Linear model parameters

```
## Warning: package 'knitr' was built under R version 3.4.3
```

Objectives

- Understand how to interpret R output and parameters in linear models
- Be able to describe the difference between an interactive and additive model
- Plot predictions from additive and interactive linear models

Model parameters: definitions

- Parameters of a linear model typically characterize *differences* in means
- These are differences per unit of change for *continuous* predictors,
- These are 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*
    - *\(\beta_1\) = expected value in baseline group (= first level of the factor variable, by default the first in alphabetical order);*
    - *\(\beta_i\) = 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),
    observers=c(1,3,2,1,5,2,1,2,1,1)
)
## 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 \(\beta_1\), 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: \(\beta_1\) is the same (mean of “field”), but \(\beta_2\) is ‘mean of “field”’ rather than (“(field)-(forest)”).

Interpretation using `predict`

- Use the `predict` function:

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

```
##     fit      lwr       upr
## 1 10.75 8.489484 13.010516
## 2  7.00 5.154296  8.845704
```

Interpretation using `effects` package

- Use the `effects` package:

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

```
##  model: colonies ~ place
##
##  place effect
## place
##  field forest
##  10.75   7.00
##
##  Lower 95 Percent Confidence Limits
## place
##    field   forest
## 8.489484 5.154296
##
##  Upper 95 Percent Confidence Limits
## place
##     field    forest
## 13.010516  8.845704
```

Interpretation using `lsmeans` package

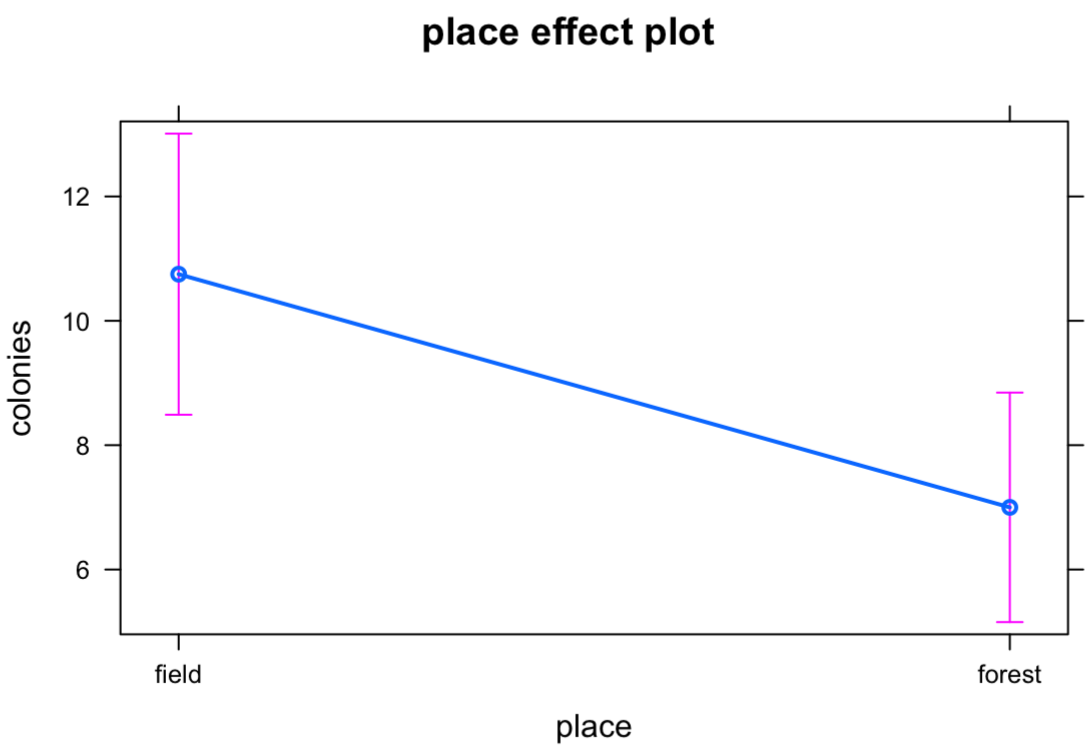
- Use the `lsmeans` package:

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

```
##  place  lsmean        SE df lower.CL  upper.CL
##  field   10.75 0.9802742  8 8.489484 13.010516
##  forest   7.00 0.8003905  8 5.154296  8.845704
##
## Confidence level used: 0.95
```

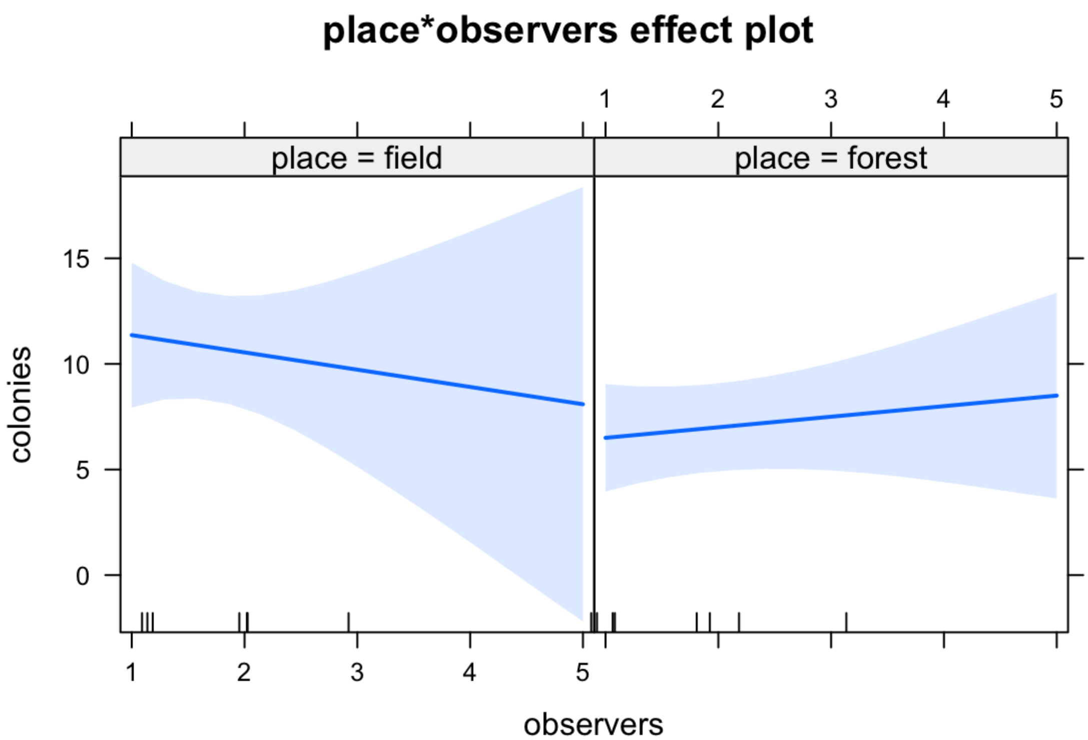
Graphical summaries from `effects` package

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



The `effects` package works on more complicated models

```
lm3 <- lm(colonies~place*observers,data=ants)
plot(allEffects(lm3))
```

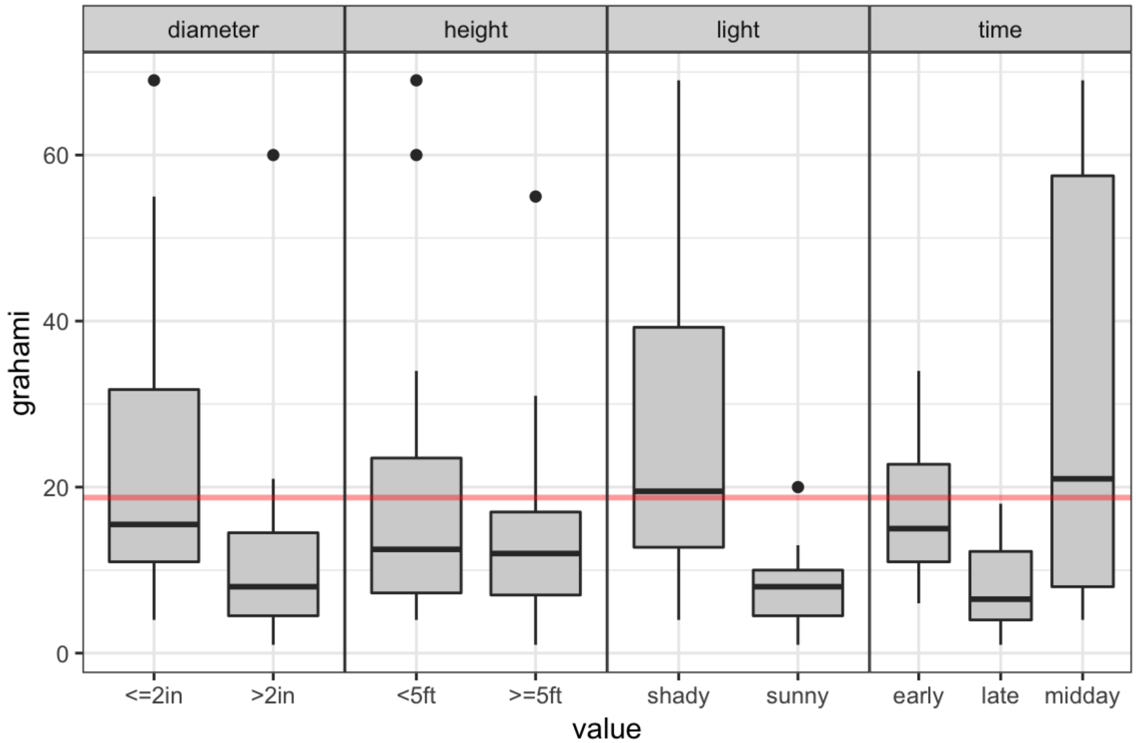


Switching to a dataset with more than two levels

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

What is the effect of time of day on lizard perching?

```
pr(lmX <- lm(grahami~time,data=lizards))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    17.63       5.97    2.95   0.0079
## timelate       -9.50       8.44   -1.13   0.2739
## timemidday     14.52       8.74    1.66   0.1123
```

If we leave the factors alphabetical then \(\beta_1\)="early", \(\beta_2\)="late"-"early", \(\beta_3\)="midday"-"early". It might be more sensible to change the levels in accordance with time progression.

Change the order of the levels:

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

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

This just swaps the definitions of \(\beta_2\) and \(\beta_3\).

```
pr(lmX <- lm(grahami~time,data=lizards))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    17.63       5.97    2.95   0.0079
## timemidday     14.52       8.74    1.66   0.1123
## timelate       -9.50       8.44   -1.13   0.2739
```

Multiple treatments and interactions

Additive model

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

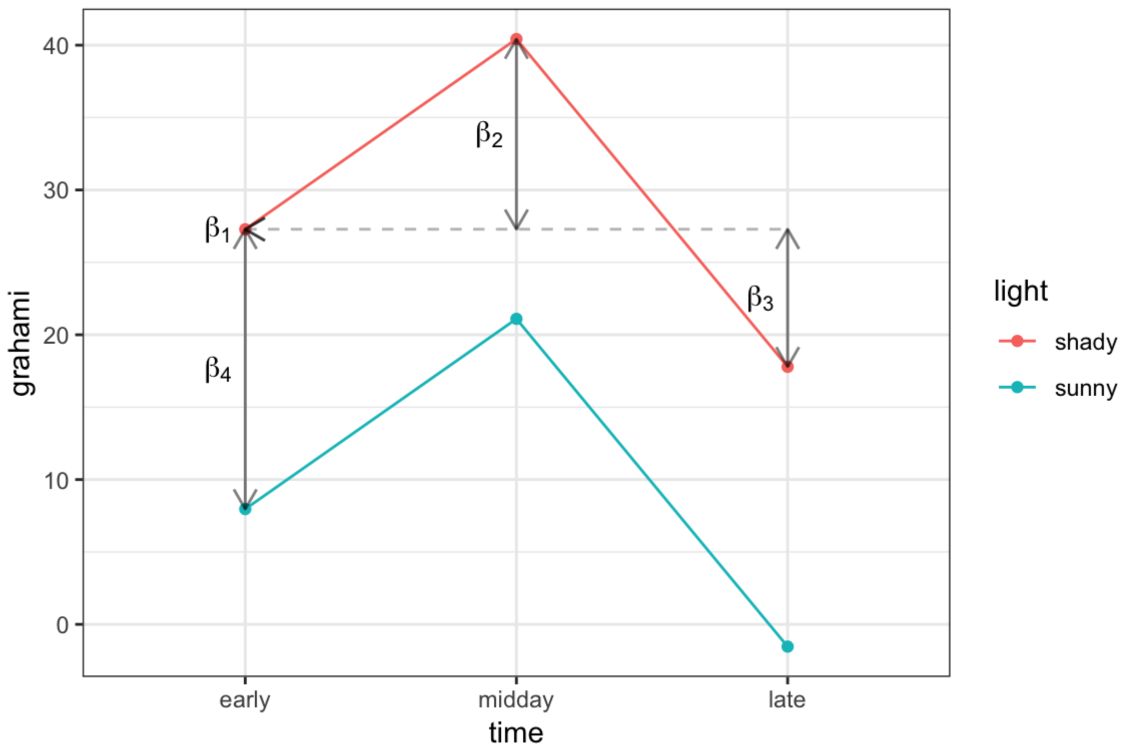
```
pr(lmTL1 <- lm(grahami~time+light,data=lizards))
```

```
##             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
```

\(\beta_1\) is the intercept ("early","sunny"); \(\beta_2\) and \(\beta_3\) are the differences from the baseline level ("early") of the *first* variable (`time`) in the *baseline* level of the other parameter(s) (`light`="shady"); \(\beta_4\) 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
```



What are the p-values?

```
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
```

The p-values tell us the difference from the baseline level, not from each other

Assessing differences among pairs of variable levels

```
library(lsmeans)
library(multcompView)
lsmeans(lmTL1, specs = "time", contr = "pairwise")
```

```
## $lsmeans
##  time    lsmean       SE df  lower.CL upper.CL
##  early  17.6250 4.846034 19  7.482134 27.76787
##  midday 30.7625 5.196793 19 19.885488 41.63951
##  late    8.1250 4.846034 19 -2.017866 18.26787
##
## Results are averaged over the levels of: light
## Confidence level used: 0.95
##
## $contrasts
##  contrast       estimate       SE df t.ratio p.value
##  early - midday -13.1375 7.105681 19  -1.849  0.1810
##  early - late     9.5000 6.853327 19   1.386  0.3677
##  midday - late   22.6375 7.105681 19   3.186  0.0129
##
## Results are averaged over the levels of: light
## P value adjustment: tukey method for comparing a family of 3 estimates
```

Getting an ABCDEF.. list

```
lsm1<-lsmeans(lmTL1,pairwise~time)
cld(lsm1,by = NULL, Letters = "ABCDEFGHIJ")
```

```
##  time    lsmean       SE df  lower.CL upper.CL .group
##  late    8.1250 4.846034 19 -2.017866 18.26787  A
##  early  17.6250 4.846034 19  7.482134 27.76787  AB
##  midday 30.7625 5.196793 19 19.885488 41.63951   B
##
## Results are averaged over the levels of: light
## Confidence level used: 0.95
## P value adjustment: tukey method for comparing a family of 3 estimates
## significance level used: alpha = 0.05
```

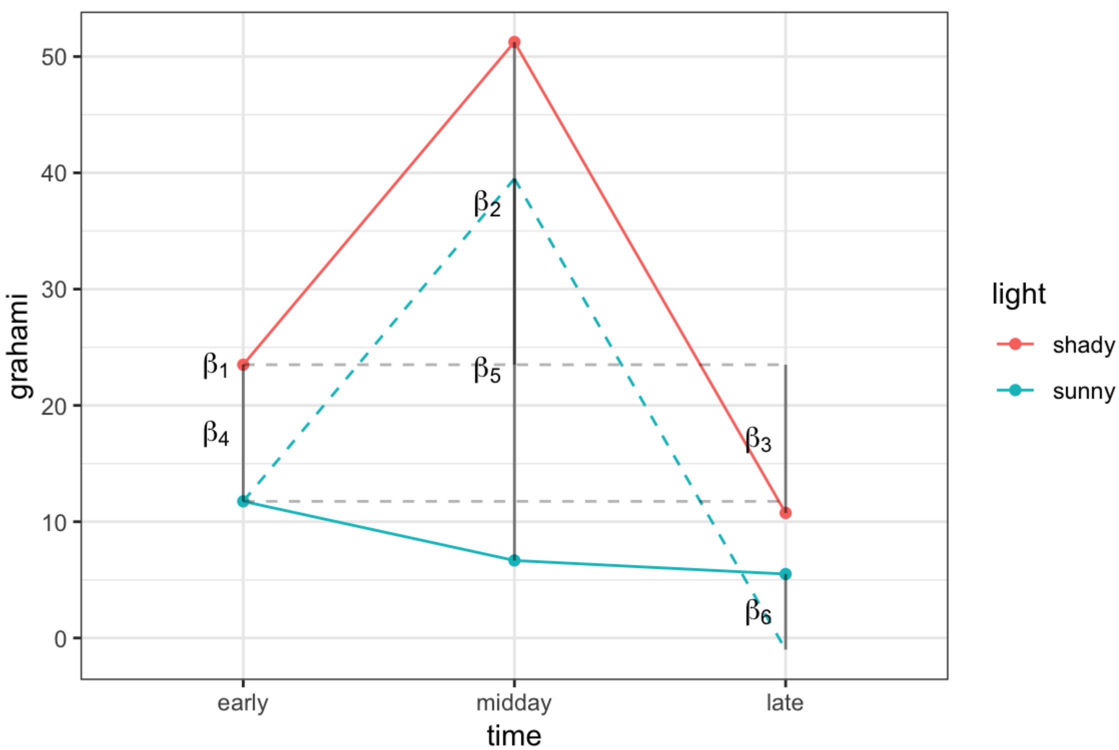
Interaction model

```
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
```

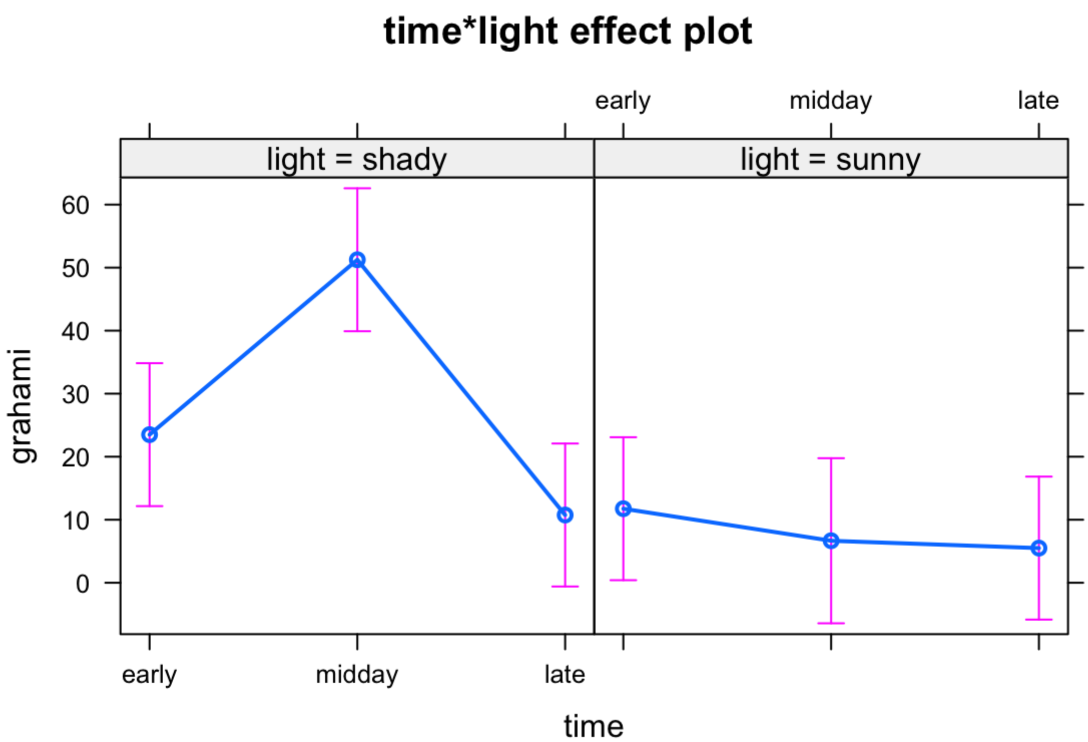
Parameters \(\beta_1\) to \(\beta_4\) have the same meanings as before. Now we also have \(\beta_5\) and \(\beta_6\), labelled "timemidday:lightsunny" and "timelate:lightsunny", which describe the difference between the expected mean value of these treatment combinations based on the additive model (which are \(\beta_1 + \beta_2 + \beta_4\) and \(\beta_1 + \beta_3 + \beta_4\) respectively) and their actual values.

Graphical version



Effects plot

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



Other refs

- http://sas-and-r.blogspot.com/2010/10/example-89-contrasts.html
- `gmodels::fit.contrast()` (show parameter estimates based on re-fitting models with new contrasts), `rms::contrast.rms()` (ditto, for `rms`-based fits)
- http://www.ats.ucla.edu/stat/r/library/contrast_coding.htm