.067880
        cCondWordFrequencyHigh      0.000439 0.020973 -0.503
        cCondCompetitorsTwo         0.000479 0.021897  0.282 -0.392
        cCondWFreqHigh:cCondCompTwo 0.000720 0.026849  0.830  0.012 -0.142
 Subj   (Intercept)                 0.017818 0.133484
        cCondWordFrequencyHigh      0.000188 0.013739 -1.000
        cCondCompetitorsTwo         0.001281 0.035795  0.777 -0.777
        cCondWFreqHigh:cCondCompTwo 0.002792 0.052843  0.778 -0.778  0.583
 Residual                           0.001919 0.043816
Number of obs: 384, groups: Item, 24; Subj, 16

Fixed effects:
                                           Estimate Std. Error t value
(Intercept)                                 0.54356    0.03620  15.015
cCondWordFrequencyHigh                      0.13165    0.00708  18.596
cCondCompetitorsTwo                        -0.31924    0.01096 -29.136
cCondWordFrequencyHigh:cCondCompetitorsTwo -0.04928    0.01687  -2.921
[...]
```

Weighted linear regression over empirical logits

```
Formula: emplog(LooksToTarget, TotalLooks$x) ~
cCondWordFrequencyHigh * cCondCompetitorsTwo +
(1 + cCondWordFrequencyHigh * cCondCompetitorsTwo | Subj) +
(1 + cCondWordFrequencyHigh * cCondCompetitorsTwo | Item)
   Data: d.agg
  AIC  BIC logLik deviance REMLdev
 1256 1355   -603     1190    1206
Random effects:
 Groups Name                       Variance   Std.Dev.  Corr
 Item   (Intercept)                5.9687e-03 0.0772574
        cCondWordFrequencyHigh     5.4188e-04 0.0232783 -0.077
        cCondCompetitorsTwo        6.7981e-04 0.0260731 -0.172 -0.413
        cCondWFreqHigh:cCondCompTwo 1.3549e-04 0.0116402  0.140  0.968 -0.331
 Subj   (Intercept)                2.3187e-02 0.1522720
        cCondWordFrequencyHigh     3.6433e-06 0.0019087 1.000
        cCondCompetitorsTwo        2.8312e-04 0.0168262 1.000  1.000
        cCondWFreqHigh:cCondCompTwo 2.5864e-04 0.0160822 0.087  0.087  0.087
 Residual                          4.2018e-02 0.2049839
[...]
                                            Estimate Std. Error t value
(Intercept)                                 0.218989   0.041284    5.30
cCondWordFrequencyHigh                      0.651736   0.006942   93.89
cCondCompetitorsTwo                        -1.483138   0.008490 -174.70
cCondWordFrequencyHigh:cCondCompetitorsTwo -0.377246   0.011098  -33.99
[...]
```

Are all random effects justified?

- Fit reduced model with just random intercepts for subject and item and compare it to full model:

```
m.full = lmer(emplog(LooksToTarget, TotalLooks$x) ~
    cCondWordFrequencyHigh * cCondCompetitorsTwo +
    (1 + cCondWordFrequencyHigh * cCondCompetitorsTwo | Subj) +
    (1 + cCondWordFrequencyHigh * cCondCompetitorsTwo | Item),
    d.agg, family = "gaussian",
    weight = emplogweight(LooksToTarget, TotalLooks$x)
)
m.simple = lmer(emplog(LooksToTarget, TotalLooks$x) ~
    cCondWordFrequencyHigh * cCondCompetitorsTwo +
    (1 | Subj) +
    (1 | Item),
    d.agg, family = "gaussian",
    weight = emplogweight(LooksToTarget, TotalLooks$x)
)
anova(m.full,m.simple)
```

```
Models:
[...]
         Df    AIC    BIC  logLik  Chisq Chi Df Pr(>Chisq)
m.simple  7 1223.3 1250.9 -604.63
m.full   25 1239.8 1338.6 -594.91 19.446     18     0.3648
```

Question

The full random effect structure does not result in model that fits the data significantly better given the increase in complexity (number of parameter). Can we just stop here?

‘Maximal random effect structure justified by the data’

- Based on model comparison, we find that the following model contains the **maximal random effect structure justified by the data** (see script for details):

```
emplog(LooksToTarget, TotalLooks$x) ~
cCondWordFrequencyHigh * cCondCompetitorsTwo +
(1 + cCondCompetitorsTwo | Subj) +
(1 + cCondWordFrequencyHigh + cCondCompetitorsTwo | Item)

  AIC  BIC logLik deviance REMLdev
 1235 1290 -603.5     1191    1207
Random effects:
 Groups   Name                   Variance   Std.Dev. Corr
 Item     (Intercept)            0.00599874 0.077452
          cCondWordFrequencyHigh 0.00056475 0.023764 -0.058
          cCondCompetitorsTwo    0.00067979 0.026073 -0.181 -0.464
 Subj     (Intercept)            0.02310328 0.151998
          cCondCompetitorsTwo    0.00028816 0.016975 1.000
 Residual                        0.04252746 0.206222
Number of obs: 384, groups: Item, 24; Subj, 16
[...]
```

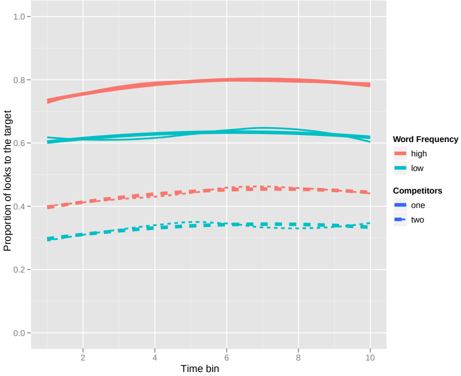
‘Maximal random effect structure justified by the data’


```
[...]
Fixed effects:
                                          Estimate Std. Error t value
(Intercept)                               0.219100   0.041236    5.31
cCondWordFrequencyHigh                    0.652125   0.007010   93.03
cCondCompetitorsTwo                      -1.483316   0.008526 -173.98
cCondWordFrequencyHigh:cCondCompetitorsTwo -0.377856   0.010117  -37.35

Correlation of Fixed Effects:
            (Intr) cCnWFH cCndCT
cCndWrdFrqH -0.012
cCndCmpttrT  0.412 -0.251
cCndWFH:CCT -0.006 -0.063  0.051
```

The Time Course

- Local smoother in general additive model (thin lines) and quadratic fit in binomial GLM (thick lines) for the four conditions over the time bins:



Mixed Logit Model


```
Formula: LooksToTarget ~ cCondWordFrequencyHigh * cCondCompetitorsTwo *
   pol(cTime, 2) + (1 | Subj) + (1 | Item)
   Data: d
   AIC   BIC logLik deviance
 40765 40883 -20368    40737
Random effects:
 Groups Name        Variance Std.Dev.
 Item   (Intercept) 0.11565  0.34007
 Subj   (Intercept) 0.42845  0.65456
Number of obs: 35236, groups: Item, 24; Subj, 16

Fixed effects:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                       0.292108   0.178705    1.63   0.1021
cCondWordFrequencyHigh            0.703270   0.036675   19.18  < 2e-16
cCondCompetitorsTwo              -1.538732   0.036954  -41.64  < 2e-16
pol(cTime, 2)cTime                0.579599   0.104796    5.53 3.19e-08
pol(cTime, 2)cTime^2             -4.536325   1.016127   -4.46 8.03e-06
cCondWFqHigh:cCondCompetitorsTwo -0.476465   0.073144   -6.51 7.31e-11
cCondWFqHigh:cTime                0.418418   0.209567    2.00   0.0459
cCondWFqHigh:cTime^2             -1.857847   2.032027   -0.91   0.3606
cCondCompTwo:cTime               -0.004828   0.209584   -0.02   0.9816
cCondCompTwo:cTime^2              0.671621   2.032296    0.33   0.7410
cCondWFqHigh:cCondCompTwo:cTime  -0.574247   0.419159   -1.37   0.1707
cCondWFqHigh:cCondCompTwo:cTime^2 4.997924   4.064680    1.23   0.2188
```

Disclaimer

- Generated data
  - $\alpha = .3$
  - $\beta_{WordFrequencyHigh} = .7$
  - $\beta_{CompetitorsTwo} = -1.5$
  - $\beta_{WordFrequencyHigh:CompetitorsTwo} = -.3$
  - $\beta_{time} = .5$ and $\beta_{time^2} = -1.5$
  - $\beta_{WordFrequencHigh:Time} = .4$
  - $\beta_{CompetitorsTwo:Time} = .1$
  - $\beta_{WordFrequencHigh:CompetitorsTwo:Time} = -.8$
  - $\sigma_{\alpha_{Subject}} = 0.5$ and $\sigma_{\alpha_{Item}} = 0.15$
- With data loss rates differing between individual participants ($\mu = 3\%$)