Lab 3: Discussion of Assignments 1 and 2

Professor Diane Lambert

Teaching Assistants:
Daniel Cheng, Tianxi Li, Jenny Qu,
Elise Zhang, Zoey Zhao

PKU Summer School

June 2010

Re-naming variables

sleepExposure is too long, so re-name that column exposure.

```
names(sleep)[names(sleep) == 'sleepExposure'] <-
      'exposure'
```

Assignment 1

2. Take a look at the NAs.

matrix and list methods apply to a dataframe

```
nNAs <- sapply(is.na(sleep), sum)
nNAs <- apply(is.na(sleep), 2, sum)
```

Find the rows with NA in any column except dreamSleep or slowWaveSleep

```
keepNames <- setdiff(names(sleep),
                     c('dreamSleep', 'slowWaveSleep'))
naRows <- apply(is.na(sleep[, keepNames]), 1, any)
```

There are 11 such rows. With more time, we would compare the species with NAs to those without.

Brain, Life, Gestation

3. Look at some quantiles

```
newX <- c('brain', 'life', 'gestation', 'predation',
          'sleepExposure')
sapply(sleep[, newX], quantile, na.rm = TRUE)
```

```
      brain  life gesta predation exposure
0%      0.1   2.0  12.0         1        1
25%     4.2   6.6  35.8         2        1
50%    17.2  15.1  79.0         3        2
75%   166.0  27.8 207.5         4        4
100% 5712.0 100.0 645.0         5        5
```

log brain weight, lifetime and gestation

Choosing One Predictor: R^2

$$R^2 = 1 - \frac{(n-K)^{-1}\sum\left(Y_i - \beta_0 - \beta_1 X_1 - \ldots - \beta_K X_K\right)^2}{(n-1)^{-1}\sum\left(Y_i - \bar{Y}\right)^2}$$

The R^2s can be computed all at once

```
allX <- c(newX[1:3], 'body', newX[4:5], 'danger')
sleepR2 <- rep(NA, 7)
names(sleepR2) <- allX
for (i in allX[1:4]) {
  z <- lm(sleep$sleep ~ log(sleep[[i]]))
  sleepR2[i] <- summary(z)$r.squared
}
for (i in allX[5:7]) {
  z <- lm(sleep$sleep ~ sleep[[i]])
  sleepR2[i] <- summary(z)$r.squared
}
```

Choosing One Predictor

One quick measure:

$$R^2 = 1 - \frac{(n-K)^{-1}\sum\left(Y_i - \beta_0 - \beta_1 X_1 - \ldots - \beta_K X_K\right)^2}{(n-1)^{-1}\sum\left(Y_i - \bar{Y}\right)^2}$$

R^2 is only an informal way to compare non-nested models

We want to compute them all at once in a loop

Computing R^2 for Many Models

```
allX <- c('brain', 'life', 'gestation','body',
          'predation','exposure', 'danger')

sleepR2 <- rep(NA, 7)
names(sleepR2) <- allX

for (i in allX[1:4]) {
  z <- lm(sleep$sleep ~ log(sleep[[i]]))
  sleepR2[i] <- summary(z)$r.squared
}
for (i in allX[5:7]) {
  z <- lm(sleep$sleep ~ sleep[[i]])
  sleepR2[i] <- summary(z)$r.squared
}
```

R^2

brain	.32
lifetime	.18
gestation	.44
body	.28
predation	.16
exposure	.41
danger	.35

R^2 suggests that the one-predictor models based on lifetime, predation and maybe body are not as good as the others so remove those.

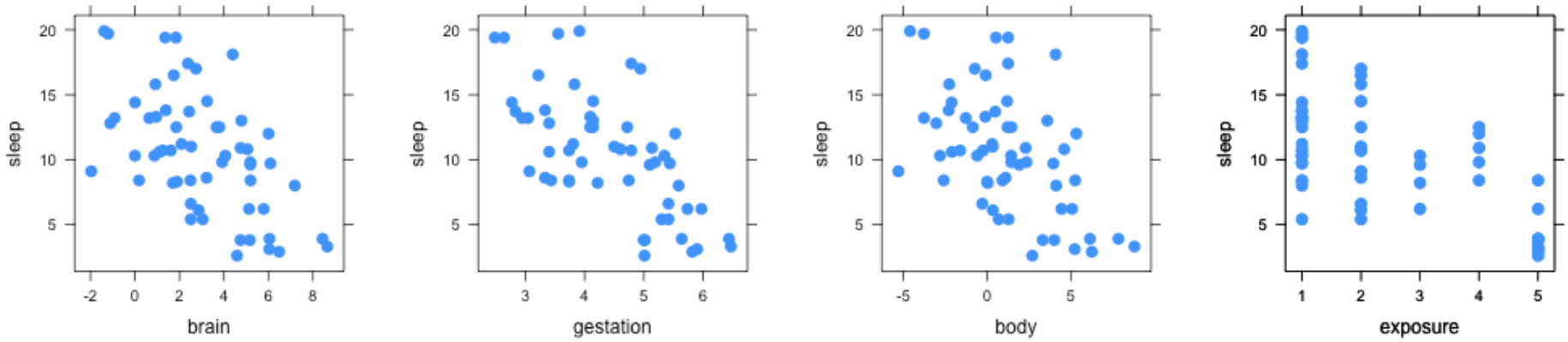
Plotting The One Predictor Models

```
allX <- c('brain', 'life', 'gestation','body',
'danger')
sleepLog <- sleep
for (i in c('brain', 'gestation', 'body')) {
  sleepLog[[i]] <- log(sleep[[i]])
}
newX <- c('brain', 'gestation', 'body',
          'exposure', 'danger')
plotList <- list()
for (i in newX) {
  plotList[[i]] <- xyplot(sleepLog$sleep ~
                          sleepLog[[i]],
                          xlab = i, ylab = 'sleep')
}
```

Plotting The One Predictor Models

```
plotList <- list()
for (i in newX) {
  plotList[[i]] <- xyplot(sleepLog$sleep ~
                          sleepLog[[i]],
                          xlab = i, ylab = 'sleep')
}
for (i in 1:4) {
  print(plotList[[i]],
  position = c(.25 * (i-1), 0, .25*i, 1),
  more = TRUE)
}
print(plotList[[i]], position = c(.75, 0, 1, 1))
```

Sleep vs Some Predictors



log(gestation) appears to have a stronger linear trend than log(body) or the other terms.

Except 4 observations have NA for gestation.

Adding a Predictor

The two best one predictor terms do not necessarily make the best two predictor model.

The two best one predictors may be highly correlated

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  21.9634     2.3703   9.266 1.96e-12 ***
gestation    -1.6673     0.5696  -2.927  0.00514 **
body         -0.3035     0.1871  -1.622  0.11111
danger       -1.4356     0.2947  -4.871 1.16e-05 ***
---
  (8 observations deleted due to missingness)
Multiple R-squared: 0.6387,      Adjusted R-squared: 0.617
```

Fitted Values When Y is NA

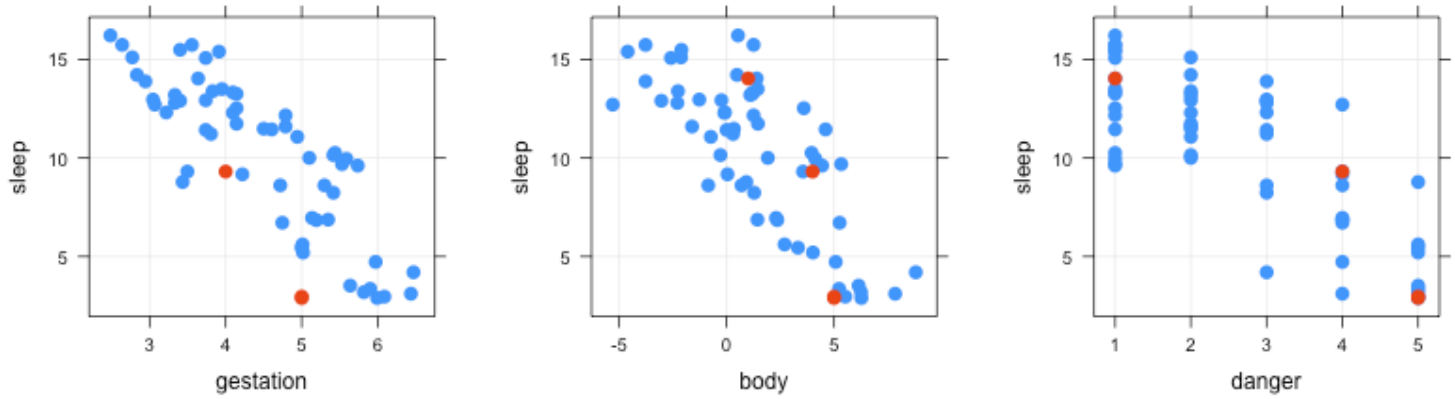
```
indx <- which(is.na(sleepLog$sleep))
predNA <- predict(z, newdata = sleepLog[indx, ])
predNA
       21         31         41         62
 2.892709   9.312259   2.961282 14.038465

predAll <- predict(z, newdata = sleepLog)
# fitted values will be NA when gestation is NA.
```

Plotting Fitted Values vs Predictors

```
plotList <- list()
for (i in c('gestation', 'body', 'danger')) {
  plotList[[i]] <- xyplot(predAll ~sleepLog[[i]],
     ylab = 'sleep',xlab = i, pch =  16,
      panel = function(x, y, ...) {
          panel.grid(h=-1, v=-1)
          panel.xyplot(x, y)
  panel.points(sleepLog[[i]][indx], predNA,
                col = 'red', pch = 16, cex = 1.25)
              })
}
print(plotList[[1]],position=c(0,0,1/3,1),more=T)
print(plotList[[2]],position=c(1/3,0,2/3,1),more=T
)
print(plotList[[3]], position = c(2/3, 0, 1, 1))
```

Fitted Values For All Outcomes

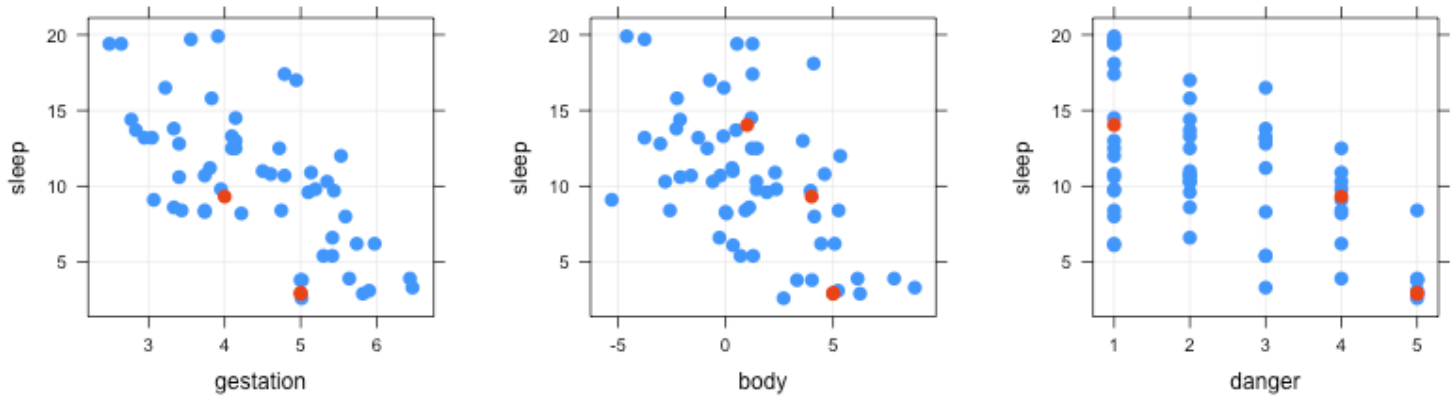


The red points have an outcome (sleep) of NA.

We would prefer them to be in the middle of the prediction clouds, not on the edge.

If they're in the middle, they're more likely to be an unbiased subset of all the data.

Observed Sleep vs Predictors



The red points have an outcome (sleep) of NA.

Use the same code as above, substituting sleep$sleep for predAll. Points with NA in one co-ordinate are not plotted (with no warning).

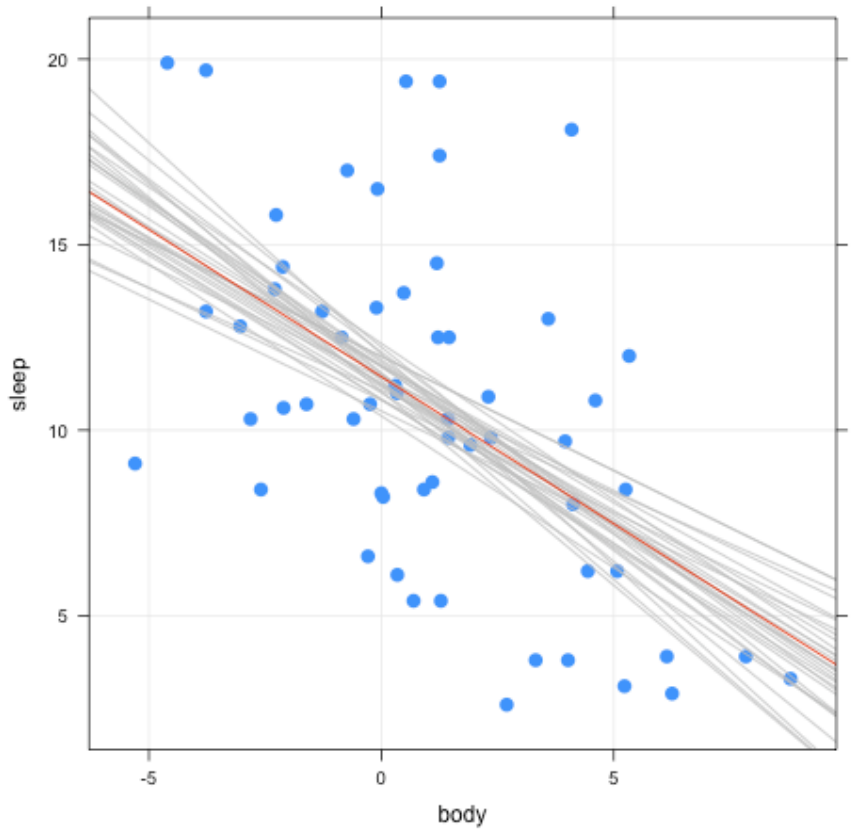
Simulating Model Uncertainty

I'll choose log(gestation)

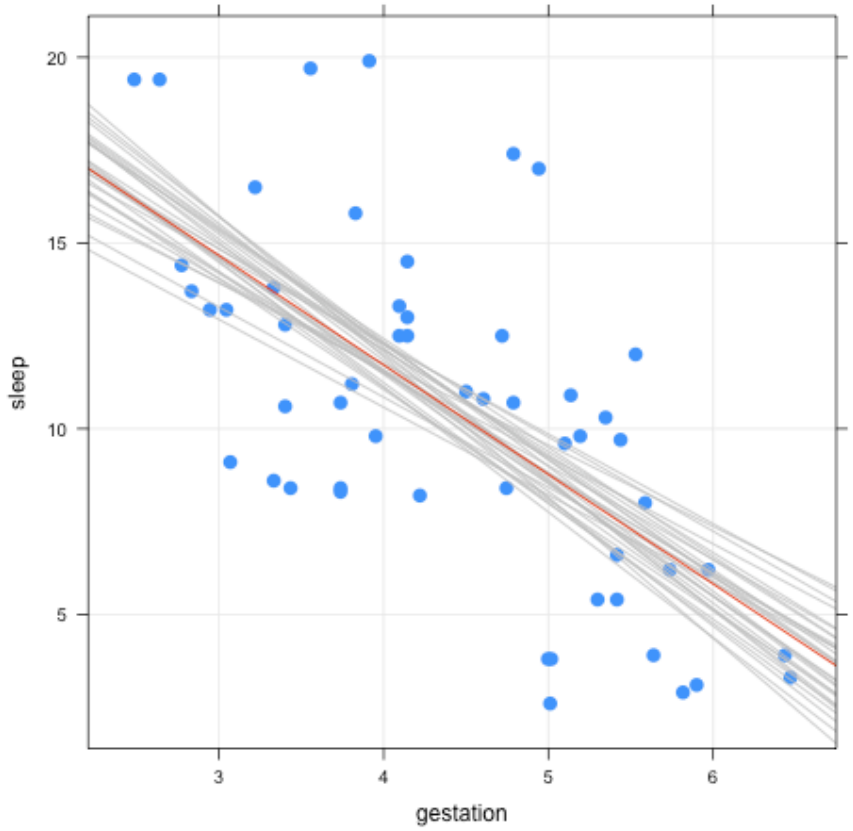
```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  23.5063     2.0903  11.246 1.55e-15
gestation    -2.9480     0.4608  -6.398 4.51e-08
xyplot(sleep ~ gestation, data = sleepLog,
       panel = function(x,y, ...) {
         panel.grid(h = -1, v = -1)
         panel.xyplot(x, y, ...)
         for (i in 1:30) {
  panel.abline(simEstimates[i,], col=gray(.9))}
         panel.abline(z$coefficients,
          col = 'red', lwd = 1.25)})
```

Model Uncertainty

Body



Gestation



Band around gestation is smaller, so prefer it over body.

Comparing Uncertainty

For a fairer comparison, show the uncertainty conditional on danger for both models.

There is no perfect way to compare non-nested models.

Plot the same regression line and simulated lines for the sleep ~ log(gestation) model in each panel (to each subset of data with the same value of danger.)

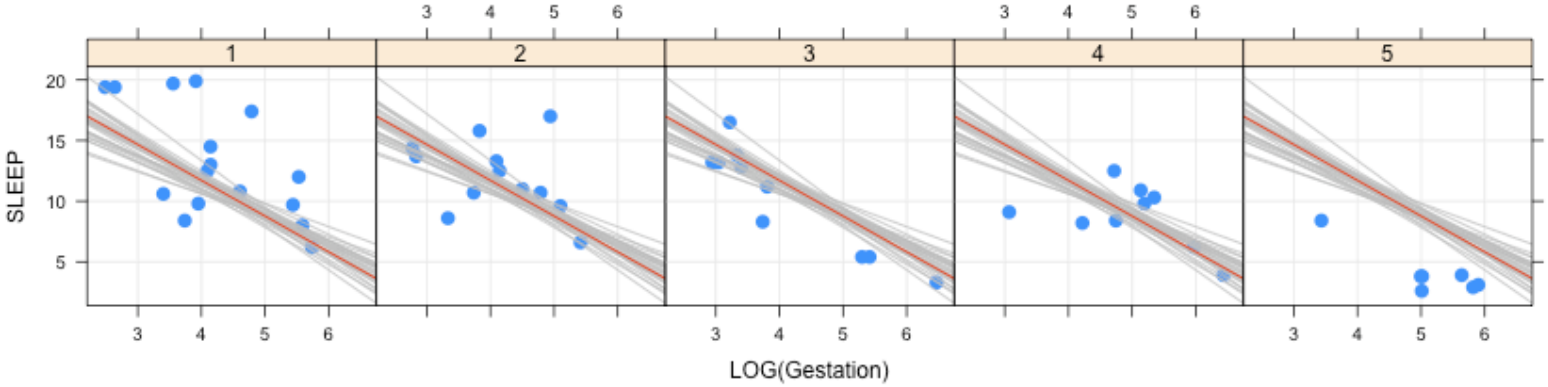
The R Code

```
sleepLog$dangerFact <- factor(sleepLog$danger)
xyplot(sleep ~ gestation | dangerFact,
       data = sleepLog,
       layout = c(5, 1),
       panel = function(x, y,  ...) {
         panel.grid(h = -1, v = -1)
         panel.xyplot(x, y, ...)
                  for (i in 1:30) {
           panel.abline(simEstimates[i, ], col =
gray(.7))}
panel.abline(z$coef, col='red', lwd = 1.25)
     }
)
```

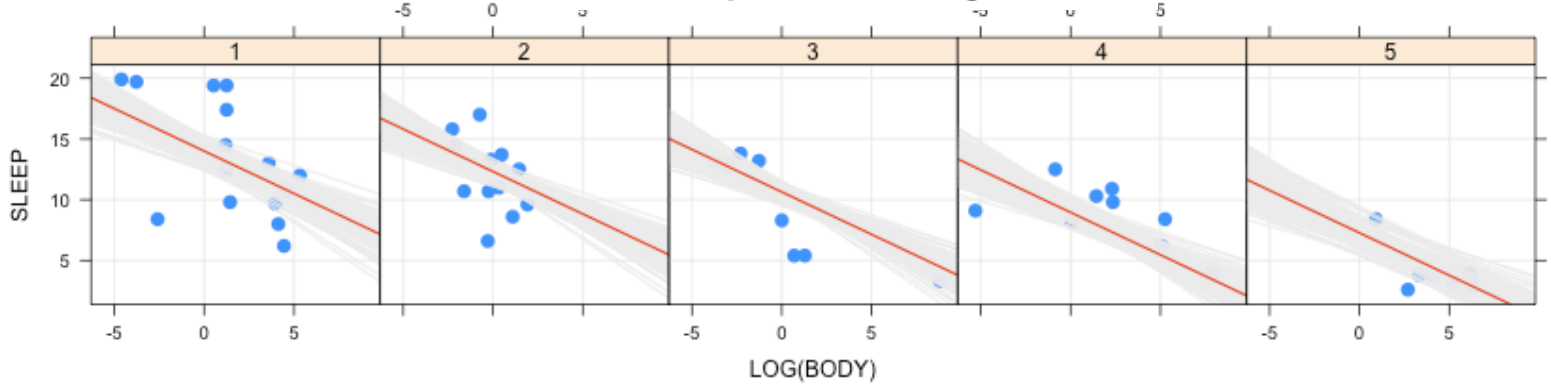
Comparing Uncertainty

Linear regression line and simulated lines.

Gestation



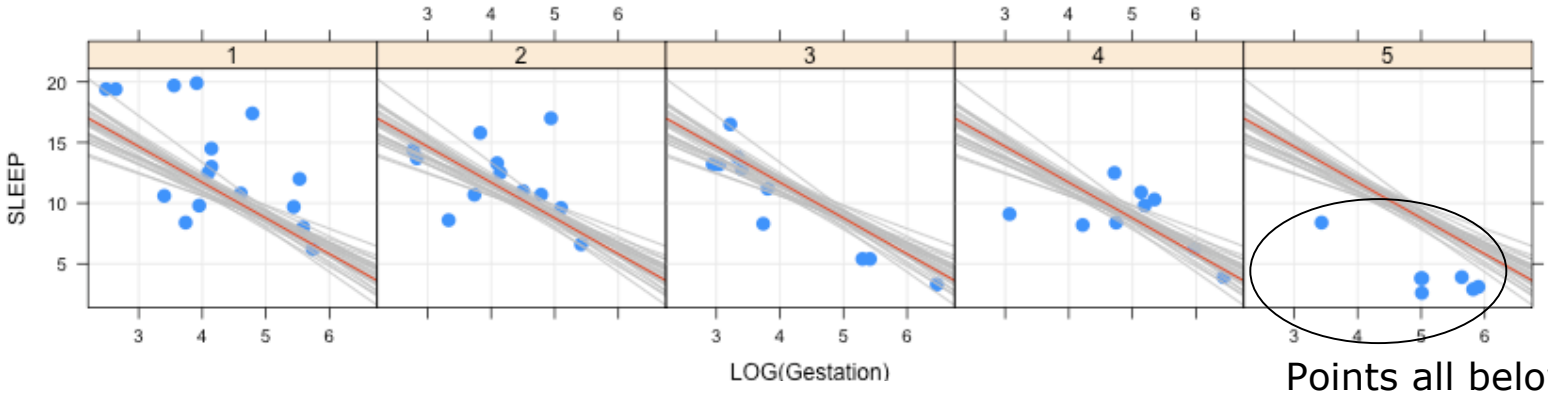
Body & Danger



Comparison

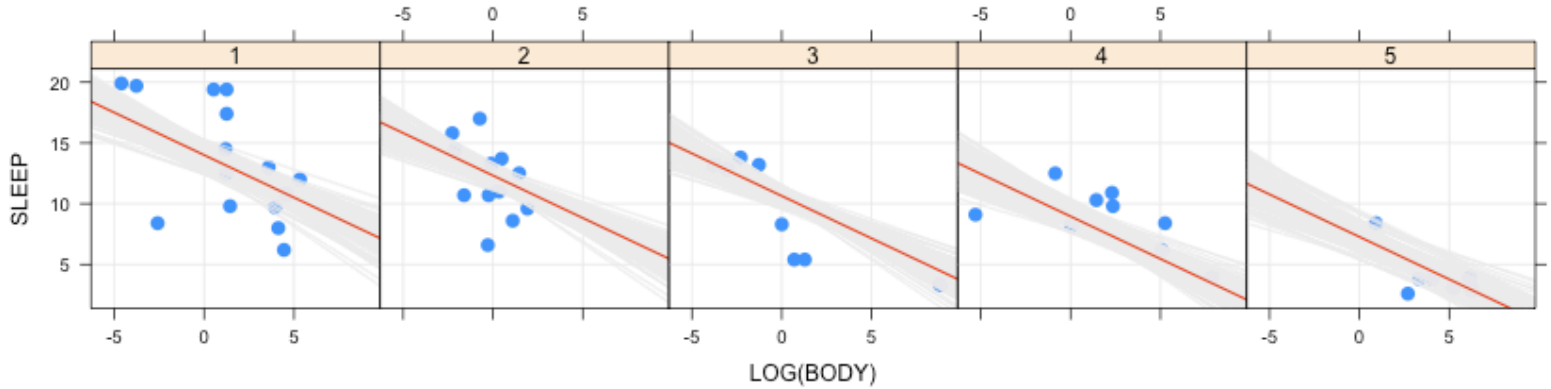
Gestation is better than body alone, but there is both bias and more error without danger too.

Gestation

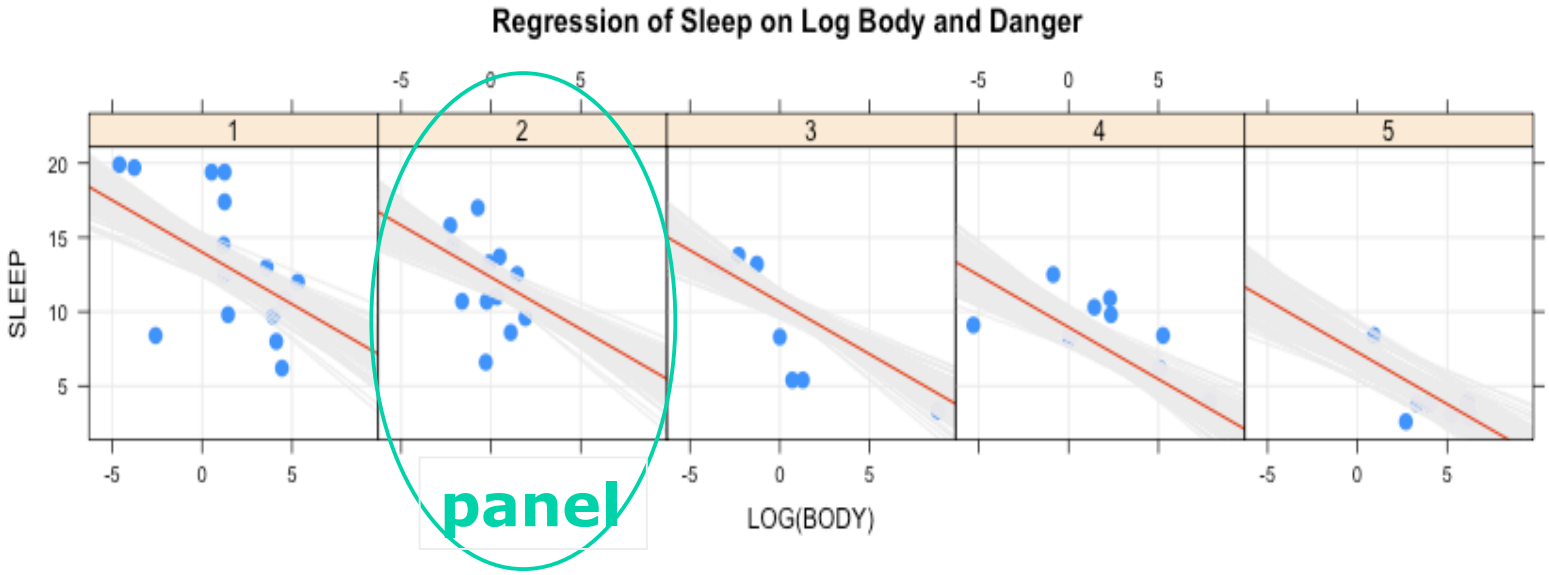


Points all below the lines so there is bias

Body & Danger



A Note About Lattice Plots



R draws the panels sequentially from left to right

panel.number() specifies the panel being drawn now.

To add a horizontal line at 10 to the second panel,

```
if (panel.number() == 2) panel.abline(h = 10)
```

Adding a Discrete Variable

R^2 for the model with log(gestation) plus

predation	danger	exposure
0.56	0.62	0.53

Not a lot of difference. The plots don't seem to suggest that predation or exposure is better than danger, so stay with danger.

(All three are statistically significant.)

log(gestation) + danger

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	24.5966	1.7543	14.021	< 2e-16
gestation	-2.3290	0.4039	-5.767	4.75e-07
danger	-1.4648	0.2988	-4.902	1.00e-05
(Intercept)	23.430	1.881	12.456	< 2e-16
gestation	-2.414	0.411	-5.875	3.89e-07
dangerFact2	-1.481	1.092	-1.356	0.18137
dangerFact3	-3.300	1.154	-2.861	0.00624
dangerFact4	-2.590	1.229	-2.107	0.04034
dangerFact5	-7.005	1.344	-5.211	3.92e-06

Danger as A Factor or Numeric

If the change from one level of the factor danger to the next is the same as the slope of the linear model with danger numeric, then keep danger as numeric

```
numeric

danger        -1.4648

factor

dangerFact2   -1.481

dangerFact3   -3.300

dangerFact4   -2.590

dangerFact5   -7.005
```

In this case, only the change from dangerFact 1 to 2 and 2 to 3 is consistent with danger as a numeric.

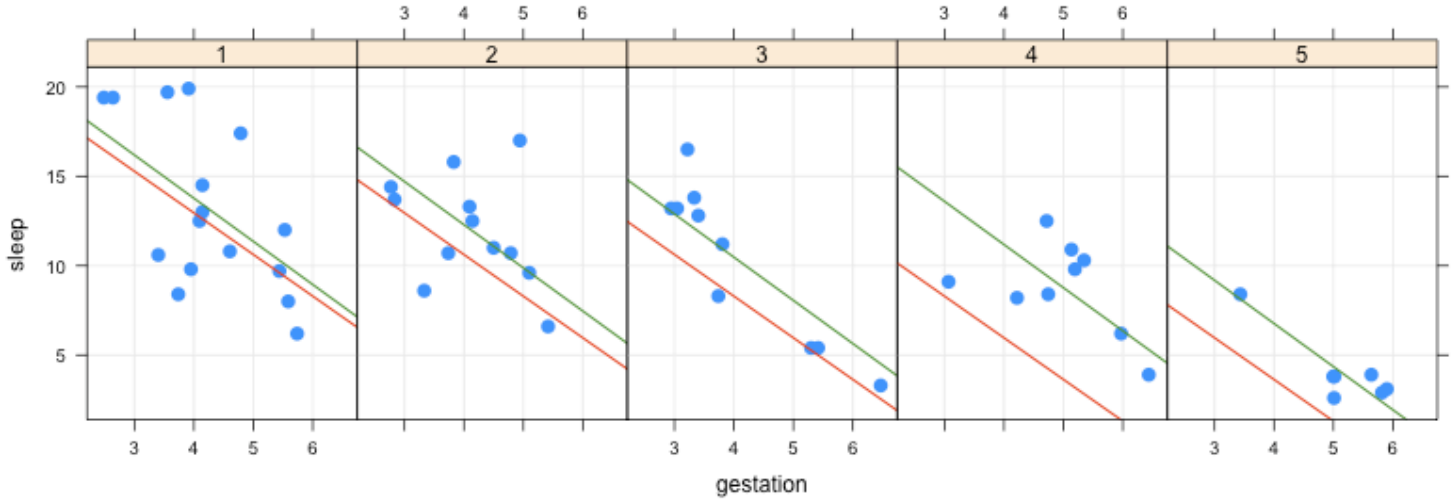
Plot Linear and Additive Lines

```
zL <- lm(sleep ~ gestation+danger, data=sleepLog)
zA <- lm(sleep~gestation+dangerFact,data=sleepLog)
aIntercept <- zA$coef[1] + c(0, zA$coef[3:6])
LIntercept <- zL$coef[1] + seq(5) * zL$coef[2]

xyplot(sleep~gestation|dangerFact, data=sleepLog,
       layout = c(5, 1),
       panel = function(x, y, ...) {
         panel.grid(h = -1, v = -1)
         pno <- panel.number()
         panel.xyplot(x, y, ...)
panel.abline(LIntercept[pno],zL$coef[2],col='red')
panel.abline(aIntercept[pno], zA$coef[2], col=2)
       })
```

Plot Linear And Additive Models

Sleep vs Gestation Given Danger



Red line: linear model

increasing danger by one increases the panel intercept by b_1

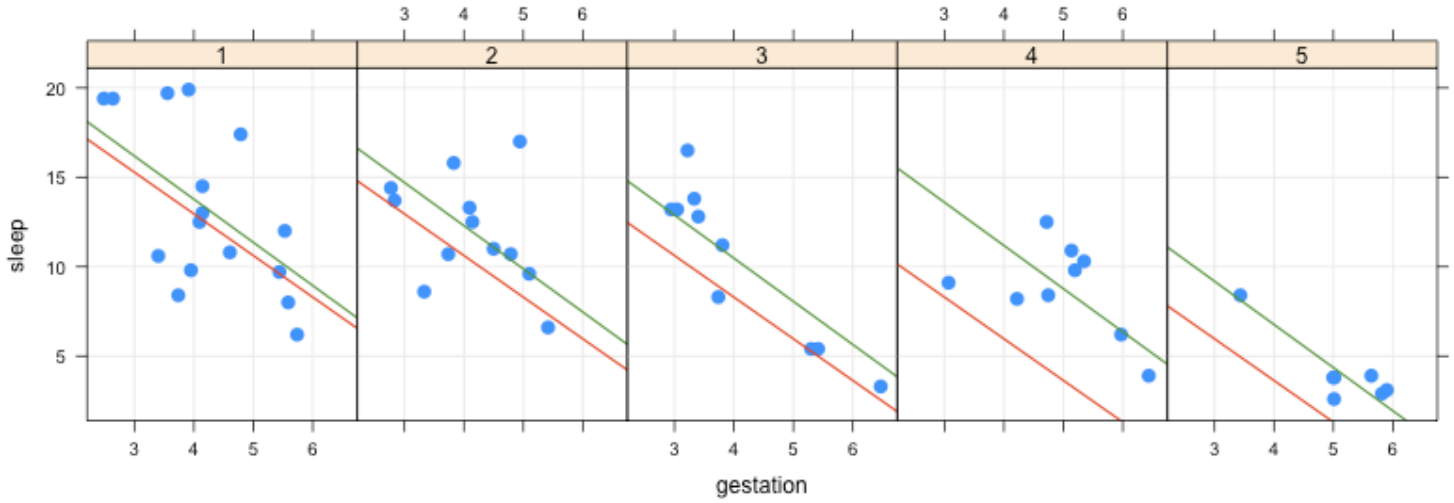
Green line: additive model

the intercepts in the panels are unrelated

fits the data much better -- consistent with the explanation on the previous slide

Plot Linear And Additive Models

Sleep vs Gestation Given Danger



The red line (danger numeric rather than factor) underpredicts for danger above 3.

Additive & Interaction Models

Danger is a factor.

	Estimate	Std. Error	t value
(Intercept)	25.6331	3.1863	8.045
gestation	-2.9349	0.7337	-4.000
dangerFact2	-8.6980	5.2326	-1.662
dangerFact3	-1.6302	4.5744	-0.356
dangerFact4	-10.5745	6.0816	-1.739
dangerFact5	-11.0240	7.7931	-1.415
gestation:dangerFact2	1.7362	1.2297	1.412
gestation:dangerFact3	-0.4320	1.0683	-0.404
gestation:dangerFact4	1.6808	1.2585	1.336
gestation:dangerFact5	0.8753	1.5578	0.562

There is no reason to include the interaction terms (not signif.)

Interaction of Numeric Variables

```
z <- lm(sleep ~ gestation * danger, data = sleepLog)
```

```
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        26.4983     4.0540   6.536 3.19e-08
gestation          -2.7509     0.9060  -3.036   0.0038
danger             -2.2126     1.4660  -1.509   0.1375
gestation:danger    0.1598     0.3066   0.521   0.6045
```

The interaction doesn't improve the fit of the linear model.

This is the same model:

```
z <- lm(sleep ~ gestation + sleep +
        I(gestation * danger), data = sleepLog)
```

Neither Interaction Model is Good

Look at the one with danger as a factor. Get **x** values.

I randomly sampled 2 of the observed not-NA values of log(gestation) conditioning on danger.

	gestation	danger
1	3.637586	1
2	2.639057	1
3	5.099866	2
4	4.094345	2
5	3.044522	3
6	6.469250	3
7	5.347108	4
8	5.192957	4
9	5.899897	5
10	5.638355	5

Uncertainty in the Means at New X's

simEstimates has columns

```
 (Intercept) gestation
dangerFact2   dangerFact3   dangerFact4    dangerFact5
gestation:dangerFact2   gestation:dangerFact3
gestation:dangerFact4   gestation:dangerFact5
```

Each row defines a different linear model

- We want 500 estimates for each of the ten **X**'s, one from each row of simEstimates
- First find the intercept and slope for each **X**
  - intercept and slope for a level of danger other than 1 is the sum of the 'original' value plus the increment for that level

Get the Intercept and Slopes

There are 10 new values of **X**, each with a different intercept and slope

```
intercepts <- array(NA, c(500, 10))
slopes <- array(NA, c(500, 10))
for (i in 1:10) {
  iDang <- newX$danger[i]
  intercepts[, i] <- simEstimates[, 1]
  slopes[, i] <- simEstimates[, 2]
  if (iDang > 1) {
    intercepts[, i] <-
      intercepts[,i] + simEstimates[, 1+iDang]
    slopes[,i] <- slopes[,i]+simEstimates[, 5+iDang]
  }
}
```

Simulated Mean Estimates

```
newMeans <- array(NA, c(500, 10))
for (i in 1:10) {
  newMeans[, i] <- intercepts[, i] +
    slopes[, i] * newX$gestation[i]
}
```

Confidence Interval For The Mean

```
names(newX) <- c('gestation', 'dangerFact')
newX$dangerFact <- factor(newX$dangerFact)
newPredInt <- predict(zI, newdata = newX,
                      interval = 'confidence', level = .9)
   gest  dang fit lwr upr
1     4     1  15  14  16
2     3     1  18  16  20
3     5     2  11   9  13
4     4     2  12  11  13
5     3     3  14  12  16
6     6     3   2  -1   6
7     5     4   8   7  10
8     5     4   8   7  10
9     6     5   2   0   5
10    6     5   3   1   5
```

Fraction of Simulated Means In CI

```
       nBelow nAbove   nIn
 [1,]  0.066  0.038 0.896
 [2,]  0.044  0.034 0.922
 [3,]  0.042  0.042 0.916
 [4,]  0.056  0.030 0.914
 [5,]  0.046  0.036 0.918
 [6,]  0.036  0.044 0.920
 [7,]  0.046  0.052 0.902
 [8,]  0.048  0.050 0.902
 [9,]  0.046  0.046 0.908
[10,]  0.044  0.056 0.900
```

The standard error of a Bin(500, p) sample mean is .013, so the simulated intervals are consistent with normal theory.

Comparing Endpoints

	simLow	ciLow	simHi	ciHi	
1	13.45	13.56	15.68	16.36	
2	16.26	15.59	20.06	20.19	
3	13.45	8.70	15.68	12.94	not so good
4	16.26	10.65	20.06	13.40	
5	13.45	11.74	15.68	15.76	
6	16.26	-1.25	20.06	5.70	not so good
7	13.45	6.65	15.68	10.06	
8	16.26	6.92	20.06	10.17	
9	13.45	-0.09	15.68	5.01	
10	16.26	0.83	20.06	5.16	

End of Linear Regression!