Linear model parameters

Model parameters: definitions

- Parameters of a linear model typically characterize *differences* in means; differences per unit of change for continuous predictors, differences between groups (or between group averages) for categorical predictors
- Interactions are **differences between differences**

Coding for categorical predictors: contrasts

- What do the parameters of a linear model mean?
- Start with categorical variables, because they're potentially more confusing ("intercept and slope" isn't too hard)
- Default R behaviour: *treatment contrasts*
  - β_1 = expected value in baseline group (= first level of the factor variable, by default the first in alphabetical order);
  - β_i = expected difference between group i and the first group.

Example

The previously explored ant-colony example:

Define data:

```
forest <- c(9, 6, 4, 6, 7, 10)
field  <- c(12, 9, 12, 10)
ants <- data.frame(
  place=rep(c("field","forest"),
            c(length(field), length(forest))),
  colonies=c(field,forest)
)
## utility function for pretty printing
pr <- function(m) printCoefmat(coef(summary(m)),
                        digits=3,signif.stars=FALSE)
```

```
pr(lm1 <- lm(colonies~place,data=ants))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10.75       0.98   10.97  4.2e-06
## placeforest    -3.75       1.27   -2.96    0.018
```

- The `(Intercept)` row refers to β_1, which is the mean density in the "field" sites ("field" comes before "forest").
- The `placeforest` row indicates we are looking at the effect of `forest` level of the `place` variable, i.e. the difference between "forest" and "field" sites. (To know that "field" is the baseline level we must (1) remember, or look at `levels(ants$place)` or (2) notice which level is *missing* from the list of parameter estimates.)

R's behaviour may seem annoying at first – it seems like the estimated values of the groups are what we're really interested in – but it is really designed for testing *differences among groups*. To get the estimates per group, you could:

- use a regression formula `colonies~place-1`, or equivalently `colonies~place+0`, to suppress the implicit intercept term:

```
##              Estimate Std. Error t value Pr(>|t|)
## placefield      10.75       0.98   10.97  4.2e-06
## placeforest      7.00       0.80    8.75  2.3e-05
```

When you use the `colonies~place-1` formula, the meanings of the parameters change: β_1 is the same (mean of "field"), but β_2 is 'mean of "field"' rather than ("(field)-(forest)").

- Use the `predict` function:

```
predict(lm1,newdata=data.frame(place=c("field","forest")),
        interval="confidence")
```

- Use the `effects` package:

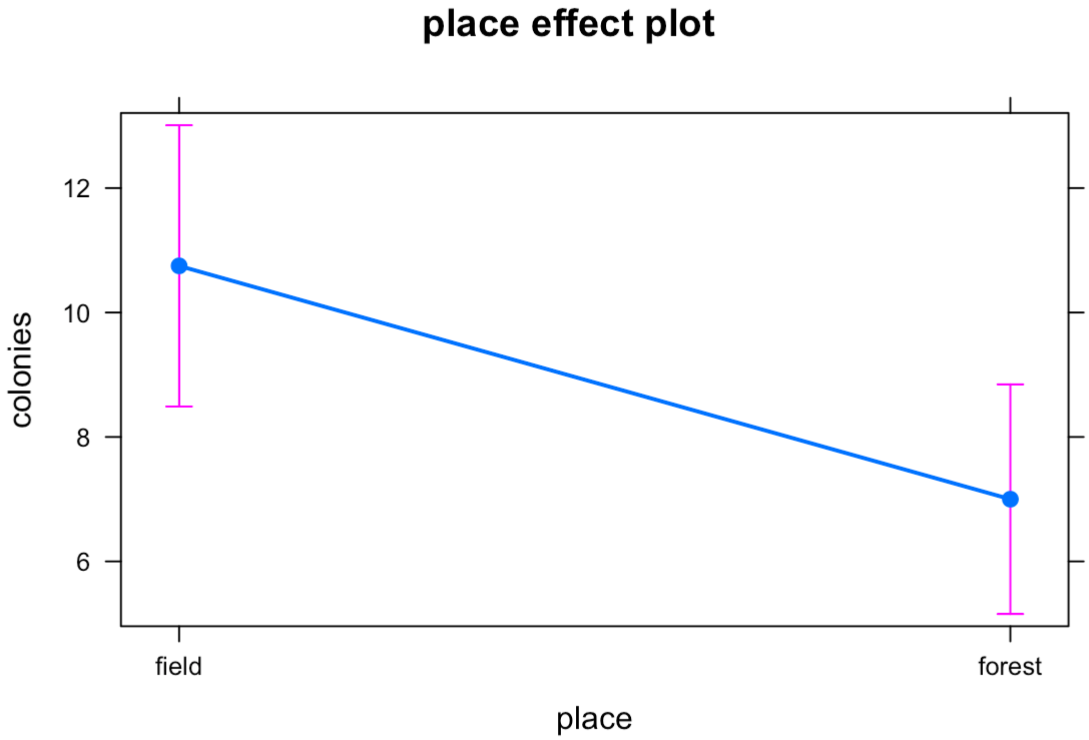
```
library("effects")
summary(allEffects(lm1))
```

- Use the `lsmeans` package:

```
library("lsmeans")
lsmeans(lm1,specs=~place)
```

Graphical summaries:

```
plot(allEffects(lm1))
```



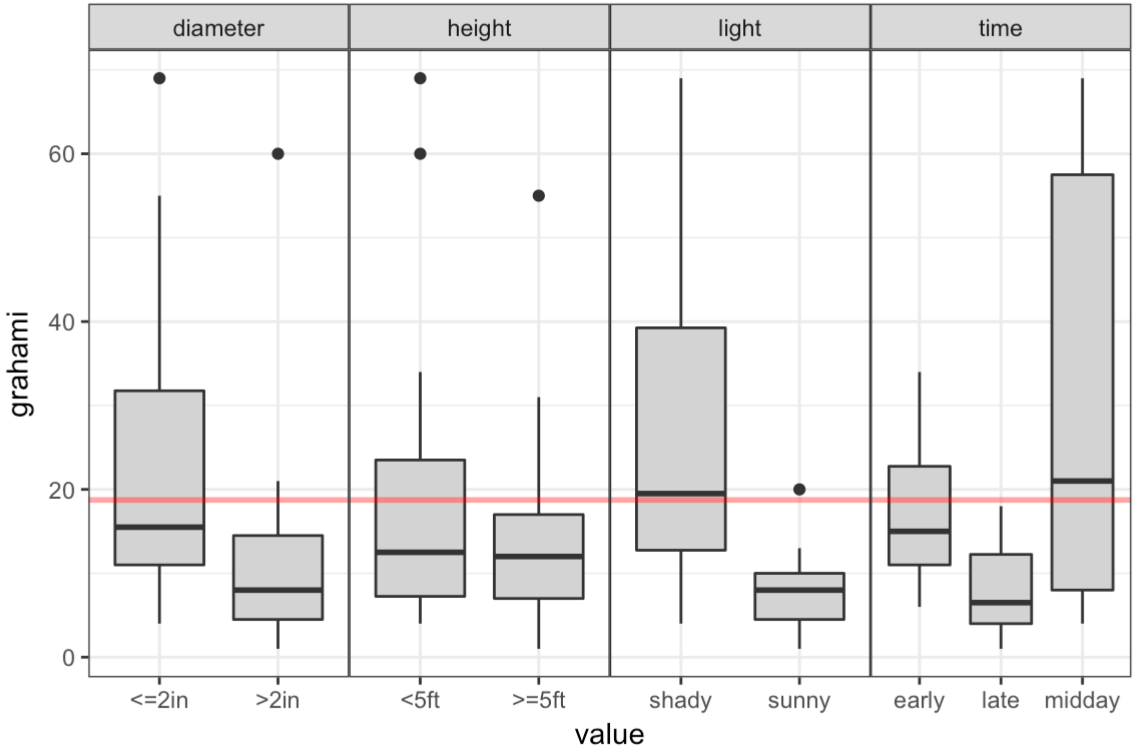
8/21

More than two levels

Some data on lizard perching behaviour (`brglm` package; Schoener 1970 *Ecology* **51**:408-418).

```
lizards <- read.csv("lizards.csv")
```

Response is number of *Anolis grahami* lizards found on perches in particular conditions.


diameter
height
light
time
grahami
60
40
20
0
<=2in
>2in
<5ft
>=5ft
shady
sunny
early
late
midday
value

Start with the `time` variable.

If we leave the factors alphabetical then β_1="early", β_2="late"-"early", β_3="midday"-"early". Change the order of the levels:

```
lizards <- mutate(lizards,
        time=factor(time,
            levels=c("early","midday","late")))
```

This just swaps the definitions of β_2 and β_3.

We could also use *sum-to-zero* contrasts:

```
pr(lm(grahami~time,data=lizards,contrasts=list(time=contr.sum)))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    19.30       3.53    5.47  2.4e-05
## time1          -1.67       4.93   -0.34     0.74
## time2          12.85       5.10    2.52     0.02
```

Here, each coefficient compares the corresponding level of the factor to the average of the other levels.

Now the `(Intercept)` parameter is the overall mean: `time1` and `time2` are the deviations of the first ("early") and second ("midday") groups from the overall mean. (See also `car::contr.Sum`.)

There are other ways to change the contrasts (i.e., use the `contrasts()` function to change the contrasts for a particular variable permanently.

There are other options for contrasts such as `MASS::contr.sdif()`, which gives the successive differences between levels.

```
library("MASS")
pr(lm(grahami~time,data=lizards,contrasts=list(time=contr.sdif)))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    19.30       3.53    5.47  2.4e-05
## time2-1        14.52       8.74    1.66    0.112
## time3-2       -24.02       8.74   -2.75    0.012
```

You might have particular contrasts in mind (e.g. "control" vs. all other treatments, then "low" vs "high" within treatments), in which case it is probably worth learning how to set contrasts.

Multiple treatments and interactions

Additive model

Consider the `light` variable in addition to `time`.

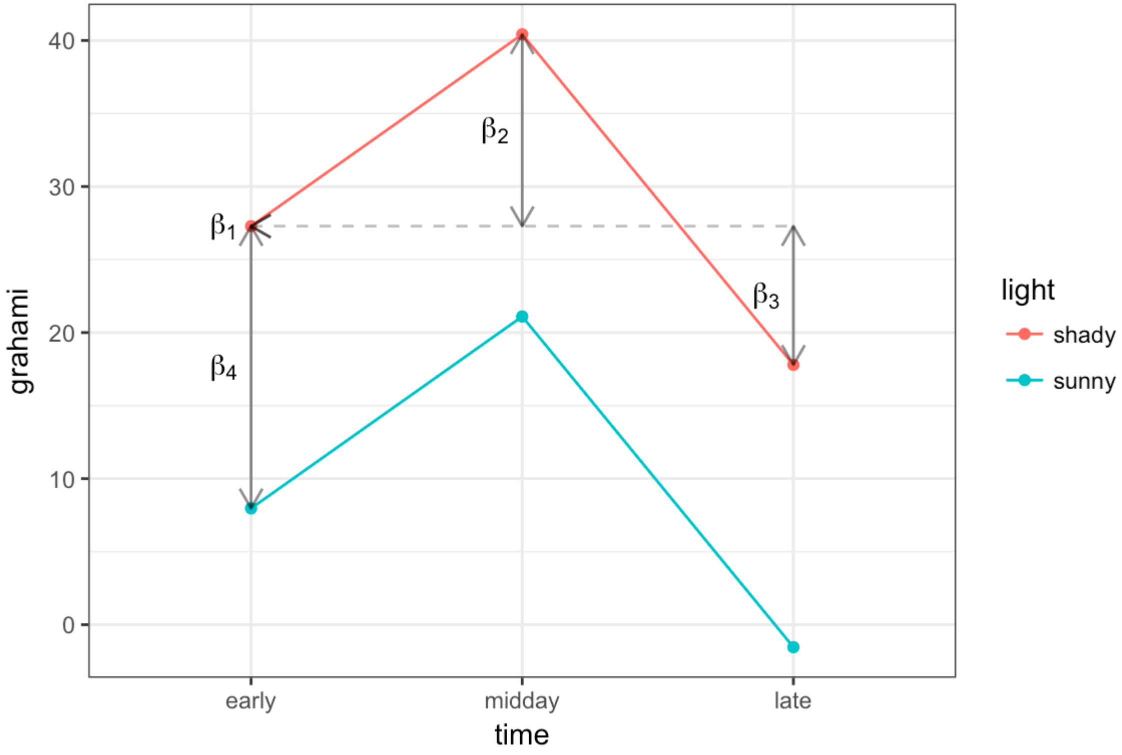
```
pr(lmTL1 <- lm(grahami~time+light,data=lizards))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    27.29       5.63    4.85  0.00011
## timemidday     13.14       7.11    1.85  0.08010
## timelate       -9.50       6.85   -1.39  0.18174
## lightsunny    -19.32       5.73   -3.37  0.00321
```

β_1 is the intercept ("early","sunny"); β_2 and β_3 are the differences from the baseline level ("early") of the *first* variable (`time`) in the *baseline* level of the other parameter(s) (`light`="shady"); β_4 is the difference from the baseline level ("sunny") of the *second* variable (`light`) in the *baseline* level of `time` ("early").

Graphical interpretation

```
## Loading required package: grid
```



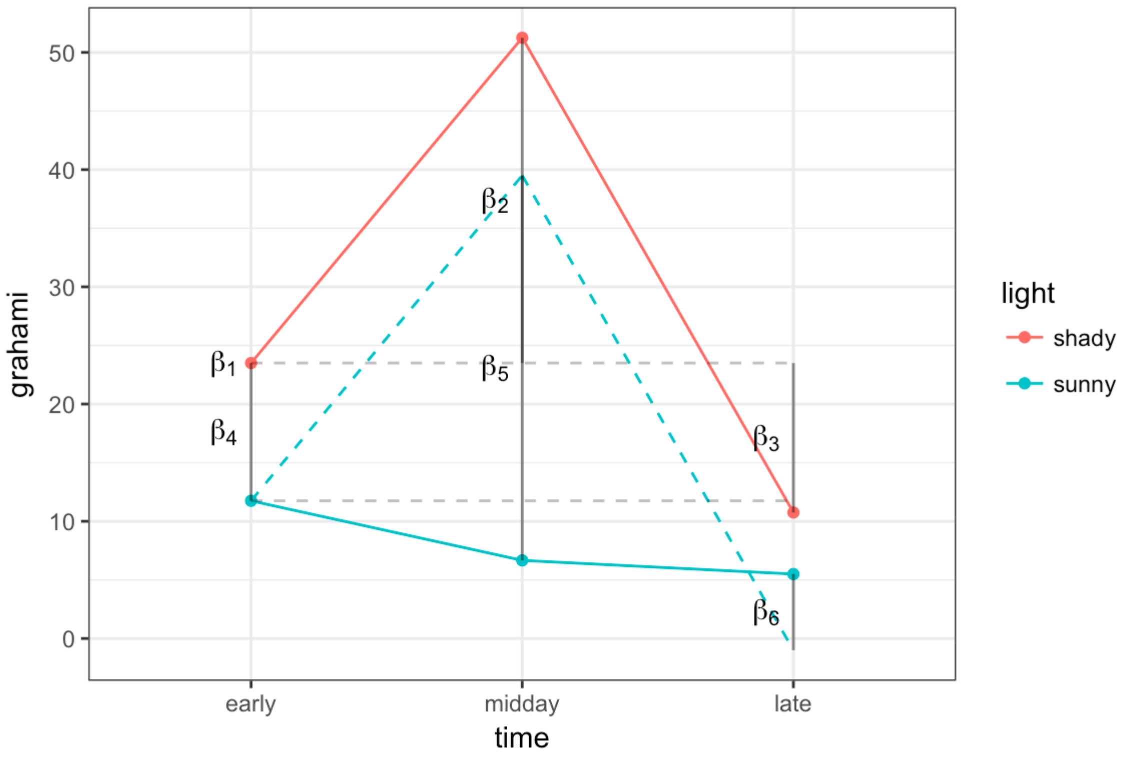
Interaction model

```
pr(lmTL2 <- lm(grahami~time*light,data=lizards))
```

```
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                23.50       5.38    4.37  0.00042
## timemidday                 27.75       7.60    3.65  0.00198
## timelate                  -12.75       7.60   -1.68  0.11180
## lightsunny                -11.75       7.60   -1.55  0.14061
## timemidday:lightsunny     -32.83      11.19   -2.93  0.00927
## timelate:lightsunny         6.50      10.75    0.60  0.55343
```

Parameters β_1 to β_4 have the same meanings as before. Now we also have β_5 and β_6, labelled "timemidday:lightsunny" and "timelate:lightsunny", which describe the difference between the expected mean value of these treatment combinations based on the additive model (which are $\beta_1 + \beta_2 + \beta_4$ and $\beta_1 + \beta_3 + \beta_4$ respectively) and their actual values.

Graphical version



Sum-to-zero contrasts

The fits are easy:

```
pr(lmTL1S <- update(lmTL1,contrasts=list(time=contr.sum,light=contr.sum)))
```

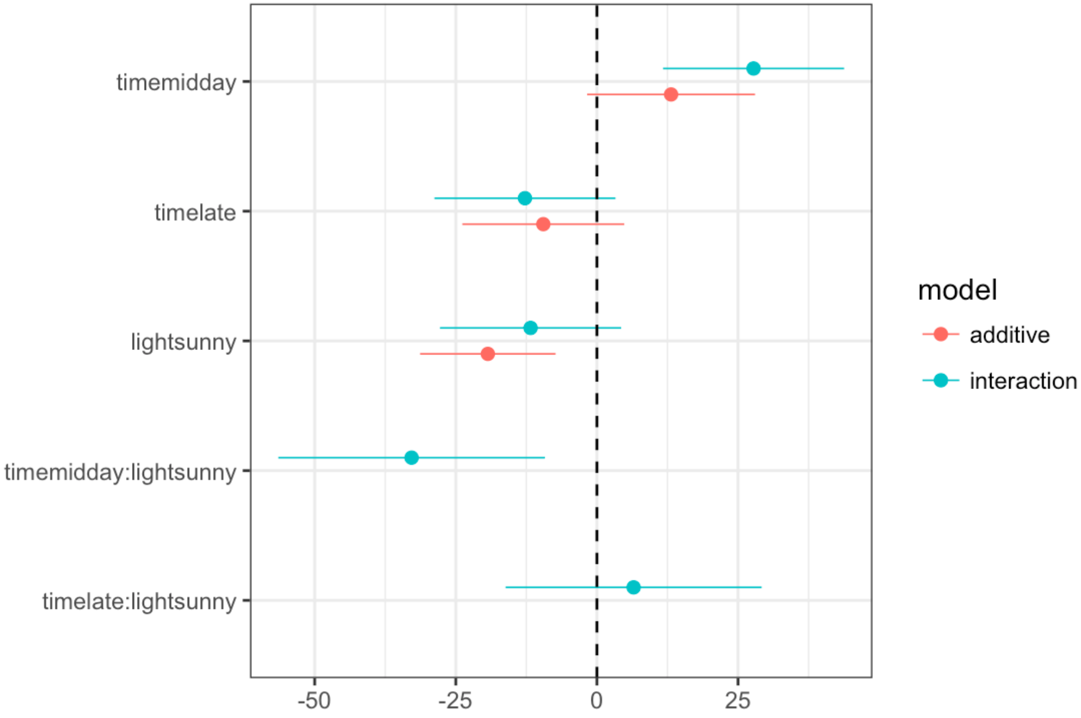
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    18.84       2.87    6.57  2.7e-06
## time1          -1.21       4.01   -0.30   0.7654
## time2          11.92       4.15    2.87   0.0097
## light1          9.66       2.87    3.37   0.0032
```

```
pr(lmTL2S <- update(lmTL2,contrasts=list(time=contr.sum,light=contr.sum)))
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    18.236      2.255    8.09  3.1e-07
## time1          -0.611      3.146   -0.19  0.84830
## time2          10.722      3.271    3.28  0.00444
## light1         10.264      2.255    4.55  0.00028
## time1:light1   -4.389      3.146   -1.39  0.18100
## time2:light1   12.028      3.271    3.68  0.00187
```

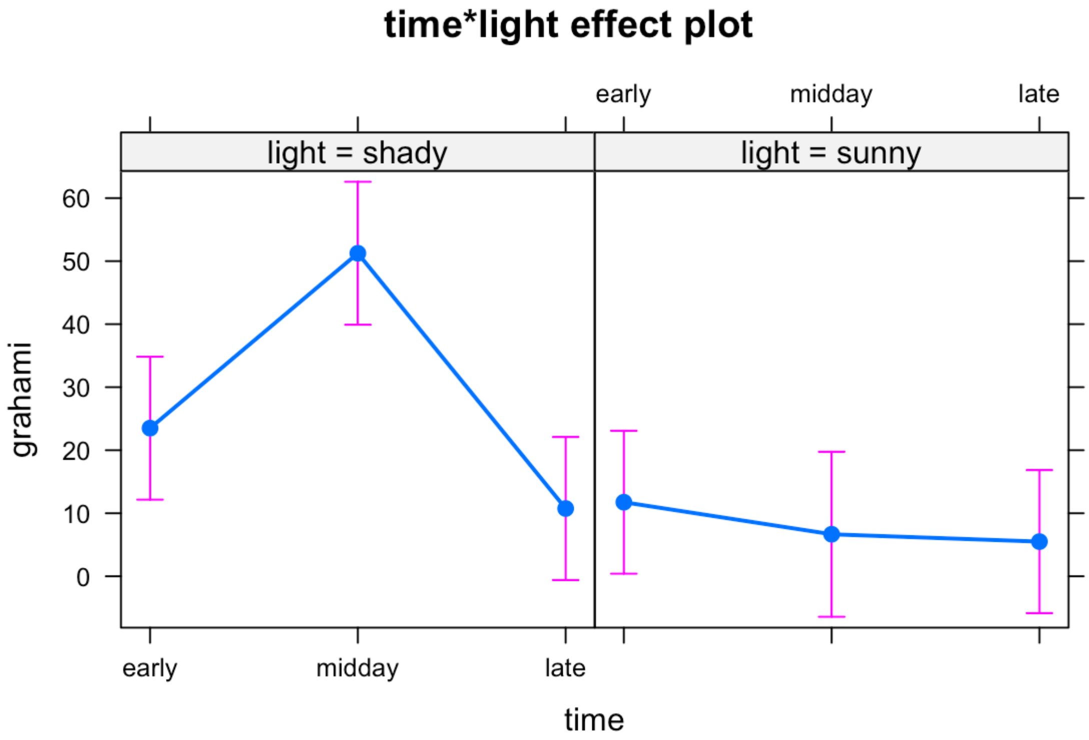
More graphics

```
dwplot(list(additive=lmTL1,interaction=lmTL2))+
    geom_vline(xintercept=0,lty=2)
```



Effects plot

```
plot(allEffects(lmTL2))
```



Session info:

```
sessionInfo()
```