

**Linear model
parameters**

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*
 - 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),
  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 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 "forest"' 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

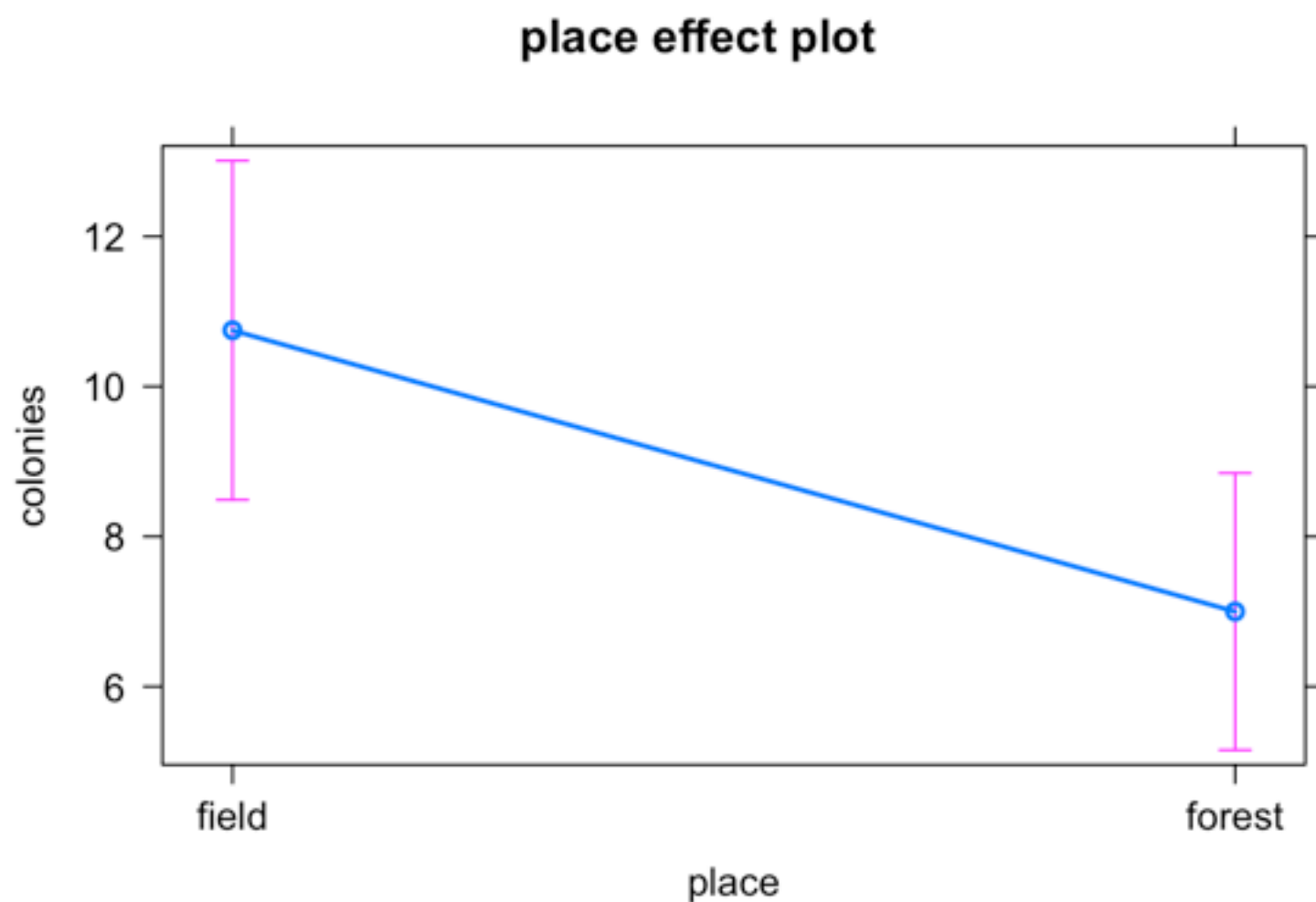
- Use the **lsmeans** package:

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

```
## place lsmean SE df lower.CL upper.CL  
## field 10.8 0.98 8 8.49 13.01  
## forest 7.0 0.80 8 5.15 8.85  
##  
## Confidence level used: 0.95
```

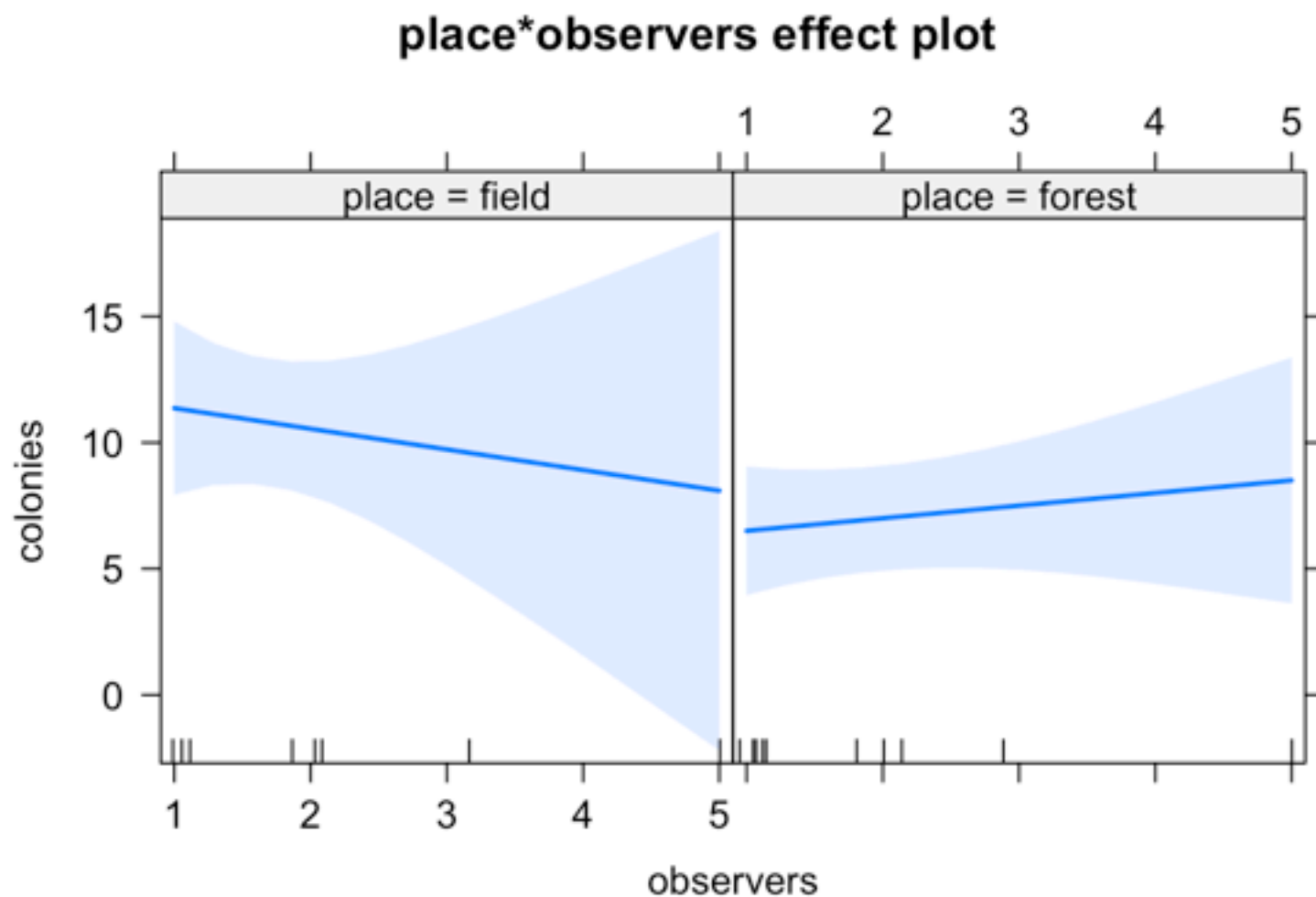
Graphical summaries from **effects** package

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



The **effects** package works on more complicated models

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

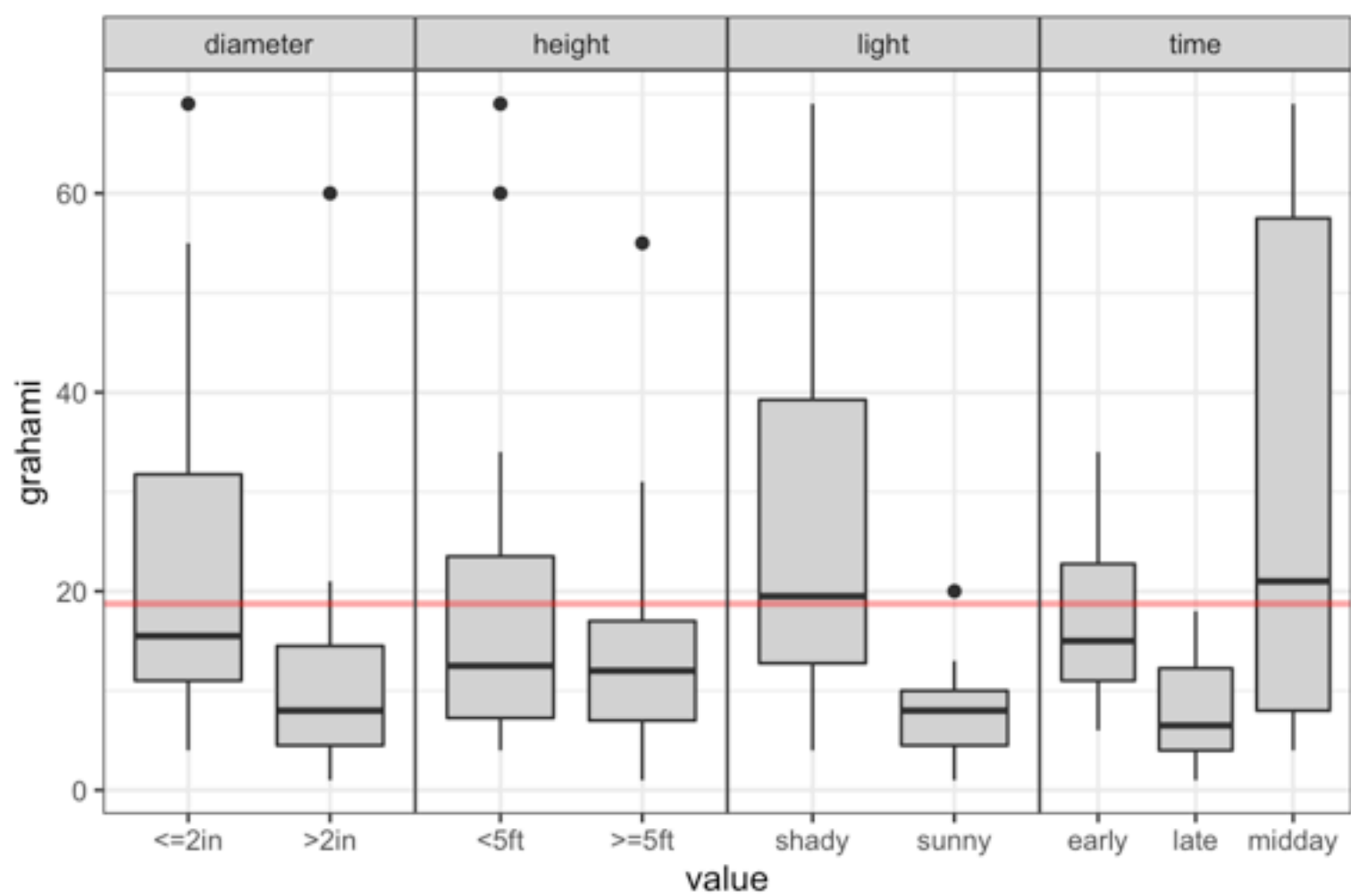


Switching to a dataset with more than two levels

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.



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 β_1 ="early", β_2 ="late"- "early", β_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")))
```

This just swaps the definitions of β_2 and β_3 .

```
pr(lmX <- lm(ghrami~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`.

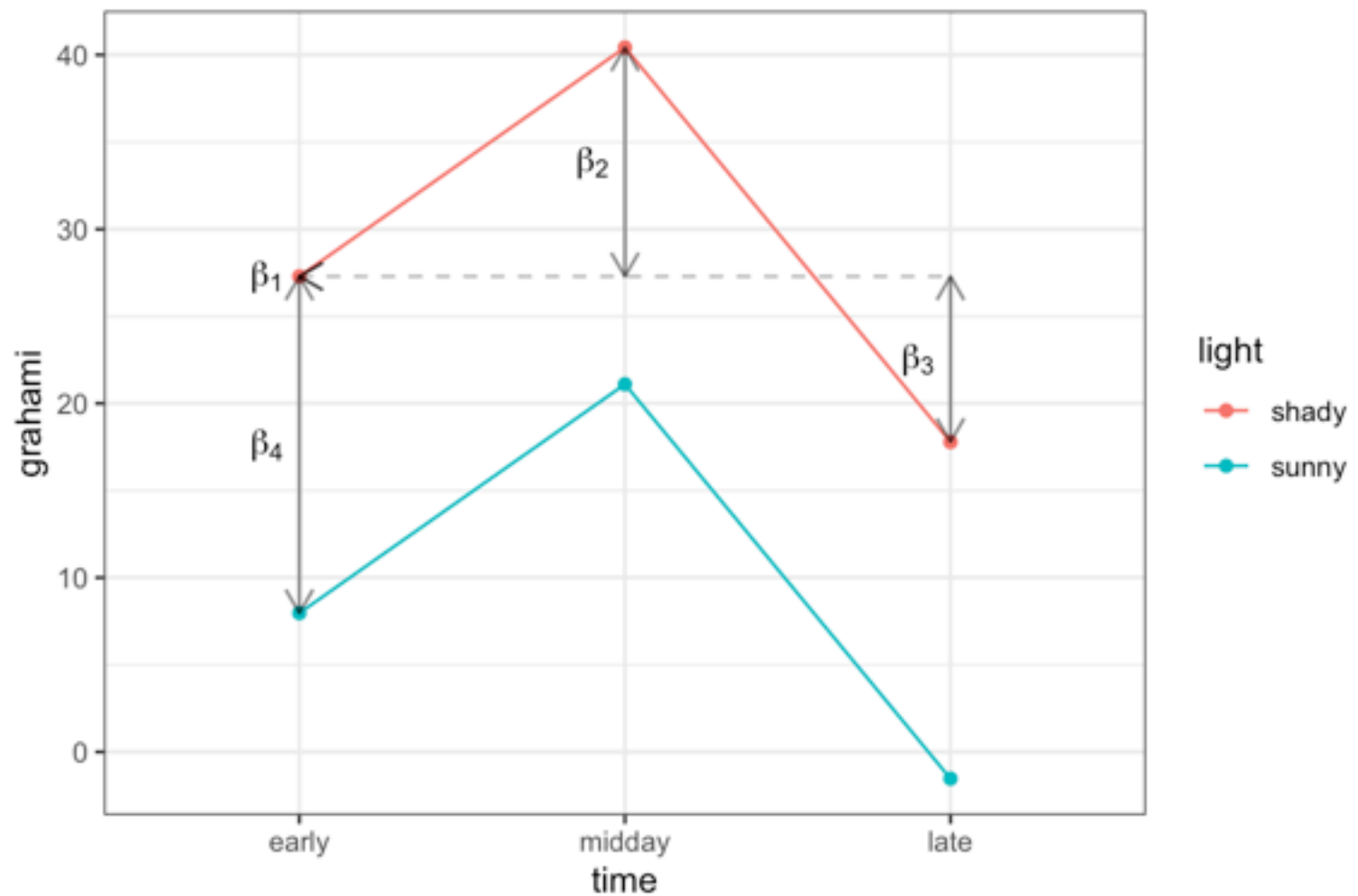
```
pr(lmTL1 <- lm(grahami~time+light,data=lizards))
```

##	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

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



What are the p-values?

```
pr(lmTL2 <- 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

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	17.62	4.85	19	7.48	27.8
## midday	30.76	5.20	19	19.89	41.6
## late	8.12	4.85	19	-2.02	18.3

```
##
```

```
## Results are averaged over the levels of: light
```

```
## Confidence level used: 0.95
```

```
##
```

```
## $contrasts
```

## contrast	estimate	SE	df	t.ratio	p.value
## early - midday	-13.1	7.11	19	-1.849	0.1810
## early - late	9.5	6.85	19	1.386	0.3677
## midday - late	22.6	7.11	19	3.186	0.0129

```
##
```

```
## Results are averaged over the levels of: light
```

```
## P value adjustment: tukey method for comparing a family o
```

Getting an ABCDEF.. list

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

```
## Warning: 'CLD' will be deprecated. Its use is discouraged
## See '? CLD' for an explanation. Use 'pwpp' or 'multcomp::
```

```
## Warning in CLD.emm_list(lsm1, by = NULL, Letters = "ABCDE
## with a list of 2 objects. Only the first one was used.
```

```
## Warning: 'CLD' will be deprecated. Its use is discouraged
## See '? CLD' for an explanation. Use 'pwpp' or 'multcomp::
```

```
##   time    lsmean    SE df lower.CL upper.CL .group
##   late      8.12 4.85 19    -2.02     18.3    A
##   early    17.62 4.85 19     7.48     27.8   AB
##   midday   30.76 5.20 19    19.89     41.6    B
##
```

```
## Results are averaged over the levels of: light
```

```
## Confidence level used: 0.95
```

```
## P value adjustment: tukey method for comparing a family o
```

```
## significance level used: alpha = 0.05
```

Interactions

- Interactions allow the value of one predictor to affect the relationship between another predictor and the response variable
- Interpreting *main effects* in the presence of interactions is tricky
- Your estimate of the effect of variable B is no longer constant
- You need to pick a fixed point, or average in some way
- Example:
$$Y = a + b_1X_1 + b_2X_2 + b_{12}X_1 * X_2$$
- The response to X_1 is
$$Y = (a + b_2X_2) + (b_1 + b_{12}X_2)X_1$$

the response to X_1 *depends on* the value of X_2 .

An example

- You think that the number of lizards on a perch on a sunny day might depend on the time day
- For example, on a very sunny day, there might be fewer lizards perching at noon; whereas on a cloudy day, the number of lizards might be highest at noon

Testing interactions

- Time is an important factor
- Use an *interaction*:

$$M = a + B_x X + B_t t + B_{xt} X t$$

- The interaction term B_{xt} represents the *difference in the response* between the two groups.
- It asks: **does the number of lizards perching depend on time of day and light?**
- Could also write:

$$M = a + B_1 \text{light} + B_2 \text{time} + B_3 \text{light} * \text{time}$$

- [Draw plot here] ## Treatment and interactions
- We previously discussed the wrong construction of linear models to compare drug treatments in mutant and non-mutant mice (erroneous interactions)
- You want to know whether a drug that

Interactions and parameters

- We can use CIs, and coefficient plots, and get a pretty good idea what's going on
- In more complicated cases, interaction terms may have many parameters
- These have all the interpretation problems of other multi-parameter variables
- Think about "differences in differences"

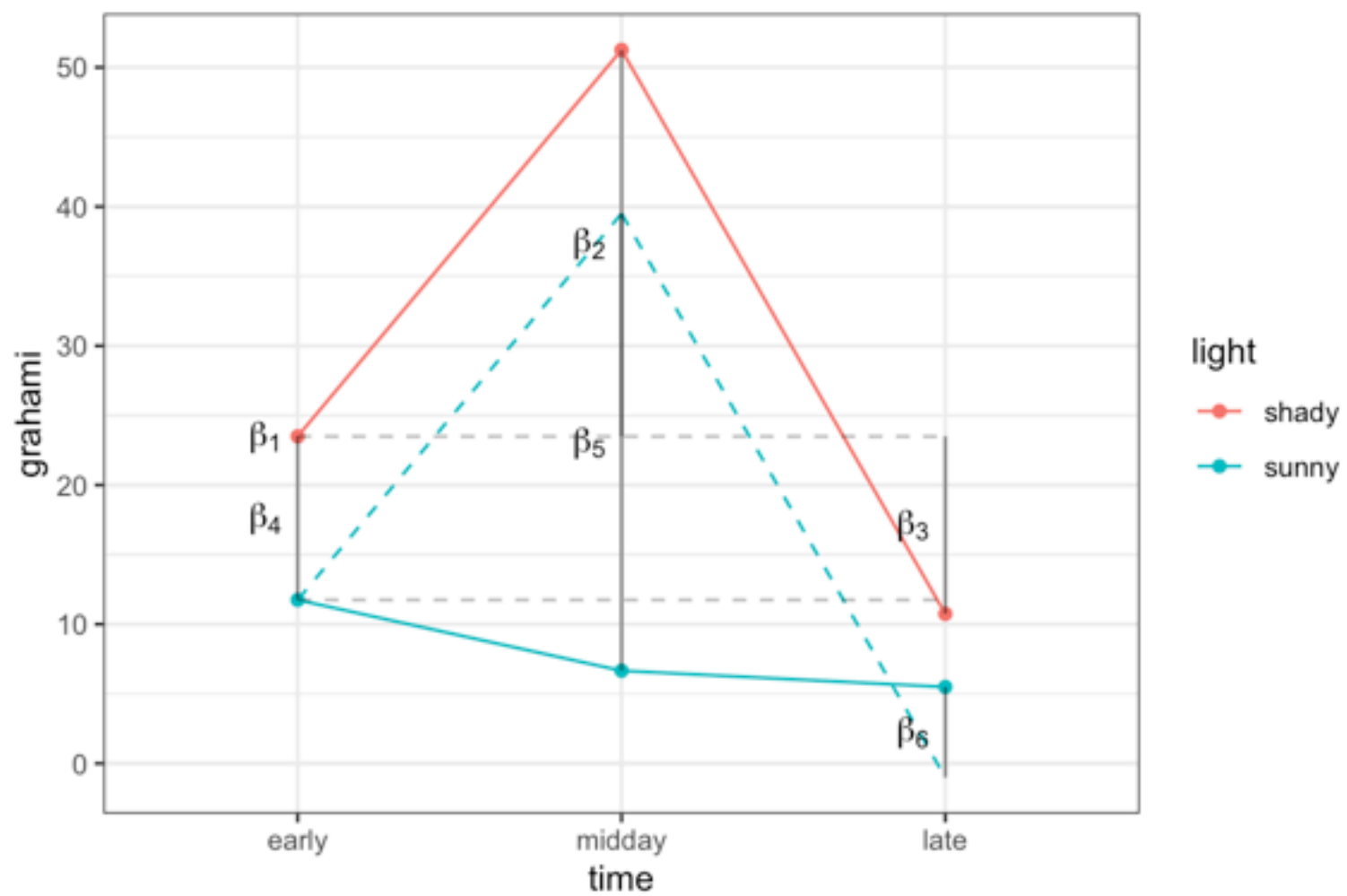
Interaction model

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

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	23.50	5.38	4.37	0.0004
## timemidday	27.75	7.60	3.65	0.0019
## timelate	-12.75	7.60	-1.68	0.1118
## lightsunny	-11.75	7.60	-1.55	0.1406
## timemidday:lightsunny	-32.83	11.19	-2.93	0.0092
## timelate:lightsunny	6.50	10.75	0.60	0.5534

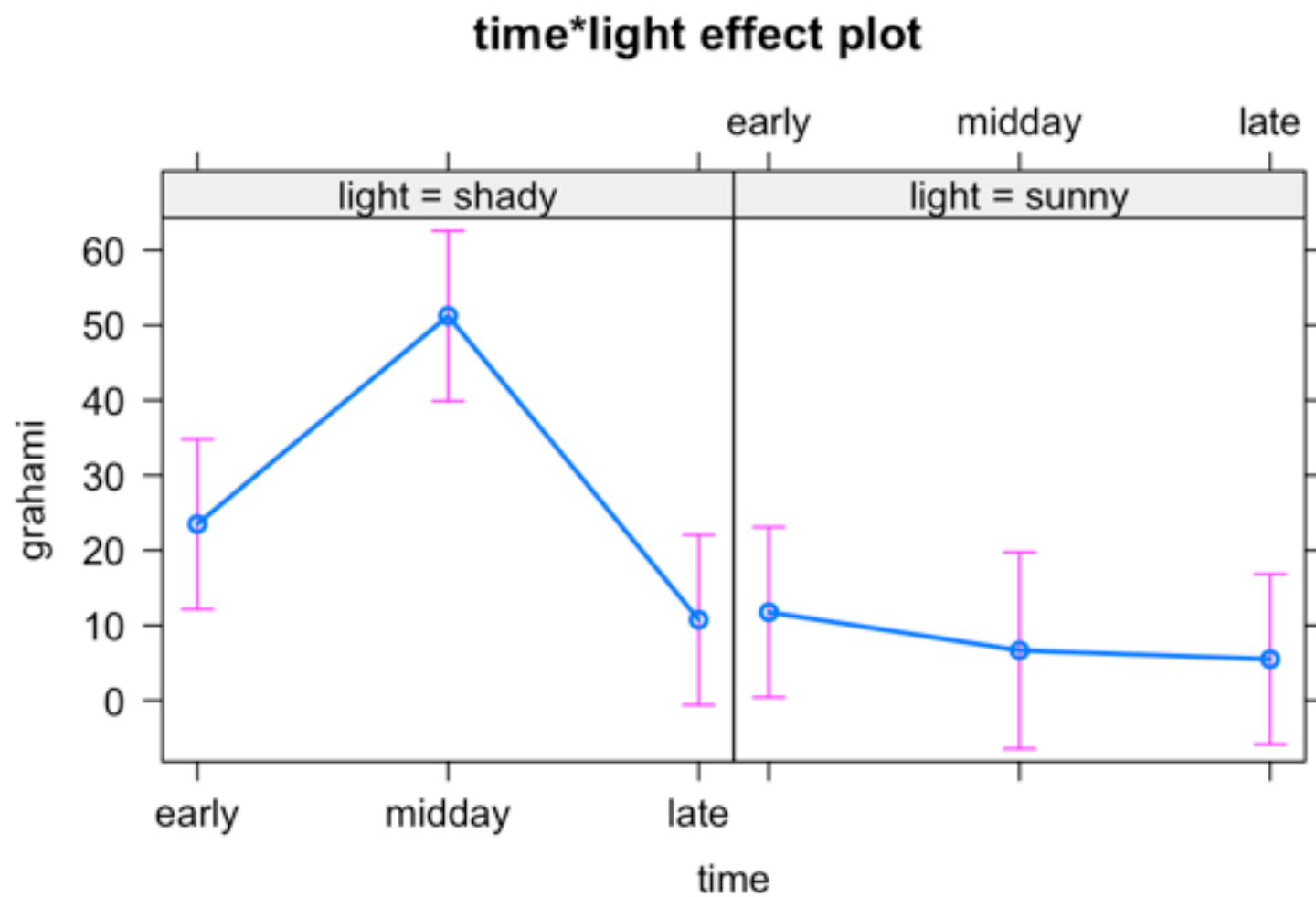
Parameters β_1 to β_4 have the same meanings as before. Now we also have β_5 and β_6 , labelled "timemidday:lightsunny" and "timelate:lightsunny", which describe the difference between the expected mean value of these treatment combinations based on the additive model (which are $\beta_1 + \beta_2 +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

Graphical version



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

Assignment

- *PART 1*

1. Make a univariate linear model for one of your hypotheses
2. Examine the assumptions of linearity (using tests or diagnostic plots)
3. Plot the relationship in ggplot using `stat_smooth`

- *PART 2*

1. Make a linear model (with more than one variable) for one of your hypotheses. Articulate which hypothesis you are testing.
2. Use an interactive model and an additive model. Explain what hypothesis each of these is testing, and what the R output is telling you about your data. (Hint: you can use `lsmeans`, `effects`, `relevel`, or `predict` to help you.) You should include this explanation in either your README or in your code.
3. Plot your model (e.g. using `predict`) and