More on Linear Regression Models

Professor Diane Lambert

June 2010

Supported by MOE-Microsoft Key Laboratory of Statistics and Information Technology and the Beijing International Center for Mathematical Research, Peking University.

With many thanks to Professor Bin Yu of University of California Berkeley, and Professor Yan Yao and Professor Ming Jiang of Peking University.

Schedule

Lectures 10:00 - 11:30 (with a break)

Labs 14:00 - 15:30

Tentative Plan

		10-11:30	14:00 - 15:30
June 15, 16	linear regression	√	√
June 17	logistic regression	√	
June 18	logistic regression		√
June 21	Google statistics		√
June 22	logistic regression	√	√
June 23, 24	multilevel models	√	√

Last Lecture

A linear regression model is defined by

$$E(Y \mid \mathbf{X}) = \beta_0 + \beta_1 X_1 + \dots \beta_K X_K$$

$$\mathrm{var}(Y \mid \mathbf{X}) = \sigma^2$$

We estimated coefficients, found residuals, made plots, looked at classical tests, interpreted R summaries for linear models, especially tests of significance for the estimated regression coefficients.

This Lecture

- More modern ways to evaluate model fit
  - simulation from the fitted model
  - bootstrapping

Understanding Regression Variability

(b_0, …, b_K) are weighted means of the Y_i's

$$\mathbf{b} = (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{Y}$$

By the CLT, they are approximately normal with

- mean (b_0, …, b_K)
- standard deviation (se(b_0), …, se(b_K))
- But the coefficients are usually not independent
  - b_j, b_k are independent when they are orthogonal
  - The CLT implies that **b** is approximately multivariate normal
    - R gives the correlation matrix for (b_0, …, b_K)

Computing the Covariance of **b** in R

```
z <- lm(sleep ~ log(body) + danger,
        data = sleep)
   summary(z) #Prints statistics.

zSummary <- summary(z) #Saves statistics.

covB <- zSummary$sigma^2 *
                 zSummary$cov.unscaled
```

covB is the cov matrix for $(b_0,\ldots,b_K)$

The Distribution of **b**

b is approximately normal with mean **b** and covariance matrix covB.

This is also the posterior distribution for **b** when the prior distribution of **b** is uniform.

Like any other distribution, this multivariate normal distribution describes which vectors of linear regression coefficients are likely, and which are not.

Each of these vectors of linear regression coefficients describes a different regression function, so the multivariate normal distribution of **b** describes the uncertainty around the regression mean.

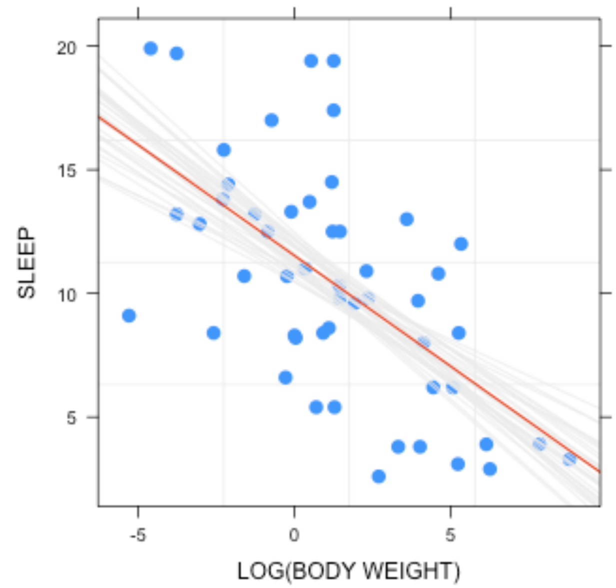
Simulating Uncertainty

Strategy

Simulate n multivariate normal vectors **b** with the `mvrnorm` function (in MASS).

If there is only one predictor, add the lines with coefficients equal to each of the simulated values to a plot of Y against the predictor (`panel.abline`)

The spread in the lines shows the uncertainty about the regression function.



30 Simulated Regression Lines for sleep against log(body).

The red line is the regression line computed from the data.

Simulating with More Than 1 Predictor

The simulation is the same.

- Regress sleep on log(body) and danger.
- Compute covB (same commands as in the 1 predictor case)
- Generate random Normal(b, covB) regression coefficients

Want to show the uncertainty in the regression mean, even though it is no longer a line.

- That is much easier to do with xyplot in lattice.

Back To The Example

Consider regressing sleep on log(body) and danger

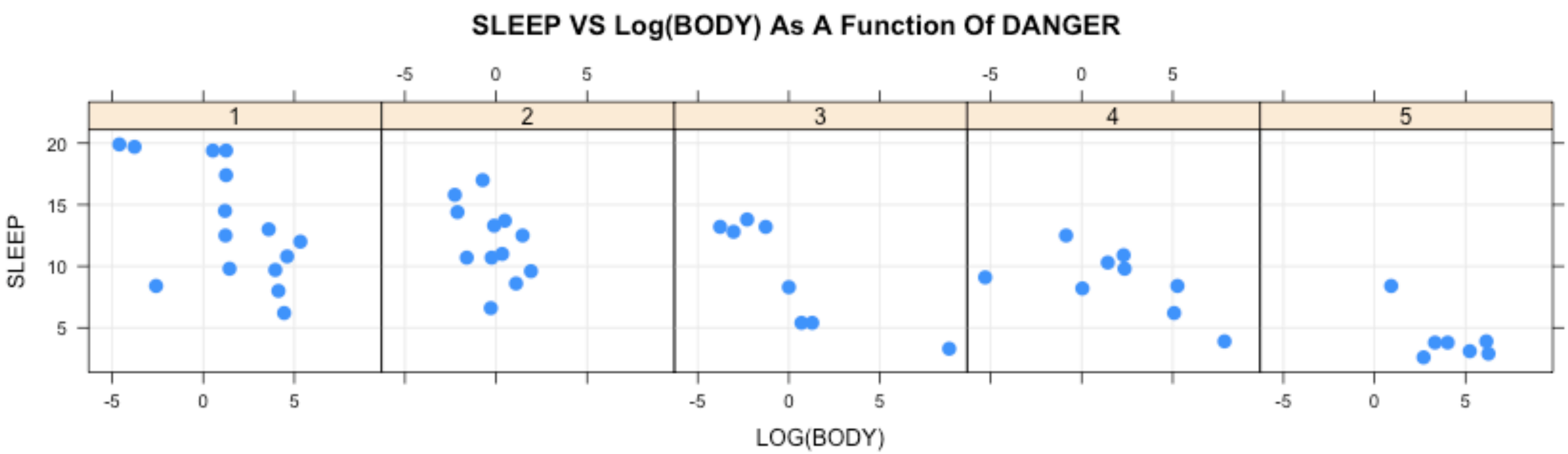
First, plot sleep vs log(body) for each value of danger

If there are too many values of both predictors, aggregate one of them

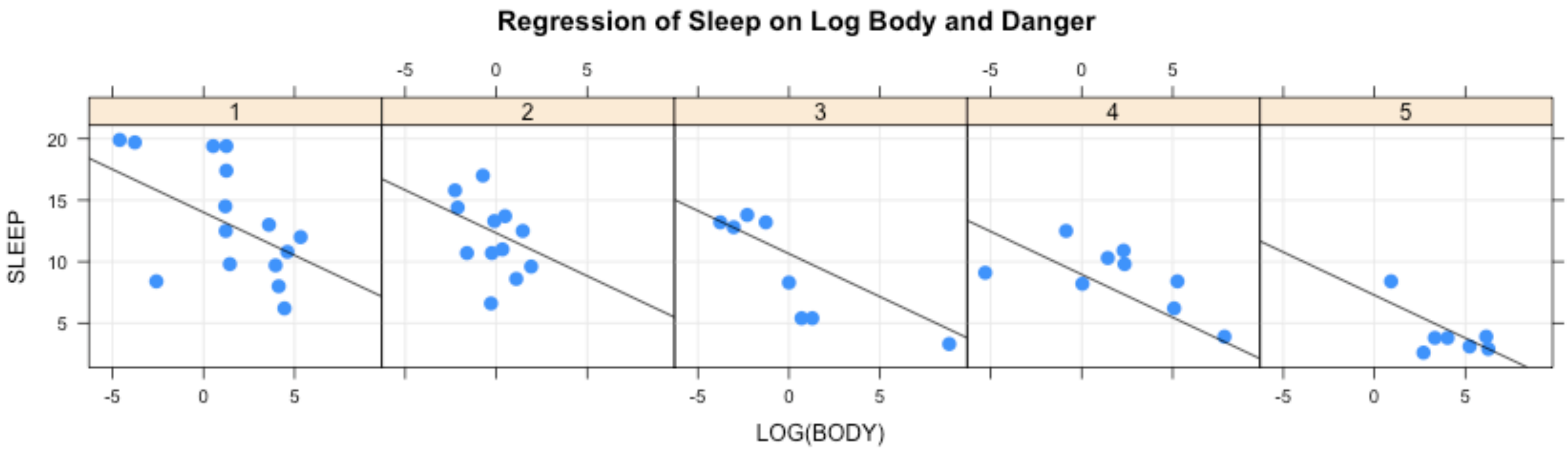
```
xyplot(sleep ~ log(body) | factor(danger))
```

Like conditional probability.

For each level of danger, plot sleep vs log(body)



Diagnosing the Regression Model

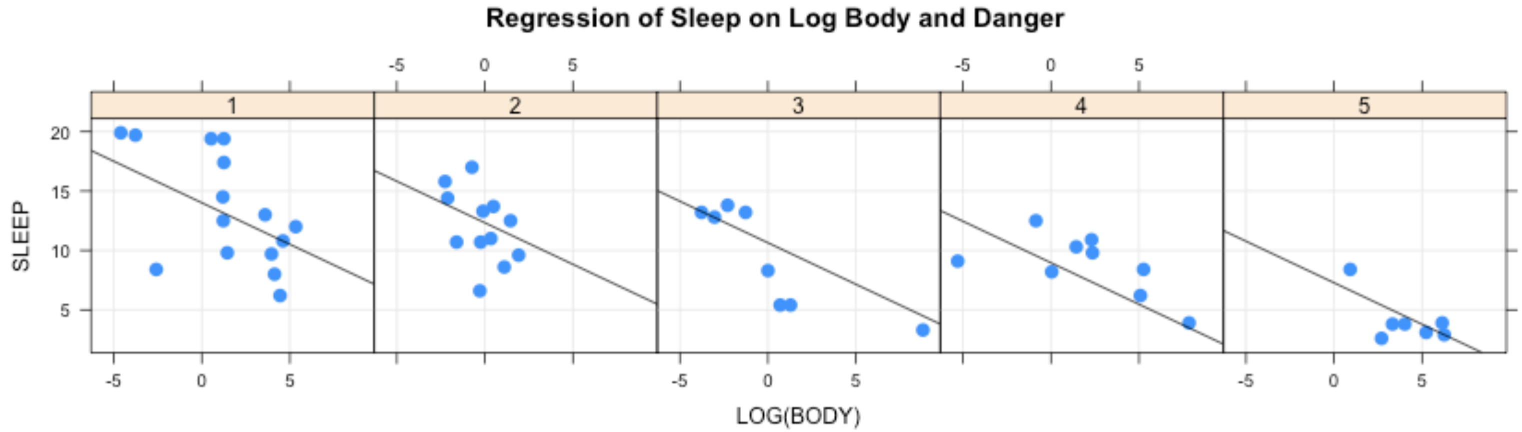


Our model

$$\text{mean(sleep)} = b_0 + b_1 \log(\text{body}) + b_2 \text{ danger}$$

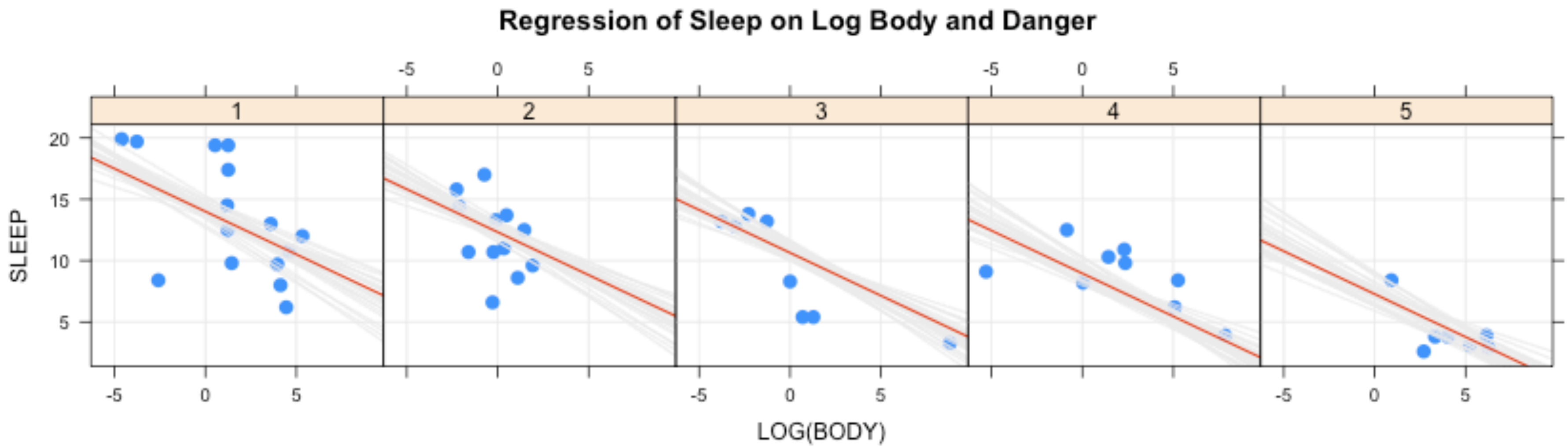
increasing danger by one adds b_2 to the intercept in a panel but the slope of sleep against log(body) in each panel is the same.

R code




```
xyplot(sleep ~ log(body) | factor(danger), data = sleep,
       layout = c(5, 1),
       panel = function(x, y, subscripts, ...) {
       panel.xyplot(x, y, ...)
         panel.abline(z2$coef[1] +
                 z2$coef[3] * sleep$danger[subscripts][1],
                      z2$coef[2])
         panel.lmline(x, y, col = 'magenta')
       })
```

Uncertainty In the Regression Model



Plot shows a random sample of 30 regression models

from the posterior distribution of **b**

from the sampling distribution of **b**

the uncertainty in the estimated mean E(Y | **X**)

Do you like this plot?

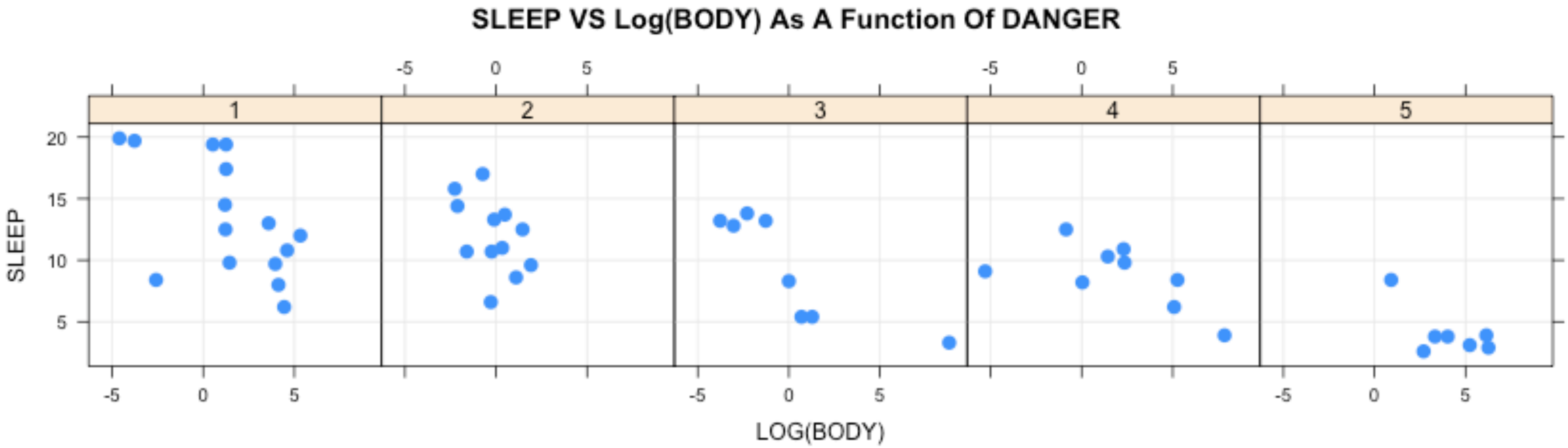
Simulated Regression Plots

We can do these for any kind of regression model, as long as we know the approximate distribution of the estimated model parameters -- no matter how fancy the model.

e.g., models with splines, glmnet.

Models With Interactions

Models With Interactions



Standard linear model:

log(body) has the same effect for every level of danger.

Interaction model:

allows different slopes in different panels

coefficient of log(body) can vary linearly with danger, or

it can be different for every level of danger.

Warning: there is a danger of overfitting!

First, Factors

A factor X_1 is a variable

that has *levels* (say L levels)

color, city, state, type of car

Allow us to add *nonparametric* terms to a model

Additive model

$$E(Y \mid \mathbf{X}) = b_0 + b_{1,j} + b_2 X_2 \qquad \sum_1^L b_{1,j} = 0$$

mean shifted differently for each level

Interaction model

$$E(Y \mid \mathbf{X}) = b_0 + b_{1,j} + \sum_j b_{2,j} X_2 \qquad \sum_1^L b_{1,j} = 0$$

mean shift and slope of X_2 changes with the level of X_1

WARNING

Additive model has L-1 parameters for the factor

$E(Y \mid \mathbf{X}) = b_0 + b_{1,j} + b_2 X_2, \quad b_{1,1} = 0$

Not using the textbook convention: $\sum_{1}^{L} b_{1,j} = 0$

only sensible for balanced models, when each

level is observed the same number of times

Interaction model has 2(L-1) more parameters for a factor compared to a numeric variable

$E(Y \mid \mathbf{X}) = b_0 + b_{1,j} + \sum_j b_{2,j} X_2$

Adding more parameters is not always good.

overfitting to one random sample

called generalization error in machine learning

Additive Model For Danger

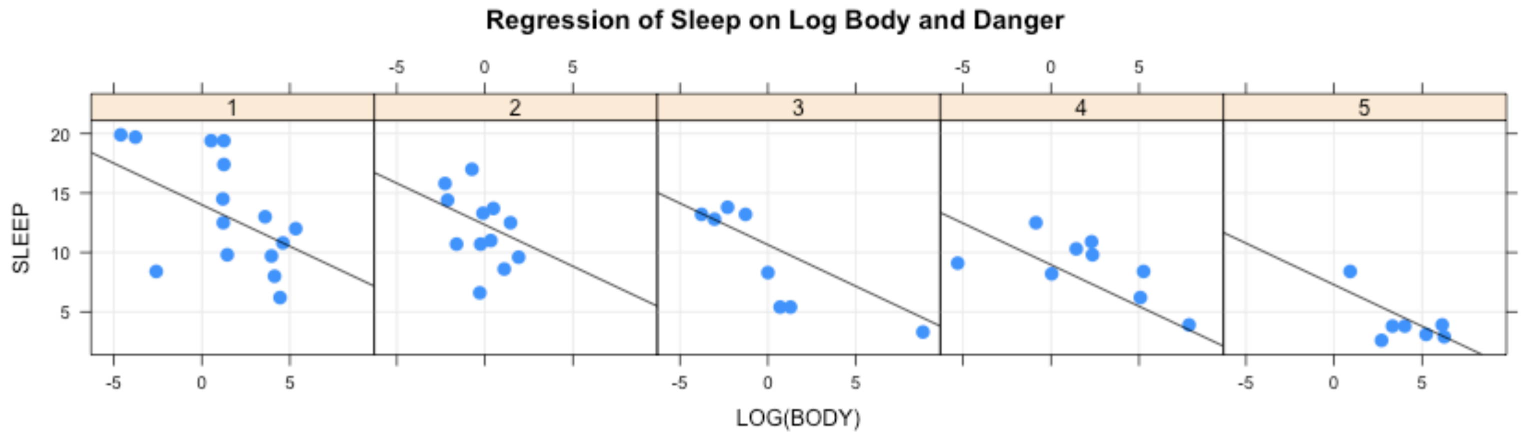
```
sleep$dangerFactor <- factor(sleep$danger)
z2Nonp <- lm(sleep ~ log(body) + dangerFactor,
             data = sleep)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	14.4847	0.7897	18.343	< 2e-16
log(body)	-0.7539	0.1461	-5.160	5.39e-06
dangerFactor2	-2.6232	1.1650	-2.252	0.029269
dangerFactor3	-5.0218	1.3053	-3.847	0.000374
dangerFactor4	-4.1531	1.2438	-3.339	0.001697
dangerFactor5	-7.3353	1.4007	-5.237	4.17e-06

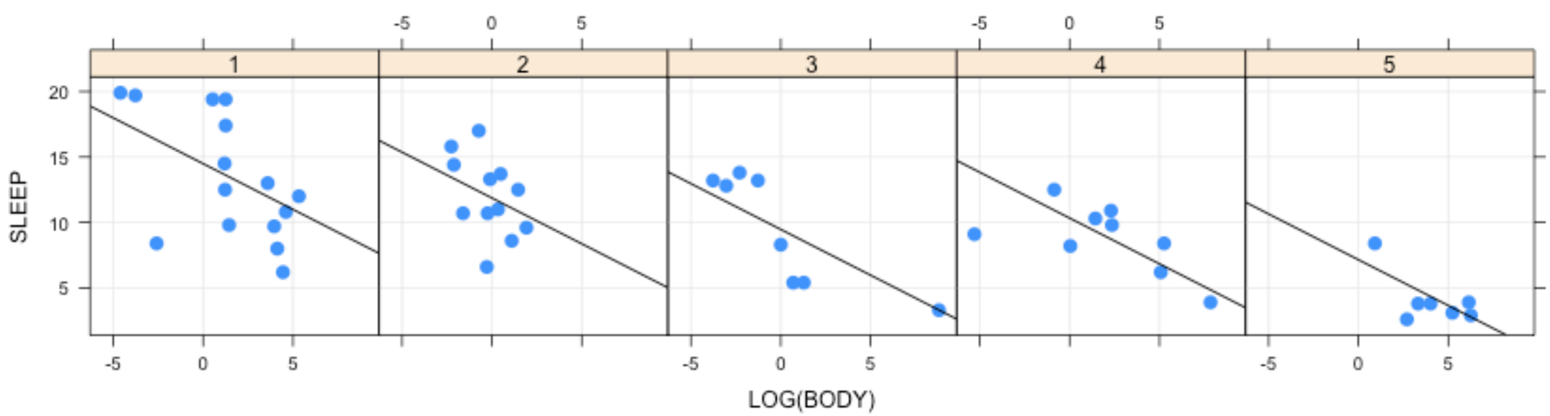
linear model with log body and danger had slope of log body of -.699

Additive Model For Danger

Linear Model in log(body) and danger



Linear Model in log(body) and level shift for danger



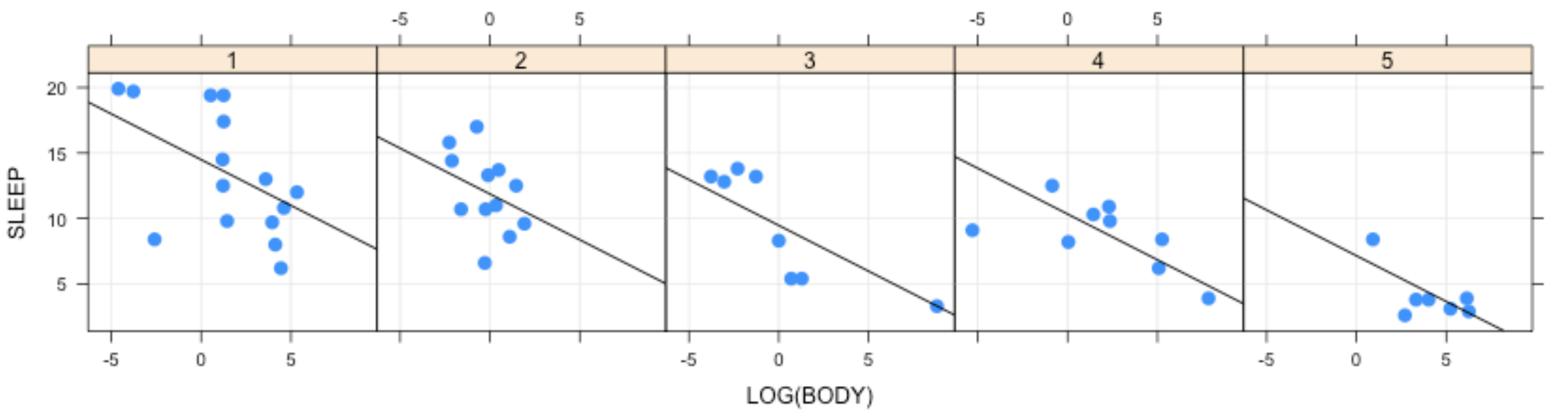
Interaction Model For Danger

```
z2NonpInt <- lm(sleep ~ log(body) * dangerFactor,
                data = sleep)
```

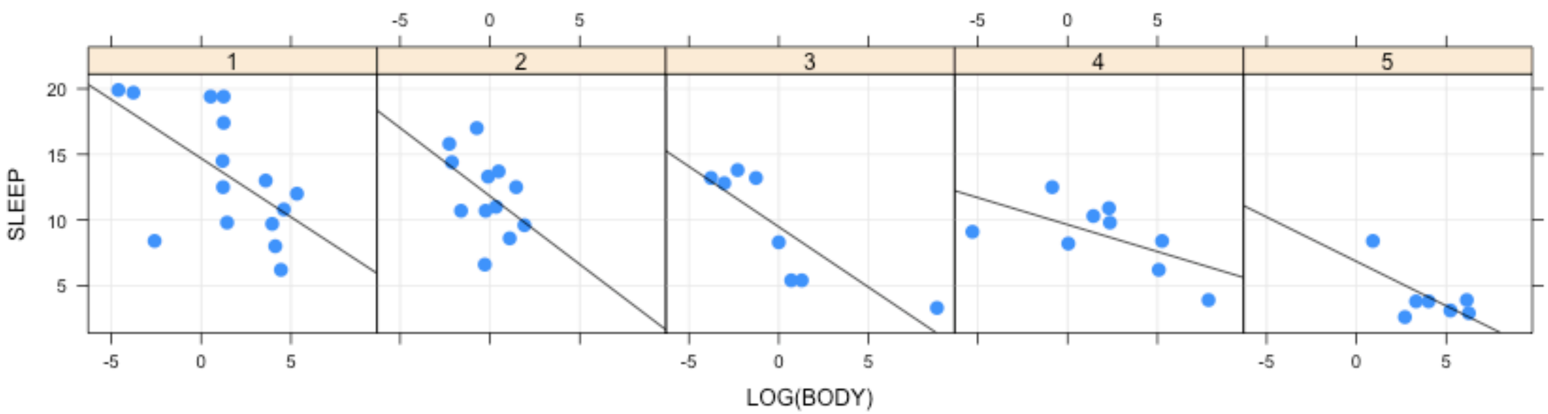
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	14.69	0.86	17.04	0.00
log(body)	-0.90	0.26	-3.45	0.00
dangerFactor2	-2.88	1.23	-2.35	0.02
dangerFactor3	-5.22	1.37	-3.82	0.00
dangerFactor4	-5.05	1.43	-3.53	0.00
dangerFactor5	-7.85	2.93	-2.68	0.01
log(body):dangerFactor2	-0.14	0.72	-0.20	0.84
log(body):dangerFactor3	-0.03	0.39	-0.07	0.95
log(body):dangerFactor4	0.48	0.38	1.28	0.21
log(body):dangerFactor5	0.22	0.68	0.32	0.75

Interaction Model For Danger

Additive Model in log(body) and level shift for danger



Interaction Model in log(body) and level of danger



Choosing A Model

Which Model Is Best

There are many ways to choose a model

Always look at the data!

You'll at least know how to scale the X_i's

Choosing a model may not scale the predictors.

When there is not too much data, the R function `leaps` will choose the best subset.

Must penalize models with more coefficients,

e.g. they choose the model with minimum

$$C_p = \sum e_i^2 / ((n-1)s^2) + 2K - N \text{ or}$$

$$BIC = \log(\sum e_i^2 / (n-K)) + (K/n) \log(n) \text{ or } \ldots$$

where s is the usual sample standard deviation.