Statistical Methods for High Dimensional Biology

STAT/BIOF/GSAT 540

Lecture 8 – Linear Models Part II

Sara Mostafavi

January 27 2016

Slides by Dr. Jennifer Bryan

outline

- Quick review of previous lecture

- Linear regression & Factorial design

Developing mouse retina – time course for the experiment

So sample collections:

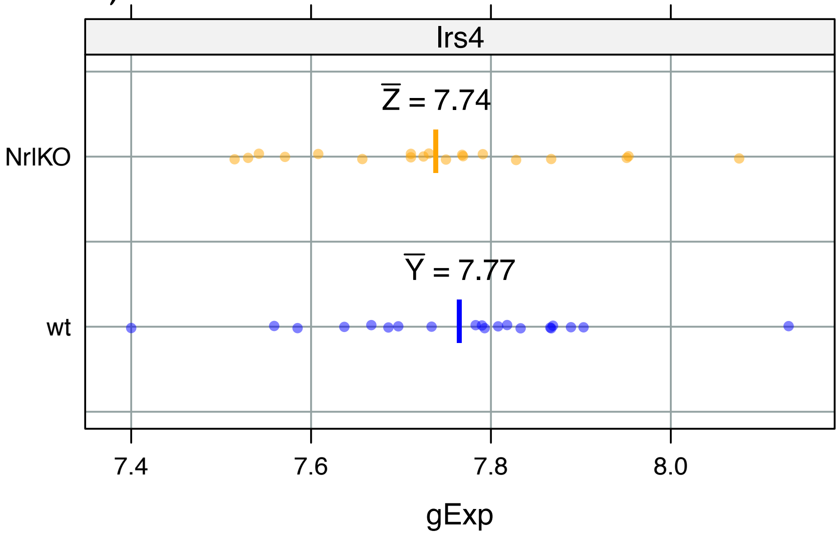
4 developmental stages
2 genotypes: wild-type , Nrl KO

Experimental design

```
devStage   wt  NrlKO
  E16       4      3
  P2        4      4
  P6        4      4
  P10       4      4
  4_weeks   4      4
```

```
> t.test(gExp ~ gType, miniDat,
+        subset = gene == "Irs4", var.equal = TRUE)
```

two sample t test



```
> summary(aov(gExp ~ gType, miniDat,
+             subset = gene == "Irs4"))
```

(one-way) analysis of variance
“ANOVA”

```
> summary(lm(gExp ~ gType, miniDat,
+            subset = gene == "Irs4"))
```

linear model
linear regression

Let's map this notation/formulation to our working example

Group 1 (WT) $Y_1 = \mu_1 + \varepsilon_1 \quad \text{where } \varepsilon_1 \sim F, E(\varepsilon_1) = 0$

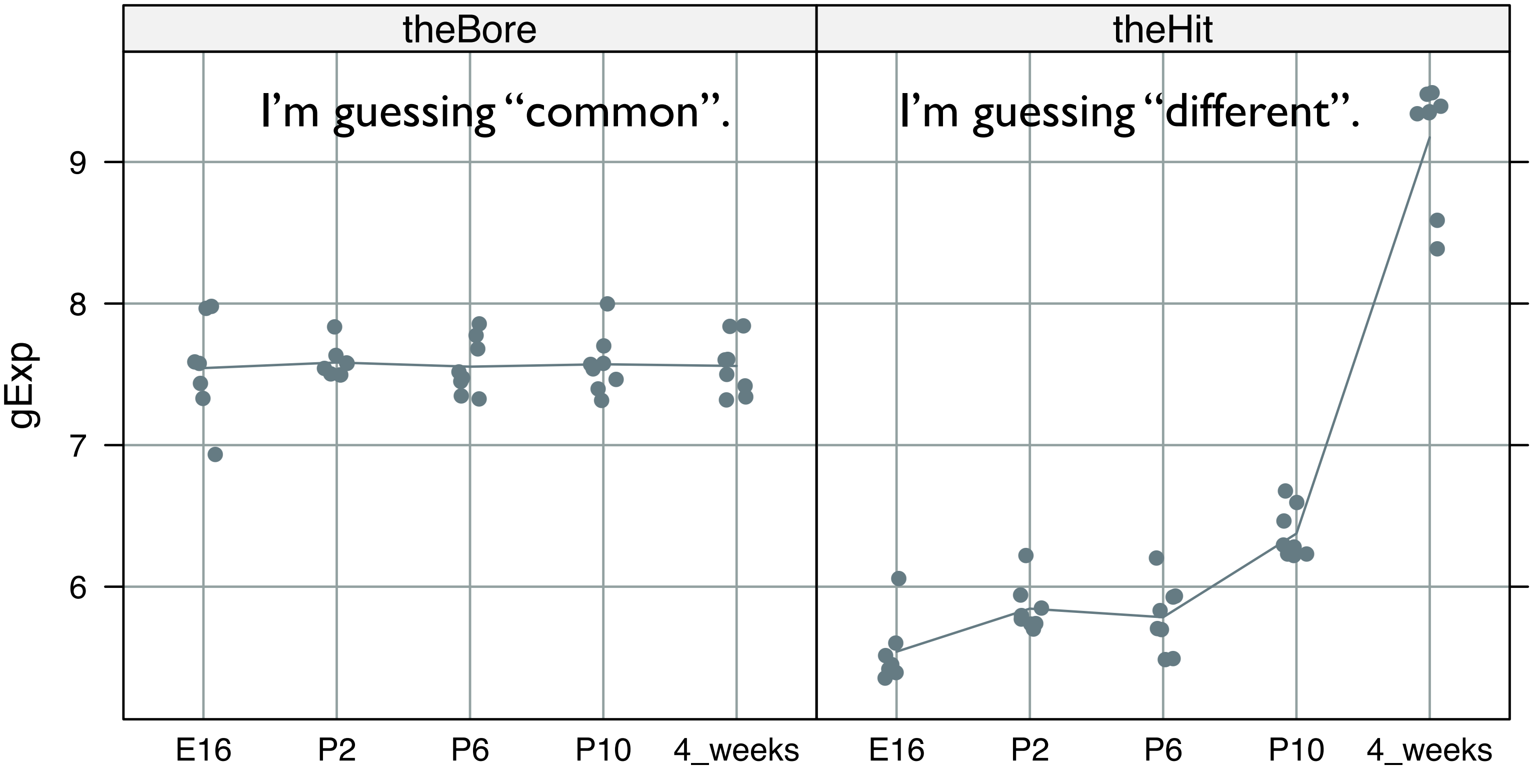
Group 2 (NrlKO) $Y_2 = \mu_2 + \varepsilon_2 \quad \text{where } \varepsilon_2 \sim F, E(\varepsilon_2) = 0$

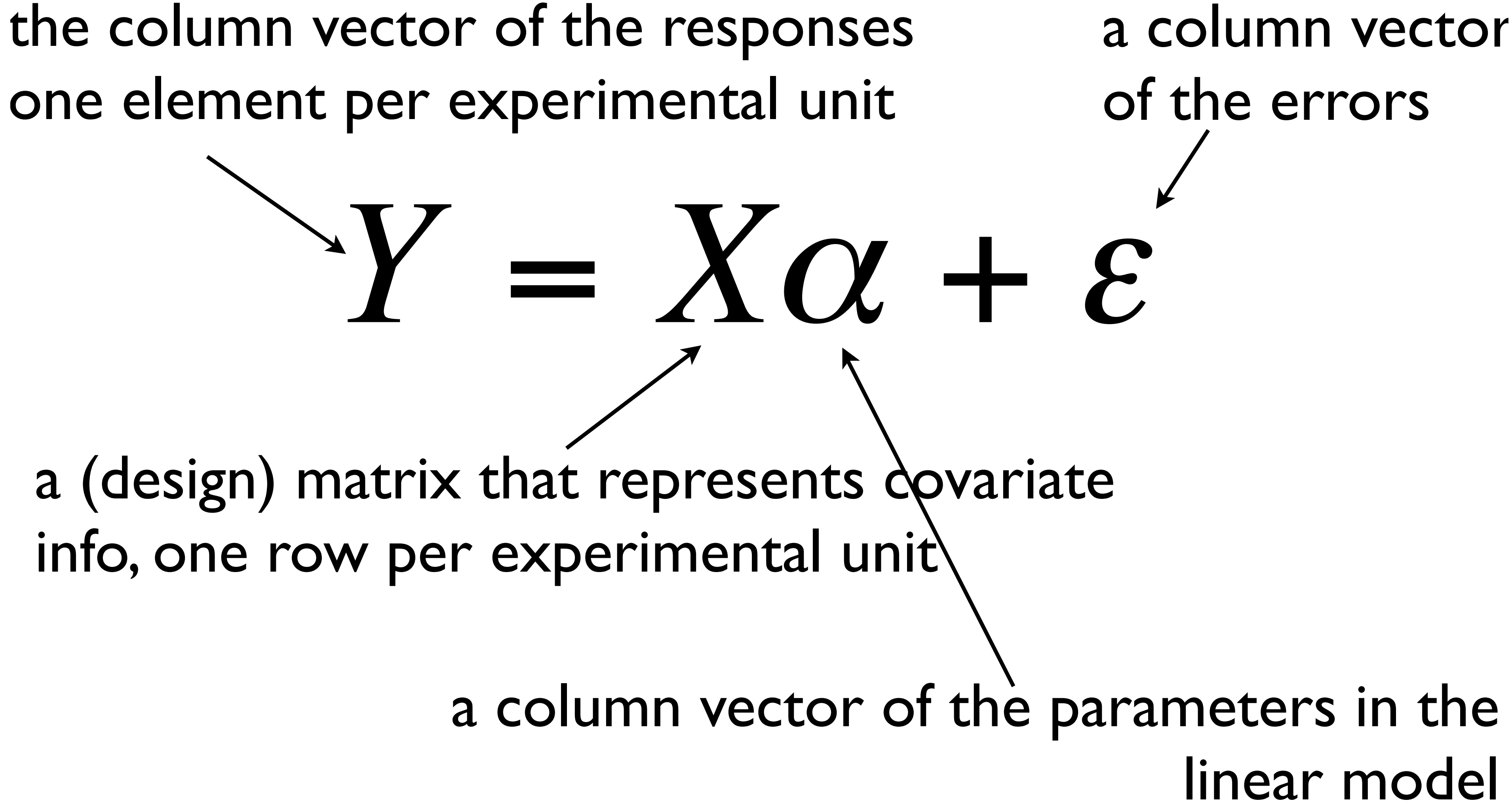
* Note that we have a different expected value μ_j for each group

* With this formulation, we can actually have many groups, not just 2!

* Note that we are assuming the same noise distribution for the two groups (can be relaxed if we think it should be …)

Do we think the expression levels at different developmental stages are generated by different underlying distributions? Or a common one?





the column vector of the responses one element per experimental unit

a column vector of the errors

$$Y = X\alpha + \varepsilon$$

a (design) matrix that represents covariate info, one row per experimental unit

a column vector of the parameters in the linear model

Generic linear model, using conventional matrix formulation

$$Y = X\alpha + \varepsilon$$

Different ways of writing this (design matrix, parameter vector) pair correspond to different parametrizations of the model.

Understanding these concepts makes it easier ...
* to interpret fitted models with confidence
* to fit models such that comparisons you care most about are directly addressed in the inferential “report”

increase the complexity ...

what if you’ve got 2 categorical covariates, e.g. **genotype and developmental stage**?

genotype = wt vs. Nrl knockout

simplifying developmental stage to a two-level factor = E16 (ref) vs. 4_weeks

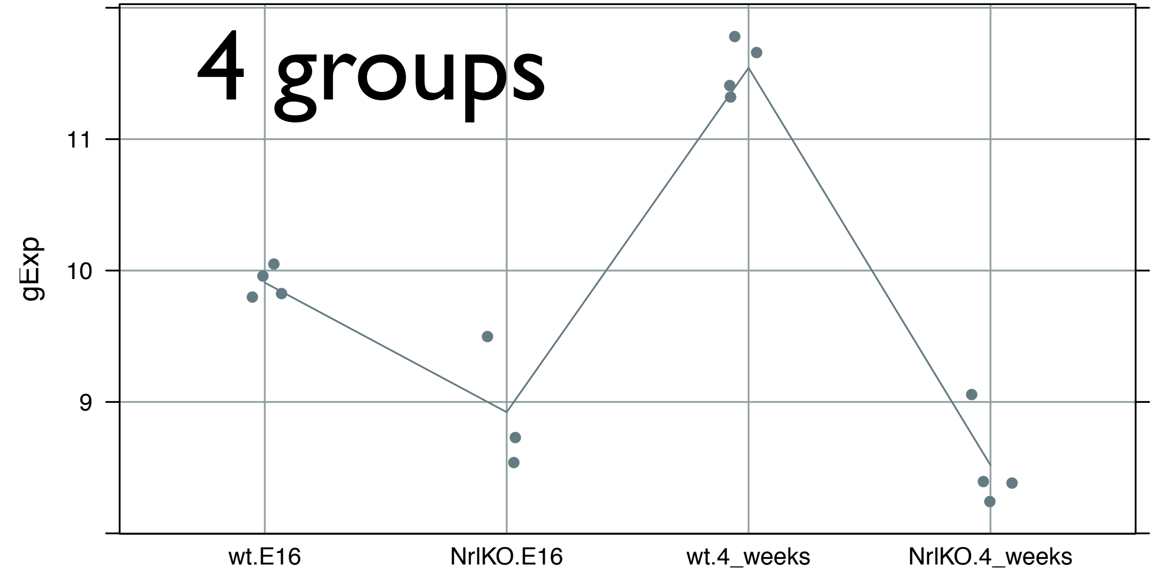
```
> str(miniDat)
'data.frame':	15 obs. of  5 variables:
 $ sample  : num  20 21 22 23 16 17 6 36 37 38 ...
 $ devStage: Factor w/ 2 levels "E16","4_weeks": 1 1 1 1 1 1 1 2 2 2 ...
 $ gType   : Factor w/ 2 levels "wt","NrlKO": 1 1 1 1 2 2 2 1 1 1 ...
 $ gExp    : num  9.96 10.05 9.82 9.8 8.54 ...
 $ grp     : Factor w/ 4 levels "wt.E16","NrlKO.E16",..: 1 1 1 1 2 2 2 3 3 3 ...
```

```
> miniDat
          sample devStage gType   gExp           grp
Sample_20     20      E16    wt  9.958        wt.E16
Sample_21     21      E16    wt 10.050        wt.E16
Sample_22     22      E16    wt  9.825        wt.E16
Sample_23     23      E16    wt  9.799        wt.E16
Sample_16     16      E16 NrlKO  8.539     NrlKO.E16
Sample_17     17      E16 NrlKO  8.730     NrlKO.E16
Sample_6       6      E16 NrlKO  9.498     NrlKO.E16
Sample_36     36  4_weeks    wt 11.410    wt.4_weeks
Sample_37     37  4_weeks    wt 11.780    wt.4_weeks
Sample_38     38  4_weeks    wt 11.320    wt.4_weeks
Sample_39     39  4_weeks    wt 11.660    wt.4_weeks
Sample_11     11  4_weeks NrlKO  8.244 NrlKO.4_weeks
Sample_12     12  4_weeks NrlKO  8.394 NrlKO.4_weeks
Sample_2       2  4_weeks NrlKO  8.382 NrlKO.4_weeks
Sample_9       9  4_weeks NrlKO  9.055 NrlKO.4_weeks
```

```
> with(miniDat, table(gType, devStage))
       devStage
gType   E16 4_weeks
  wt      4       4
  NrlKO   3       4

> table(miniDat$grp)

      wt.E16    NrlKO.E16   wt.4_weeks NrlKO.4_weeks
           4            3            4            4
```

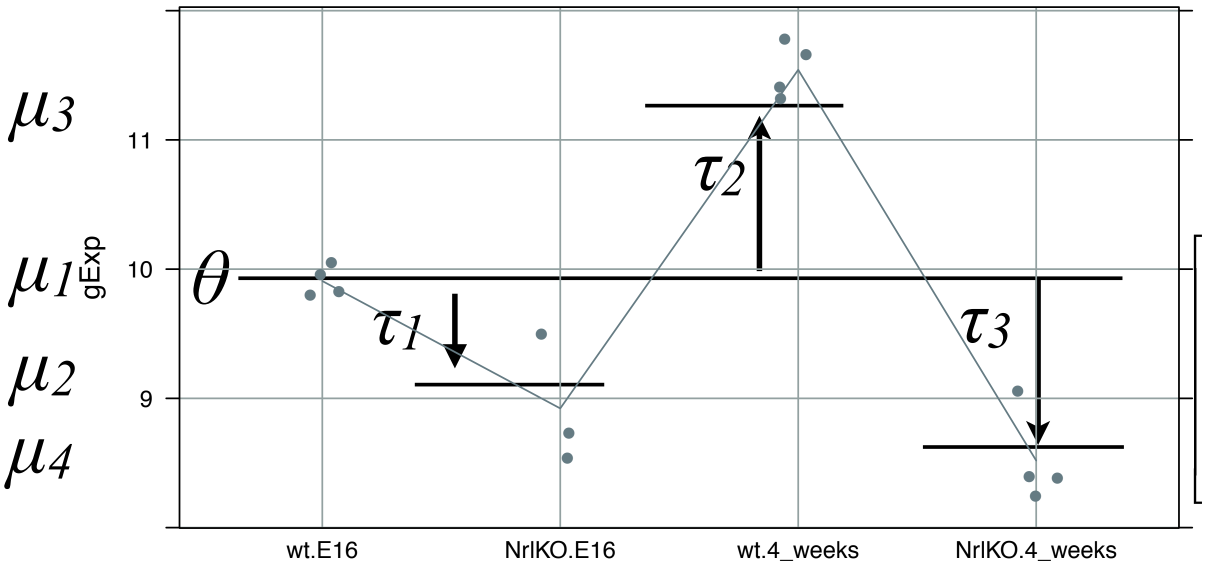
Does it make sense to you to analyze this data like this? What do the parameters of response-trx model mean?

```
> table(miniDat$grp)

      wt.E16     NrlKO.E16    wt.4_weeks NrlKO.4_wee
           4             3             4
```



```
(theAvgs <- with(miniDat, tapply(gExp, grp, mean)))


      wt.E16     NrlKO.E16    wt.4_weeks NrlKO.4_weeks
    9.908000      8.922333     11.542500      8.518750
```

4 groups



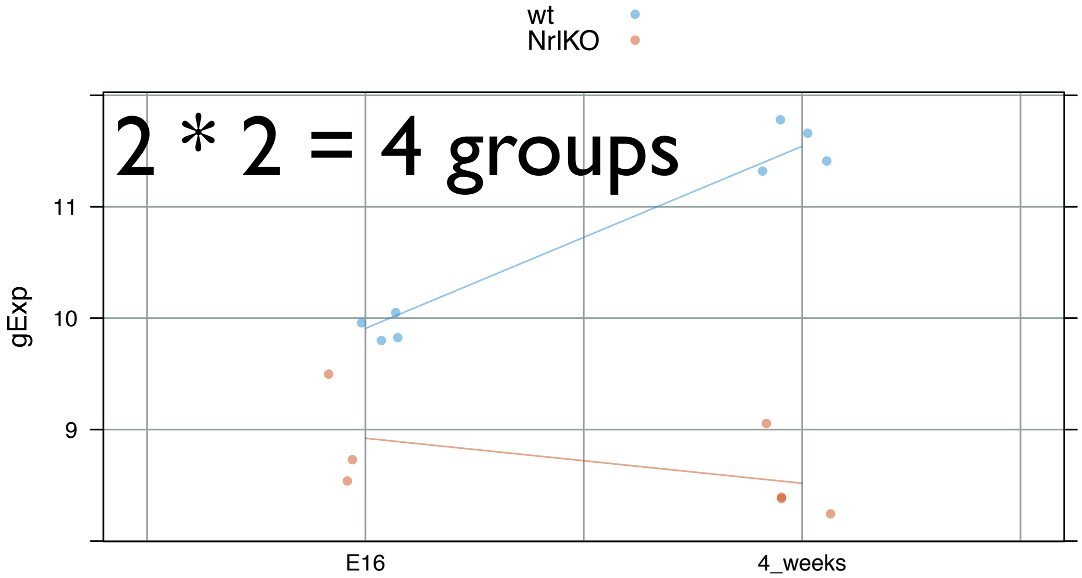
$$Y = X\alpha + \varepsilon$$

$$\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \\ y_{n_4 4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \\ \varepsilon_{n_4 4} \end{bmatrix}$$

model parameters	R	stats
θ	`(Intercept)`	wt, E16
τ_1	`grpNrlKO.E16`	effect of NrlKO
τ_2	`grpwt.4_weeks`	effect of 4_weeks
τ_3	`grpNrlKO.4_weeks`	effect of NrlKO and 4_weeks

More intuitive to model each genotype separately

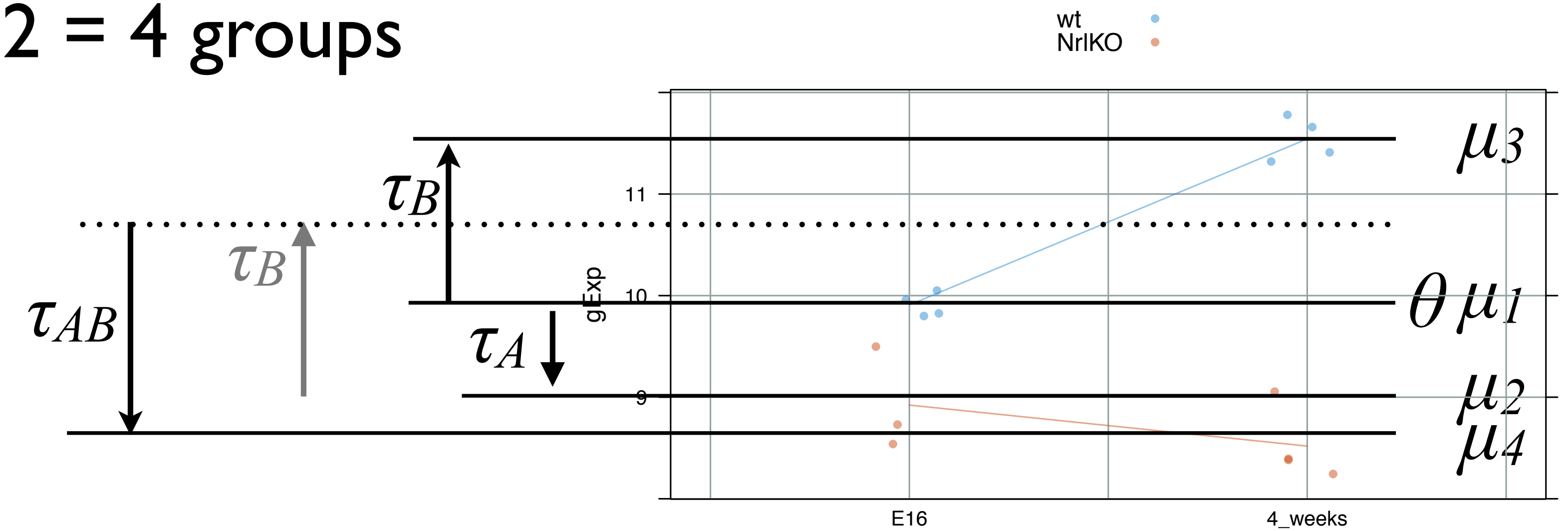
```
> with(miniDat, table(gType, devStage))
          devStage
gType      E16  4_weeks
  wt        4         4
  NrlKO     3         4
```



2 * 2 = 4 groups

```
> with(miniDat,
          tapply(gExp, list(gType, devStage), mean))

           E16   4_weeks
wt    9.908000 11.54250
NrlKO 8.922333  8.51875
```

2 * 2 = 4 groups

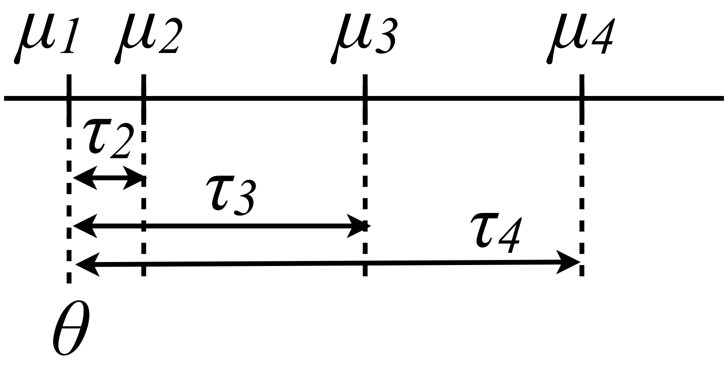


$$Y = X\alpha + \varepsilon$$

$$\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \\ y_{n_4 4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \theta \\ \tau_A \\ \tau_B \\ \tau_{AB} \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \\ \varepsilon_{n_4 4} \end{bmatrix}$$

model paramet	R	stats
θ	`(Intercept)`	wt, E16
τ_A	`gTypeNrlKO`	effect of NrlKO
τ_B	`devStage4_weeks`	effect of 4_weeks
τ_{AB}	`gTypeNrlKO:devStage4_weeks`	interaction effect of NrlKO and 4_weeks

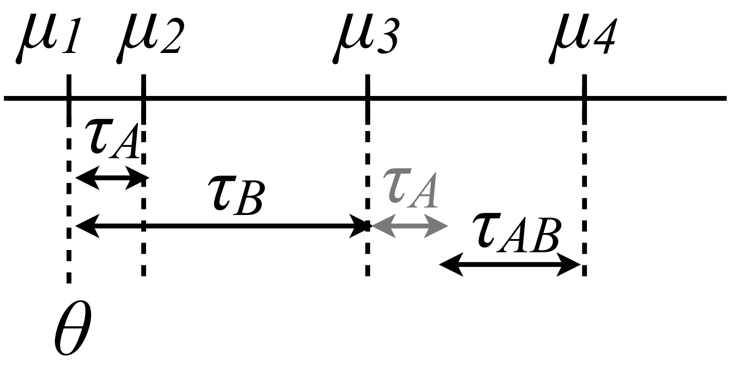
$$\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \\ y_{n_4 3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \\ \varepsilon_{n_4 3} \end{bmatrix}$$



“it’s just 4 groups”

```
lm(y ~ grp)
```

$$\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \\ y_{n_4 3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \theta \\ \tau_A \\ \tau_B \\ \tau_{AB} \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \\ \varepsilon_{n_4 3} \end{bmatrix}$$

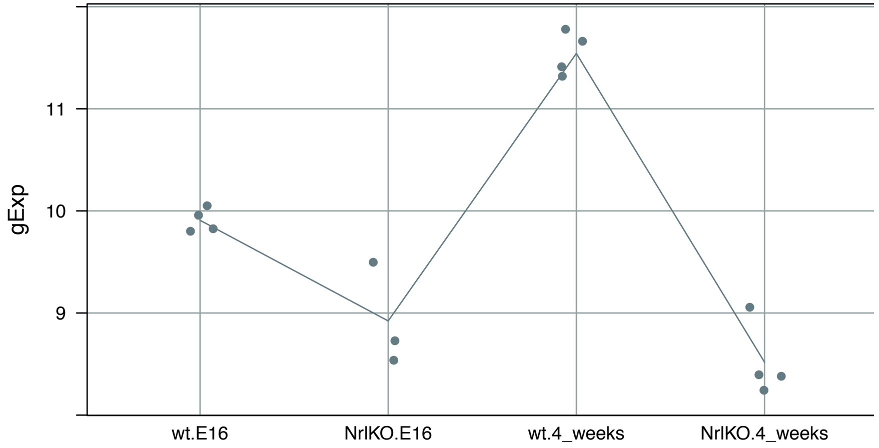


“it’s a 2x2 factorial design”

```
lm(y ~ thingA * thingB)
```

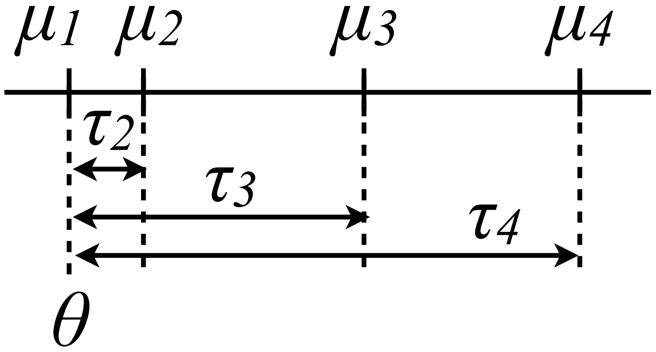
“it’s just 4 groups”

```
> cbind(sampleMeans = theAvgs,
+       minuRef = theAvgs - theAvgs["wt.E16"],
+       grpFit = coef(grpFit))
              sampleMeans    minuRef     grpFit
wt.E16           9.908000  0.0000000  9.9080000
NrlKO.E16        8.922333 -0.9856667 -0.9856667
wt.4_weeks      11.542500  1.6345000  1.6345000
NrlKO.4_weeks    8.518750 -1.3892500 -1.3892500
```



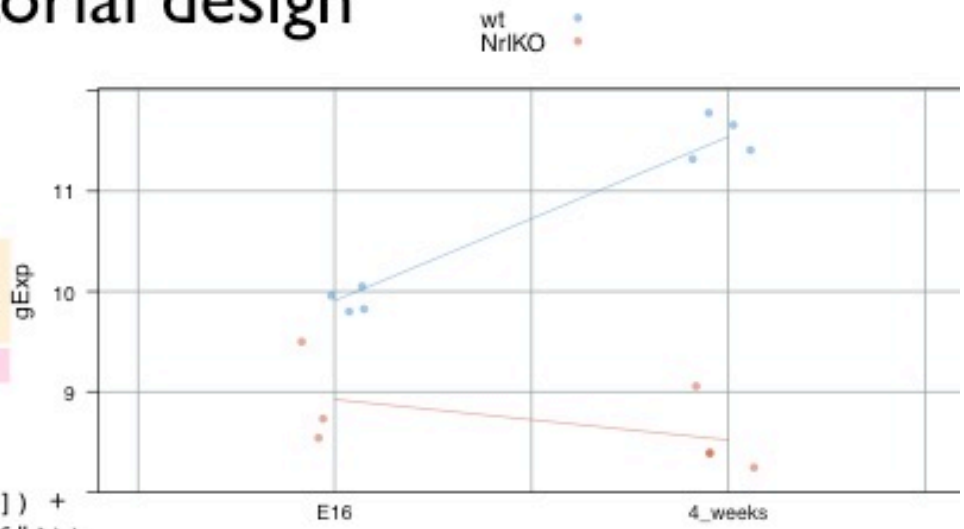
```
> summary(grpFit)
lm(formula = gExp ~ grp, data = miniDat)
<snip, snip>
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        9.9080     0.1575  62.911 2.03e-15 ***
grpNrlKO.E16      -0.9857     0.2406  -4.097  0.00177 **
grpwt.4_weeks      1.6345     0.2227   7.339 1.47e-05 ***
grpNrlKO.4_weeks  -1.3893     0.2227  -6.237 6.37e-05 ***
---
Residual standard error: 0.315 on 11 degrees of freedom
F-statistic: 70.76 on 3 and 11 DF,  p-value: 1.78e-07
```

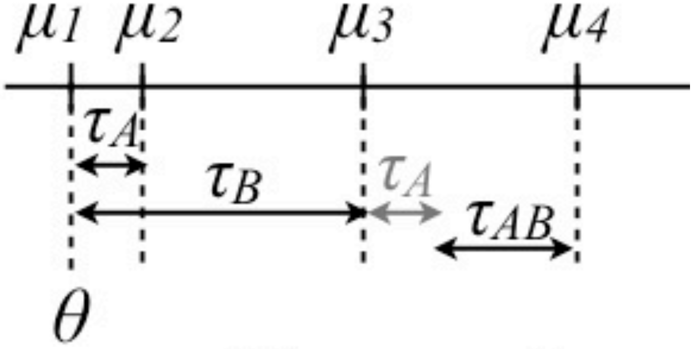
$$H_0 : \tau_j = 0$$



“it’s a 2x2 factorial design”

```
> cbind(sampleMeans = theAvgs,
+       minuRef = theAvgs - theAvgs["wt.E16"],
+       twoFactFit = coef(twoFactFit))
             sampleMeans    minuRef twoFactFit
wt.E16          9.908000  0.0000000  9.9080000
NrlKO.E16       8.922333 -0.9856667 -0.9856667
wt.4_weeks     11.542500  1.6345000  1.6345000
NrlKO.4_weeks   8.518750 -1.3892500 -2.0380833

> theAvgs["NrlKO.4_weeks"] -
+     (theAvgs["wt.E16"] +
+      (theAvgs["NrlKO.E16"] - theAvgs["wt.E16"]) +
+       (theAvgs["wt.4_weeks"] - theAvgs["wt.E16"]))
NrlKO.4_weeks
    -2.038083
```





```
> summary(twoFactFit)
lm(formula = gExp ~ gType * devStage, data = miniDat)
<snip, snip>
Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                   9.9080     0.1575  62.911 2.03e-15 ***
gTypeNrlKO                   -0.9857     0.2406  -4.097  0.00177 **
devStage4_weeks               1.6345     0.2227   7.339 1.47e-05 ***
gTypeNrlKO:devStage4_weeks   -2.0381     0.3278  -6.217 6.56e-05 ***
```

$$H_0 : \tau_A = 0$$

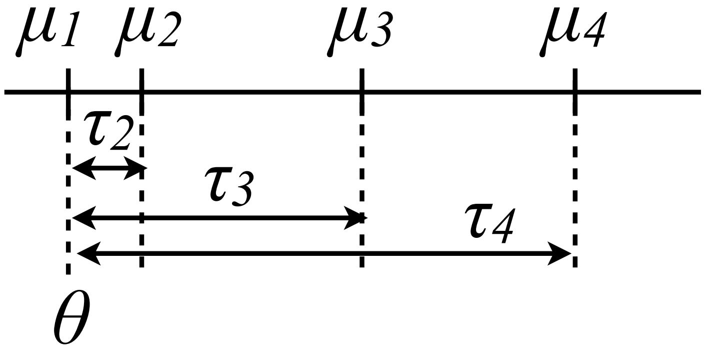
$$H_0 : \tau_B = 0$$

$$H_0 : \tau_{AB} = 0$$

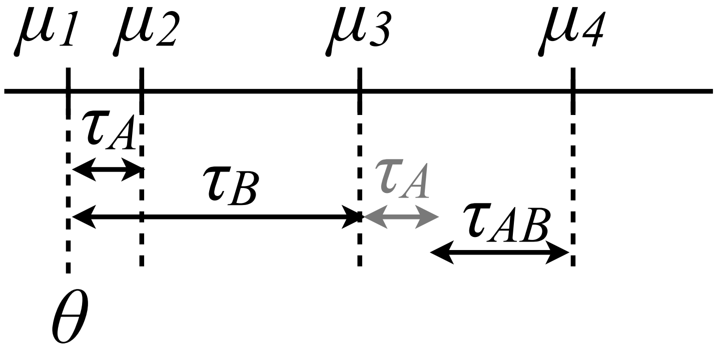
Under the hood, the same linear model is being fit in all three cases*.

However, the analyst is causing the model to be ***parametrized*** differently, in accordance with distinct analytical interests.

```
lm(y ~ 0 + grp)

lm(y ~ grp)
```



```
lm(y ~ thingA * thingB)
```



* Go ahead and check me -- the fitted values and residuals are exactly the same!

optional take-home challenges:

fit the model various ways and verify my claim that the fitted values (see fitted()) and residuals (see resid()) are the same (this, like many simple facts I'm pointing out, will not hold up in messier situations)

try this for yet another way to fit the model:

```
    lm(gExp ~ gType/devStage, miniDat)
```

figure out how that's being parametrized and double check yourself with numbers

Recall: R formulas are expressed in 'Wilkinson-Rogers' notation. See Venables and Ripley 3.7 and 6.2 for an introduction. And/or read Ch. 11 of "An Introduction to R".

hopefully now it is clear how there are different ways to look at data arising from, e.g., four separate groups

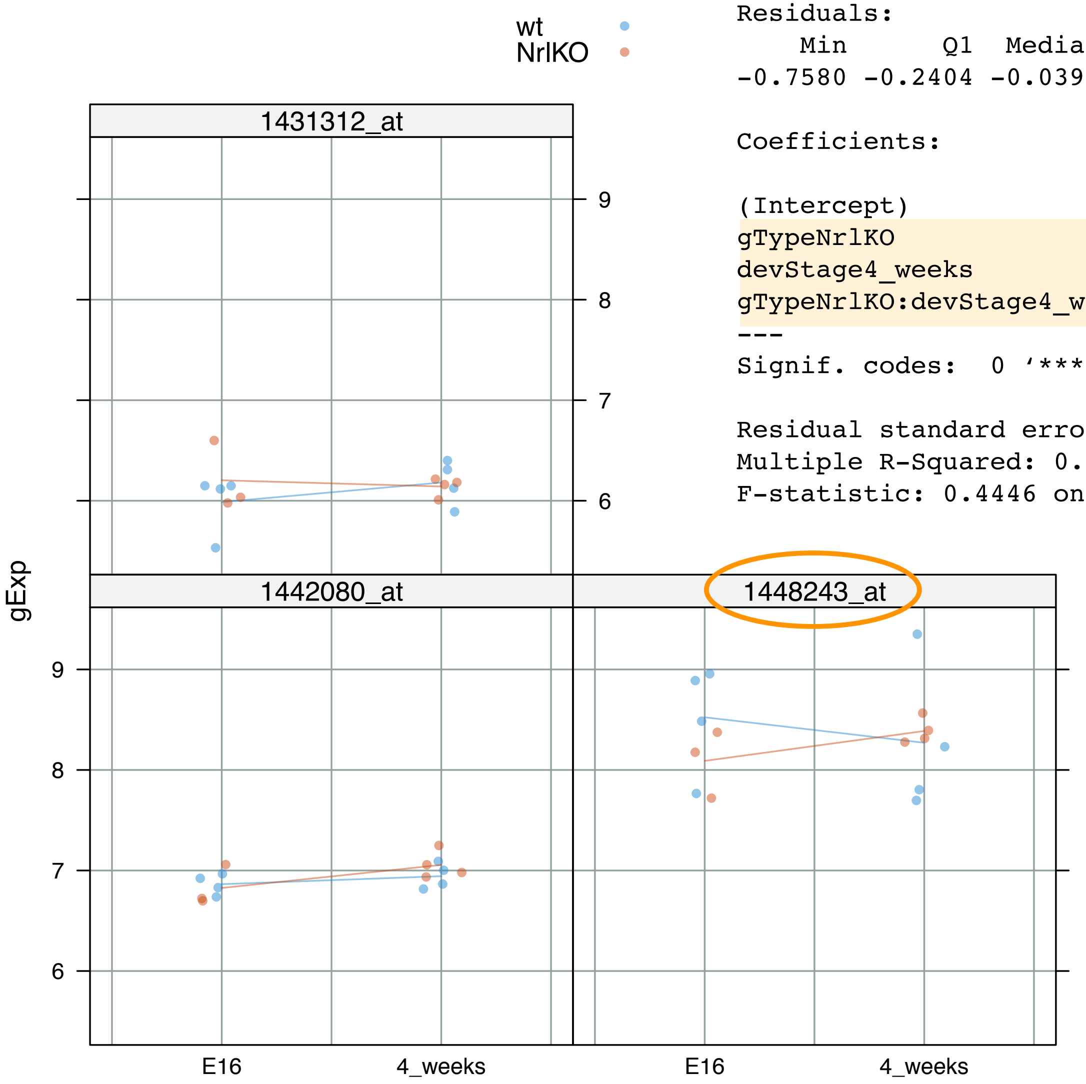
hopefully you now have some sense of how there can be different ways to “parametrize” a model and why you might do that

let’s look at a handful of genes/probesets to get a feel for all the ways a gene could be interesting or boring now ....

approaching with 2x2 factorial mindset

sketch a plot for a boring gene
no knockout effect
no developmental stage effect
no interaction
yawn

boring genes



```
Call:
 lm(formula = prMat ~ gType * devStage, data = prDes)

----------------------------
Response[21641]: 1448243_at

Residuals:
    Min       Q1   Median       Q3      Max 
-0.7580 -0.2404 -0.0390  0.2316  1.0803 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  8.5240     0.2561  33.280 2.15e-12 ***
gTypeNrlKO                  -0.4337     0.3912  -1.108    0.291    
devStage4_weeks             -0.2533     0.3622  -0.699    0.499    
gTypeNrlKO:devStage4_weeks   0.5504     0.5332   1.032    0.324    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 0.5123 on 11 degrees of freedom
Multiple R-Squared: 0.1081,Adjusted R-squared: -0.1351 
F-statistic: 0.4446 on 3 and 11 DF,  p-value: 0.726
```

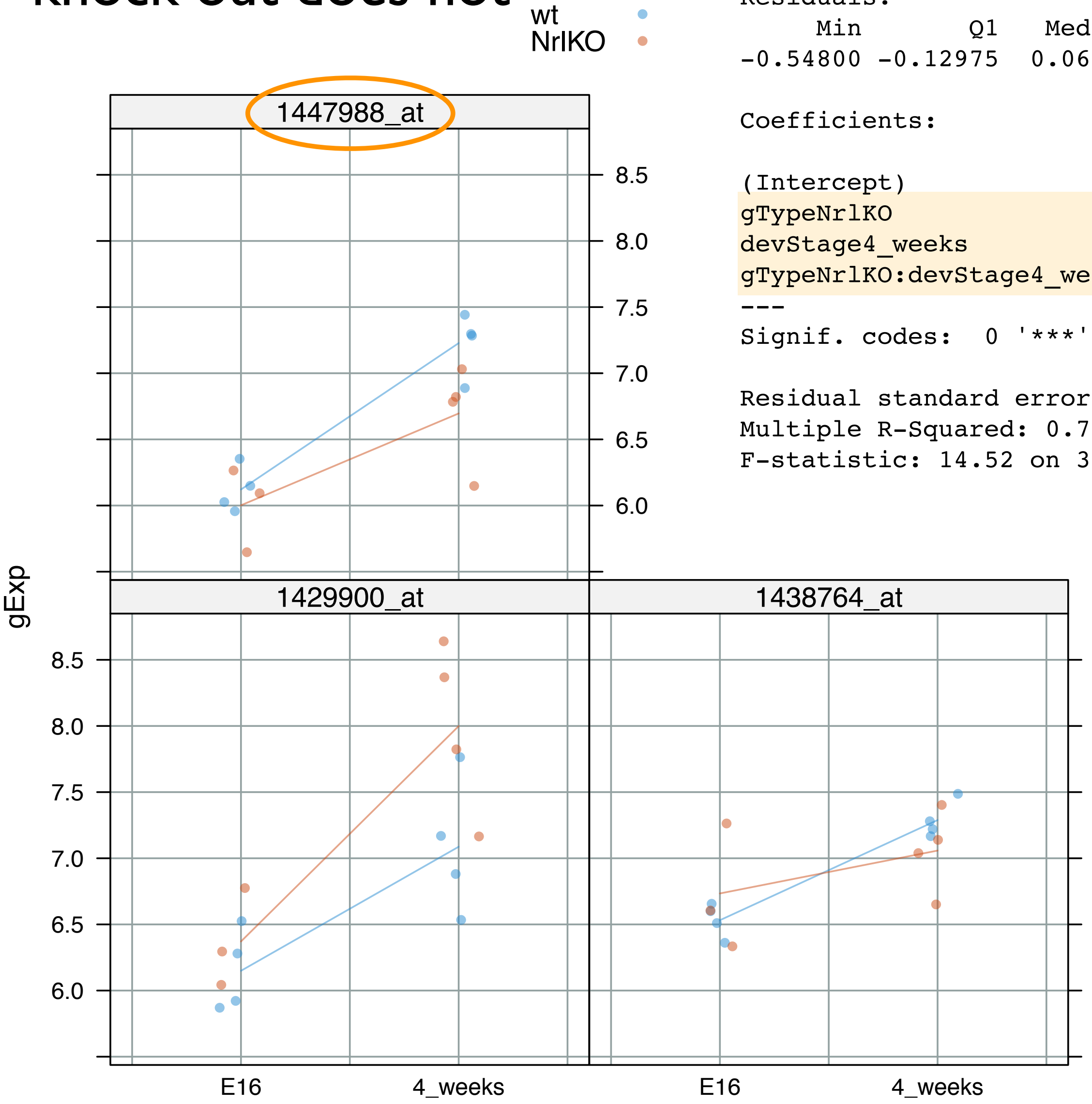
$$H_0 : \tau_{\Delta Nrl} = 0$$ ✔

$$H_0 : \tau_{4\_weeks} = 0$$ ✔

$$H_0 : \tau_{\Delta Nrl, 4\_weeks} = 0$$ ✔

sketch a plot for this:
no knockout effect
YES developmental stage effect
no interaction

developmental stage matters, but gene knock out does not



```
Call:
 lm(formula = prMatSimple ~ gType * devStage)

----------------------------
Response[21450]: 1447988_at

Residuals:
     Min       Q1   Median       Q3      Max 
-0.54800 -0.12975  0.06925  0.16963  0.33500 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  6.1212     0.1430  42.819 1.37e-13 ***
gTypeNrlKO                  -0.1196     0.2184  -0.548 0.594888    
devStage4_weeks              1.1065     0.2022   5.473 0.000194 ***
gTypeNrlKO:devStage4_weeks  -0.4122     0.2976  -1.385 0.193486    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.2859 on 11 degrees of freedom
Multiple R-Squared: 0.7983,Adjusted R-squared: 0.7433 
F-statistic: 14.52 on 3 and 11 DF,  p-value: 0.0003849
```

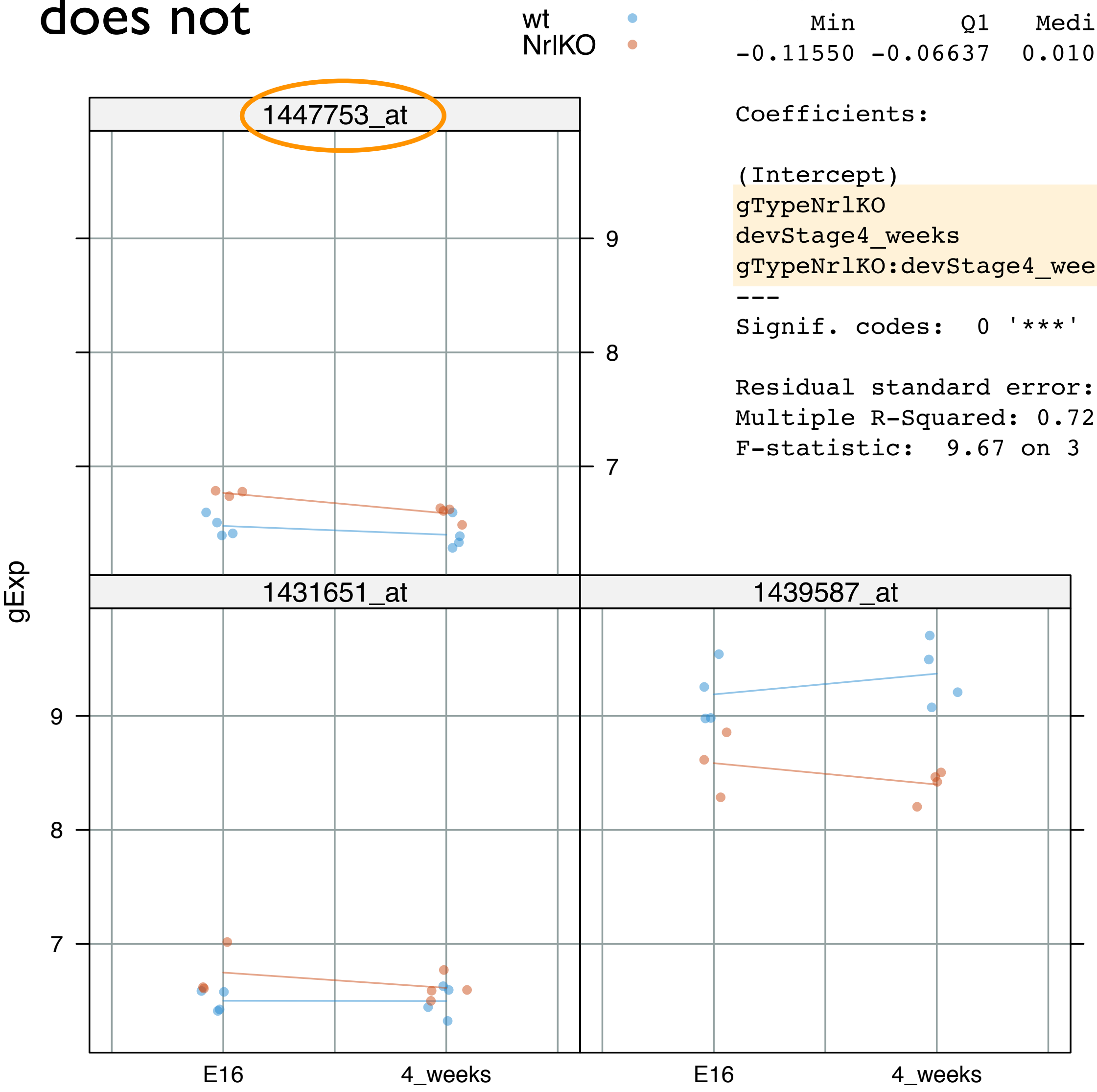
$$H_0 : \tau_{\Delta Nrl} = 0$$ ✓

$$H_0 : \tau_{4\_weeks} \cancel{=} 0$$

$$H_0 : \tau_{\Delta Nrl, 4\_weeks} = 0$$ ✓

sketch a plot for this:
YES knockout effect
no developmental stage effect
no interaction

gene knock out matters, but developmental stage does not



```
Call:
 lm(formula = prMatSimple ~ gType * devStage)

----------------------------
Response[21306]: 1447753_at

Residuals:
     Min       Q1   Median       Q3      Max
-0.11550 -0.06637  0.01067  0.03238  0.19550

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 6.47725    0.04711 137.484  < 2e-16 ***
gTypeNrlKO                  0.29008    0.07197   4.031  0.00198 **
devStage4_weeks            -0.07675    0.06663  -1.152  0.27377
gTypeNrlKO:devStage4_weeks -0.10258    0.09807  -1.046  0.31801
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09423 on 11 degrees of freedom
Multiple R-Squared: 0.7251,Adjusted R-squared: 0.6501
F-statistic:  9.67 on 3 and 11 DF,  p-value: 0.002035
```

$$H_0 : \tau_{\Delta Nrl} \cancel{=} 0$$

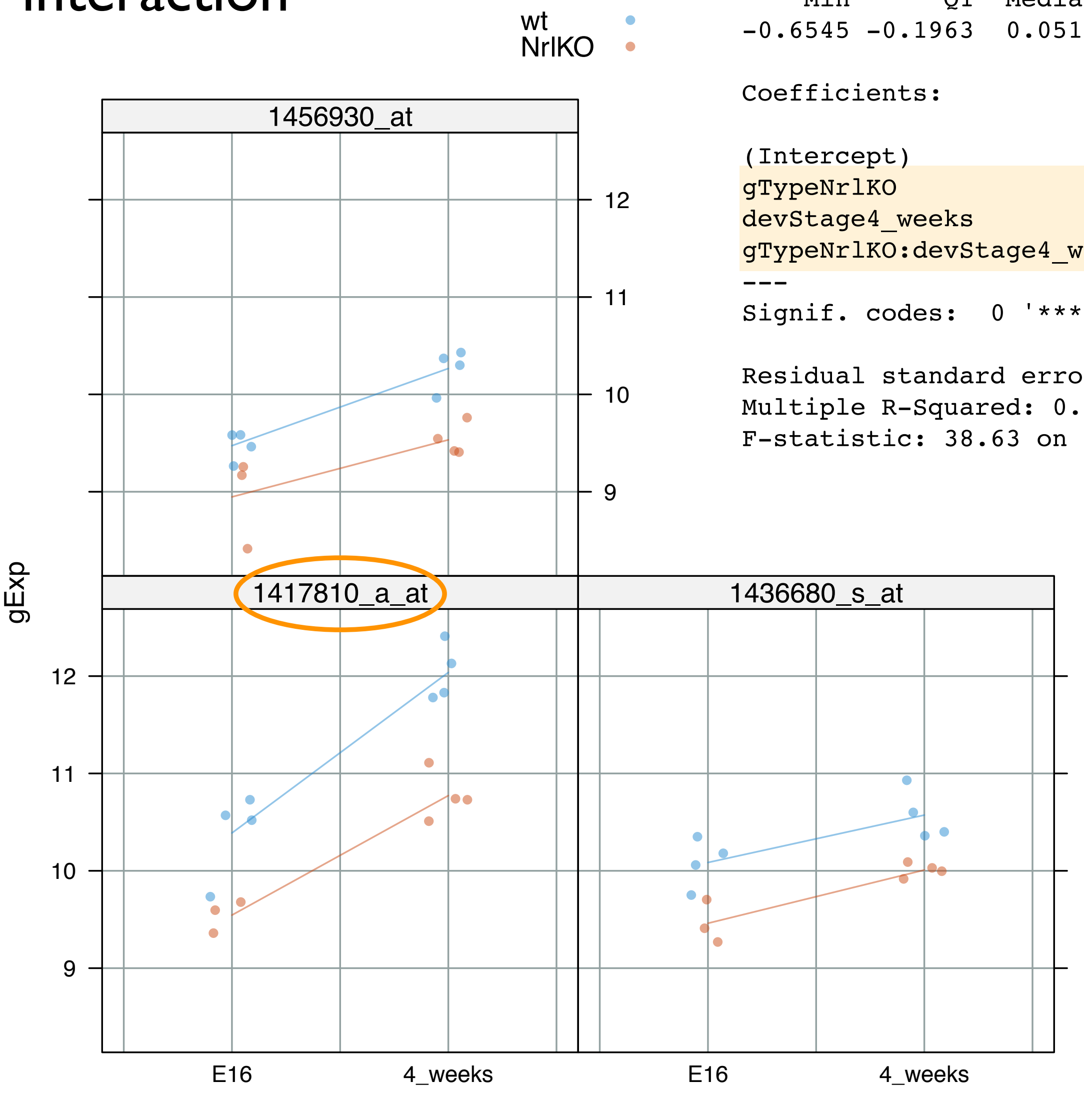
$$H_0 : \tau_{4\_weeks} = 0$$ ✔

$$H_0 : \tau_{\Delta Nrl,4\_weeks} = 0$$ ✔

sketch a plot for this:
YES knockout effect
YES developmental stage effect
no interaction

gene knock out & developmental stage matter, but no interaction

```
Call:
 lm(formula = prMatSimple ~ gType * devStage)

----------------------------
Response[1784]: 1417810_a_at

Residuals:
    Min      Q1  Median      Q3     Max 
-0.6545 -0.1963  0.0510  0.1578  0.3725 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)                10.3885     0.1576  65.932 1.21e-15 ***
gTypeNrlKO                 -0.8435     0.2407  -3.505  0.00493 ** 
devStage4_weeks             1.6490     0.2228   7.400 1.36e-05 ***
gTypeNrlKO:devStage4_weeks -0.4215     0.3280  -1.285  0.22516    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.3151 on 11 degrees of freedom
Multiple R-Squared: 0.9133,Adjusted R-squared: 0.8897 
F-statistic: 38.63 on 3 and 11 DF,  p-value: 3.914e-06
```



$$H_0 : \tau_{\Delta Nrl} \cancel{=} 0$$

$$H_0 : \tau_{4\_weeks} \cancel{=} 0$$

$$H_0 : \tau_{\Delta Nrl,4\_weeks} = 0$$ ✓

sketch a plot for this:

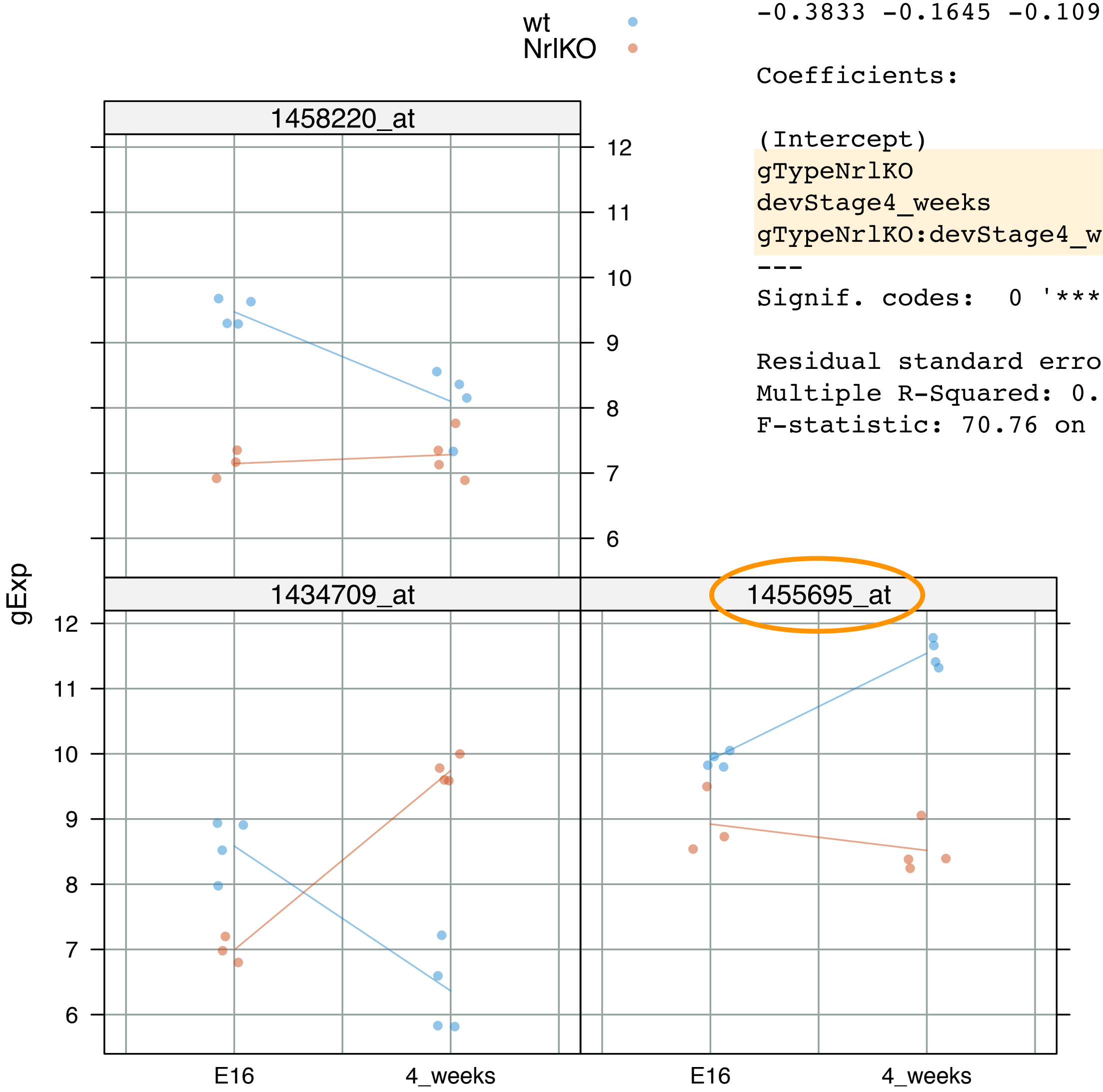
YES knockout effect

YES developmental stage effect

YES interaction

as exciting as it gets, folks

gene knock out & developmental stage matter AND there's interaction



```
Call:
 lm(formula = prMatSimple ~ gType * devStage)

----------------------------
Response[26861]: 1455695_at

Residuals:
    Min      Q1  Median      Q3     Max 
-0.3833 -0.1645 -0.1090  0.1297  0.5757 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  9.9080     0.1575  62.911 2.03e-15 ***
gTypeNrlKO                  -0.9857     0.2406  -4.097  0.00177 ** 
devStage4_weeks              1.6345     0.2227   7.339 1.47e-05 ***
gTypeNrlKO:devStage4_weeks  -2.0381     0.3278  -6.217 6.56e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.315 on 11 degrees of freedom
Multiple R-Squared: 0.9507,Adjusted R-squared: 0.9373 
F-statistic: 70.76 on 3 and 11 DF,  p-value: 1.78e-07 
```

$$H_0 : \tau_{\Delta Nrl} \cancel{=} 0$$

$$H_0 : \tau_{4\_weeks} \cancel{=} 0$$

$$H_0 : \tau_{\Delta Nrl, 4\_weeks} \cancel{=} 0$$

increase the complexity ...

2 categorical covariates:

genotype = wt vs. Nrl knockout

developmental stage = **E16 (ref) vs. P2 vs P6 vs P10 vs 4_weeks**

Challenge:
We will take a “ref + tx effects” and “factorial design” approach.

How many parameters will we be estimating (other than residual variance)?

What are they?

How do they break down in terms of intercept, effects relating to just 1 covariate, interaction effects?

“two-way ANOVA” or ... just a linear model!

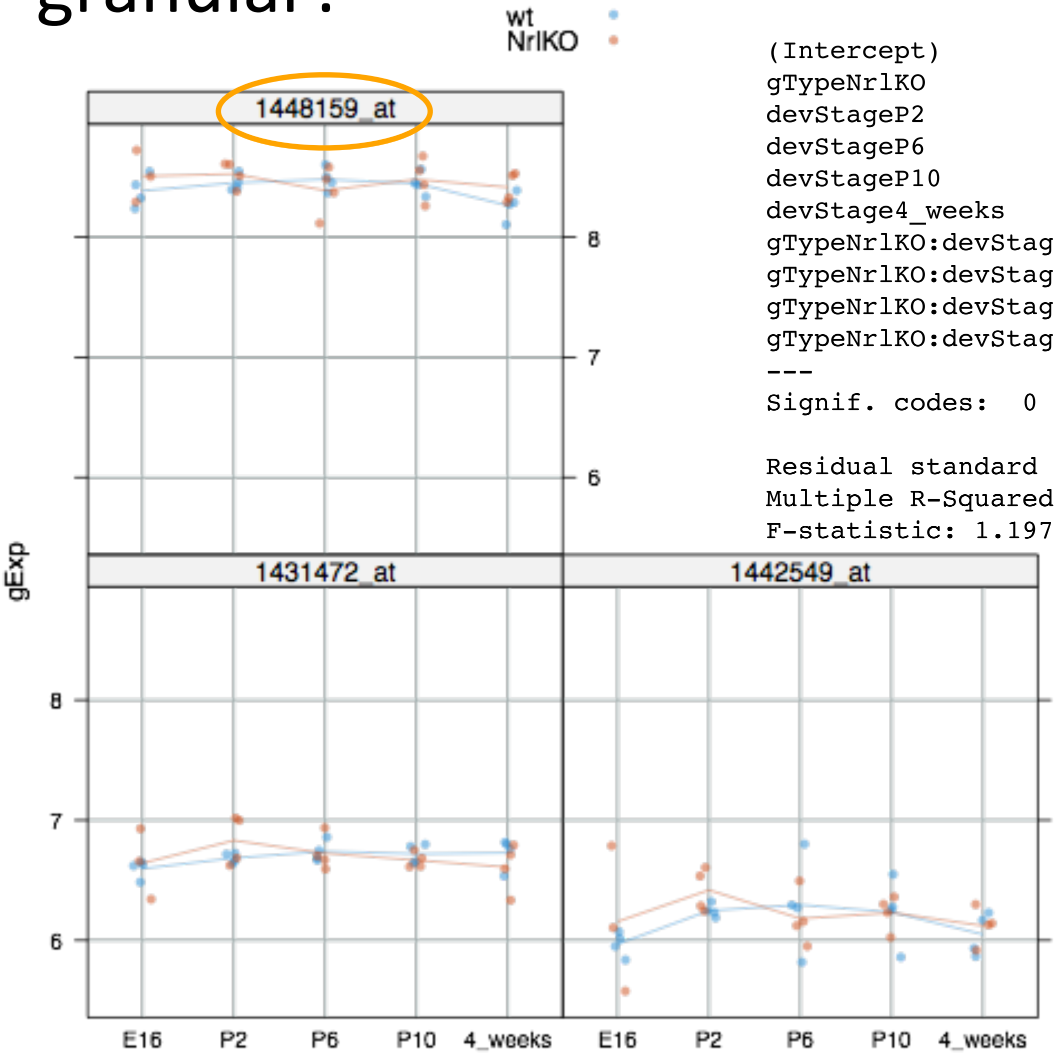
$$y_{ijk} = \theta + \tau_j + \beta_k + (\tau\beta)_{jk} + \varepsilon_{ijk}$$

devStage gType	E16	P2	P6	P10	4_weeks
wt	θ	β_{P2}	β_{P6}	β_{P10}	β_{4_weeks}
NrlKO	τ_{NrlKO}	$(\tau\beta)_{NrlKO,P2}$	$(\tau\beta)_{NrlKO,P6}$	$(\tau\beta)_{NrlKO,P10}$	$(\tau\beta)_{NrlKO,4\_weeks}$

anticipate the plot and inferential results for a boring gene
no knockout effect
no developmental stage effects
no interaction
yawn

linear model style inferential output ... too granular?

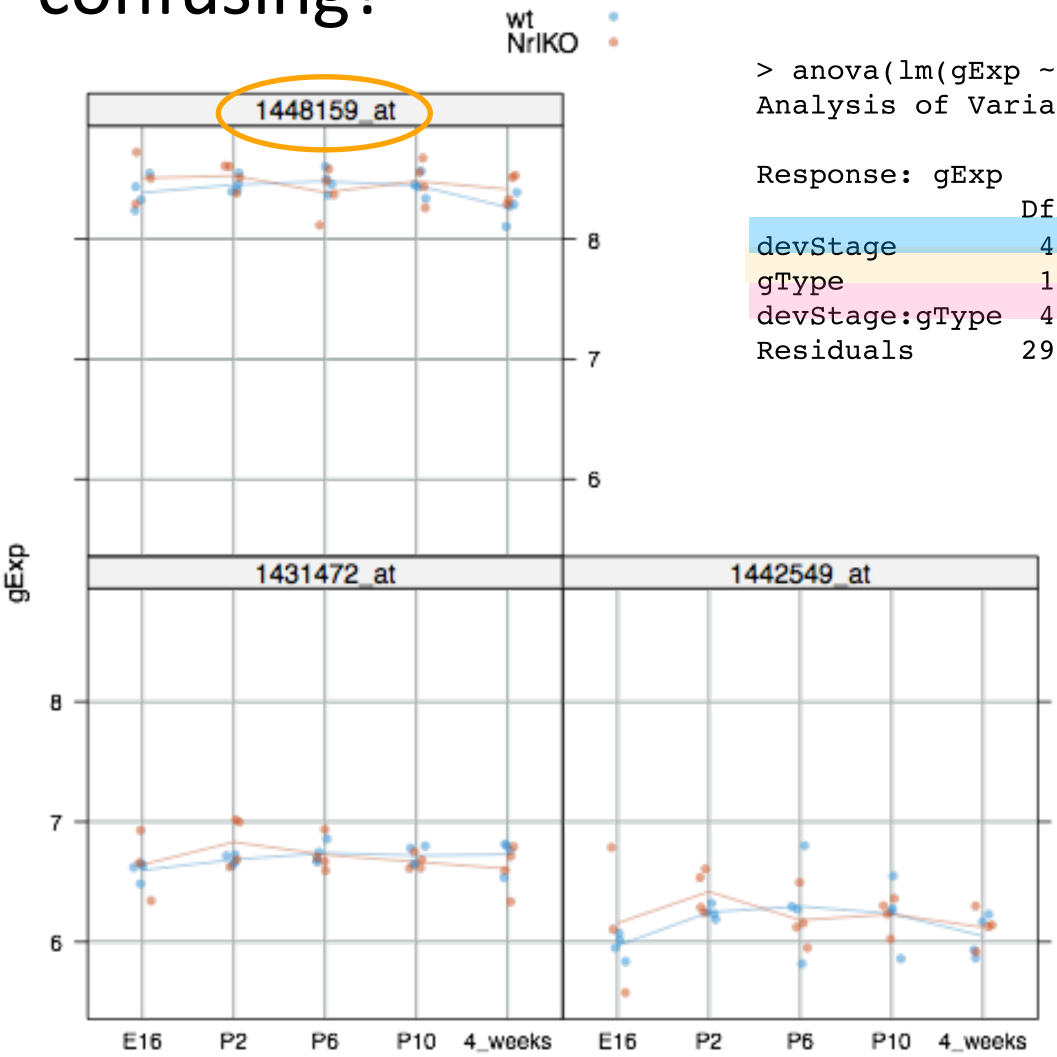
```
Call:
 lm(formula = prMat ~ gType * devStage)

----------------------------
Response[21567]: 1448159_at

Residuals:
    Min       Q1   Median       Q3      Max 
-0.2725 -0.0735  0.0025  0.0955  0.2163 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 8.38600    0.06903 121.475   <2e-16 ***
gTypeNrlKO                  0.12067    0.10545   1.144    0.262    
devStageP2                  0.06550    0.09763   0.671    0.508    
devStageP6                  0.09500    0.09763   0.973    0.339    
devStageP10                 0.06050    0.09763   0.620    0.540    
devStage4_weeks            -0.12300    0.09763  -1.260    0.218    
gTypeNrlKO:devStageP2      -0.04617    0.14371  -0.321    0.750    
gTypeNrlKO:devStageP6      -0.21417    0.14371  -1.490    0.147    
gTypeNrlKO:devStageP10     -0.08617    0.14371  -0.600    0.553    
gTypeNrlKO:devStage4_weeks  0.03133    0.14371   0.218    0.829    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.1381 on 29 degrees of freedom
Multiple R-Squared: 0.2709,   Adjusted R-squared: 0.04463 
F-statistic: 1.197 on 9 and 29 DF,  p-value: 0.3339
```



two-way ANOVA style inferential output ... too confusing?

```
> anova(lm(gExp ~ gType * devStage, jDat))
Analysis of Variance Table

Response: gExp
               Df  Sum Sq  Mean Sq F value Pr(>F)
gType           1 0.02985 0.029848  1.5657 0.2208
devStage        4 0.10365 0.025914  1.3594 0.2722
gType:devStage  4 0.07191 0.017977  0.9430 0.4532
Residuals      29 0.55283 0.019063
```

```
> anova(lm(gExp ~ devStage * gType, jDat))
Analysis of Variance Table

Response: gExp
               Df  Sum Sq  Mean Sq F value Pr(>F)
devStage        4 0.10328 0.025819  1.3544 0.2739
gType           1 0.03022 0.030225  1.5855 0.2180
devStage:gType  4 0.07191 0.017977  0.9430 0.4532
Residuals      29 0.55283 0.019063
```



ANOVA tables address whether, e.g., all the interaction effects, are non-zero

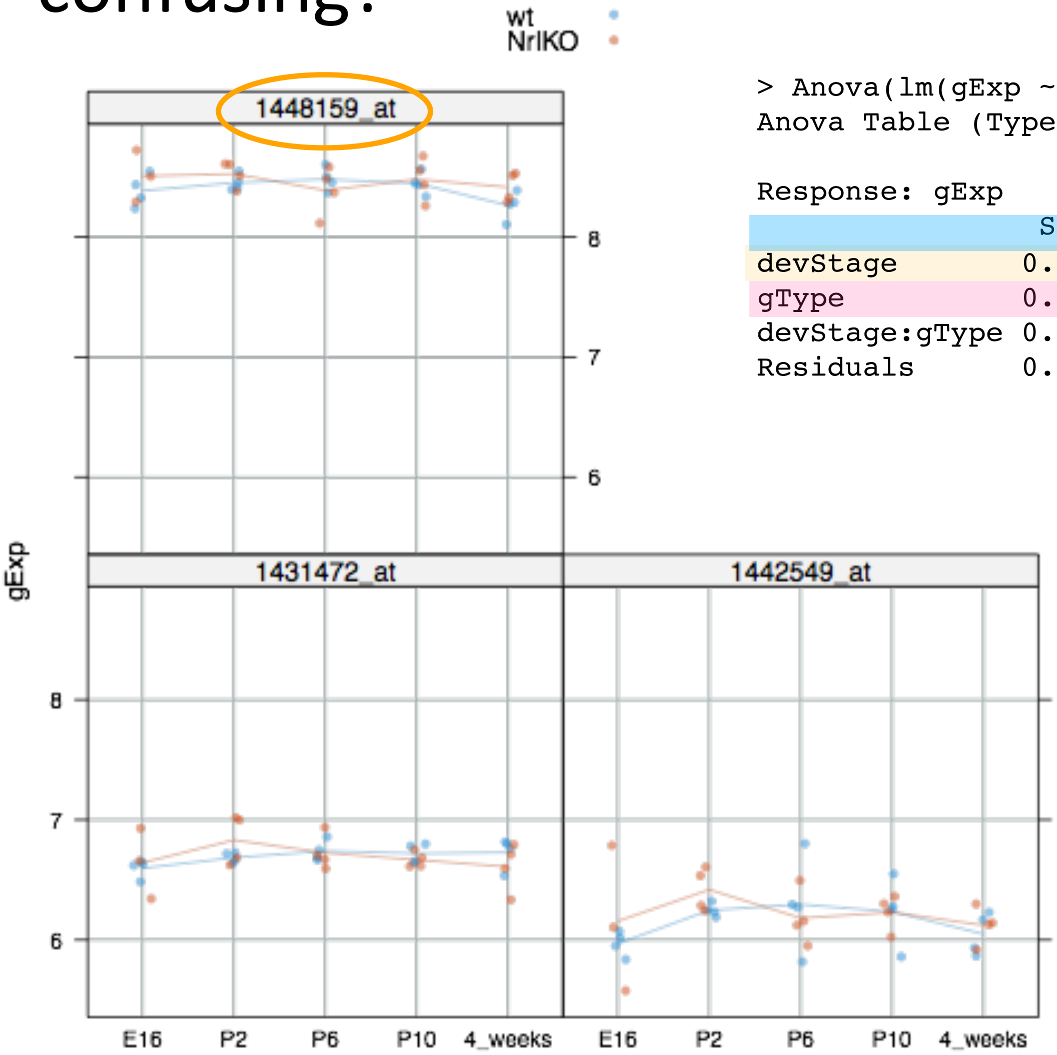
note the agreement above for the interaction gType:devStage

note the discrepancies above for main effects ... depends on order ... related to the sequential nature of Type I sums of squares

we are suffering for our unbalanced design :(

two-way ANOVA style inferential output … too confusing?

```
> Anova(lm(gExp ~ gType * devStage, jDat))
Anova Table (Type II tests)

Response: gExp
                Sum Sq Df F value Pr(>F)
gType          0.03022  1  1.5855 0.2180
devStage       0.10365  4  1.3594 0.2722
gType:devStage 0.07191  4  0.9430 0.4532
Residuals      0.55283 29
```



```
> Anova(lm(gExp ~ devStage * gType, jDat))
Anova Table (Type II tests)

Response: gExp
                Sum Sq Df F value Pr(>F)
devStage       0.10365  4  1.3594 0.2722
gType          0.03022  1  1.5855 0.2180
devStage:gType 0.07191  4  0.9430 0.4532
Residuals      0.55283 29
```

Anova() from the car package computes Type II sums of squares which are non-sequential

tests for each main effect after the other main effect

arguably only makes real sense in the absence of interaction?

F tests in regression

small model is nested within big -- it's a special case where some parameters are equal to zero

model	example	# params = DF	RSS
small	lm(y ~ gType + devStage)	$p_{small} = 6$	RSS_{small}
big	lm(y ~ gType * devStage)	$p_{big} = 10$	RSS_{big}

$$y_{ijk} = \theta + \tau_j + \beta_k + (\tau\beta)_{jk} + \varepsilon_{ijk} \quad \text{“big”}$$

$$y_{ijk} = \theta + \tau_j + \beta_k + (\cancel{\tau\beta})_{jk} + \varepsilon_{ijk} \quad \text{“small”}$$

by definition:

$p_{small} < p_{big}$

$RSS_{small} \geq RSS_{big}$

$$F = \frac{\left(\dfrac{RSS_{small} - RSS_{big}}{p_{big} - p_{small}}\right)}{\dfrac{RSS_{big}}{n - p_{big}}} \sim_{H_0} F_{(p_{big} - p_{small}, n - p_{big})}$$

we can’t replicate an entire linear models course here .... and you won’t be using single-dataset tools like lm() or anova(lm()) for much longer anyway

good rules of thumb

try to have a balanced experiment!

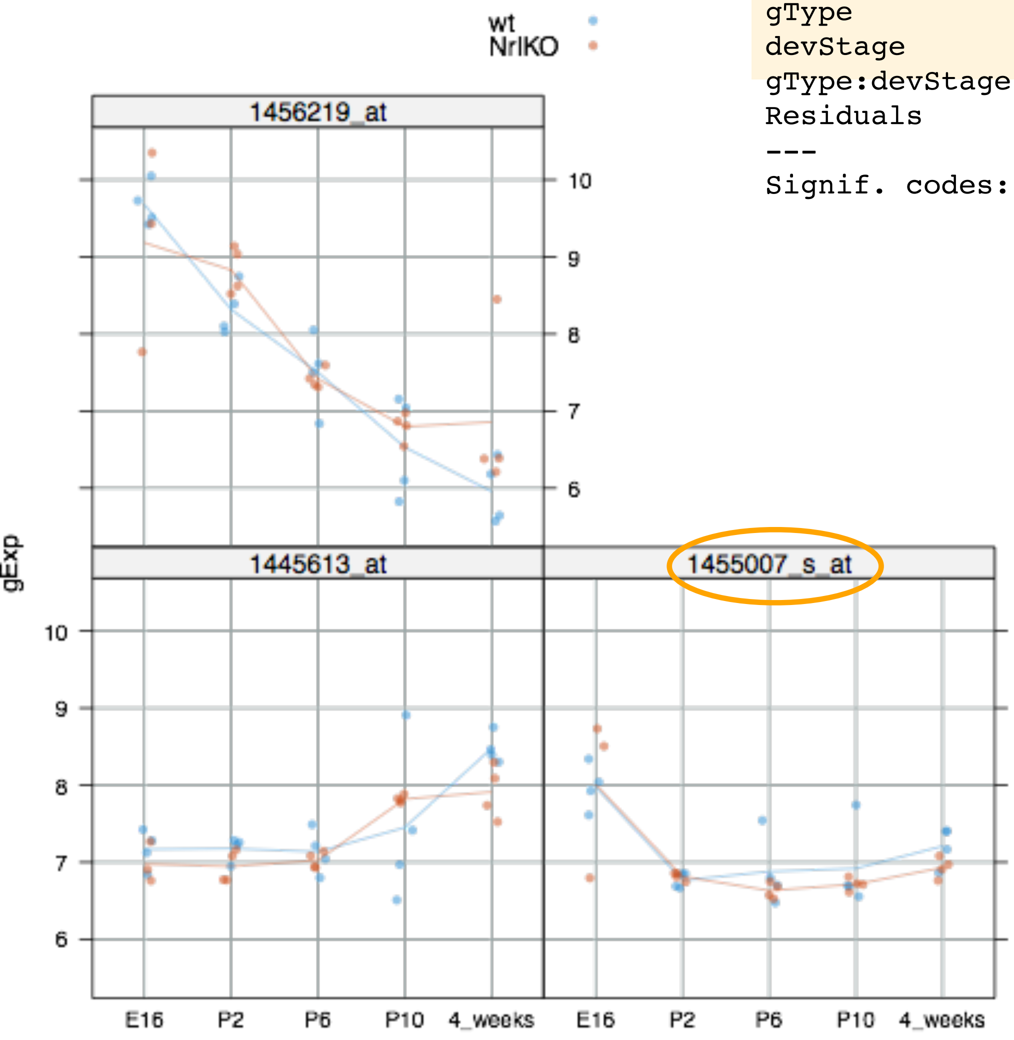
first check for interaction, e.g. using anova()
the presence / absence of interaction should influence how vigorously you delve into and interpret main effects of gType or devStage

good references for further reading about unbalanced designs:
http://goanna.cs.rmit.edu.au/~fscholer/anova.php
http://prometheus.scp.rochester.edu/zlab/sites/default/files/InteractionsAndTypesOfSS.pdf

interaction	gType main effect	devStage main effect	the deal
no	no	no	boring
no	no	yes	only devStage matters
no	yes	no	only gType matters
no	yes	yes	both matter but don’t interact
yes	no	no	weird and I don’t go here
yes	no	yes	
yes	yes	no	
yes	yes	yes	exciting!

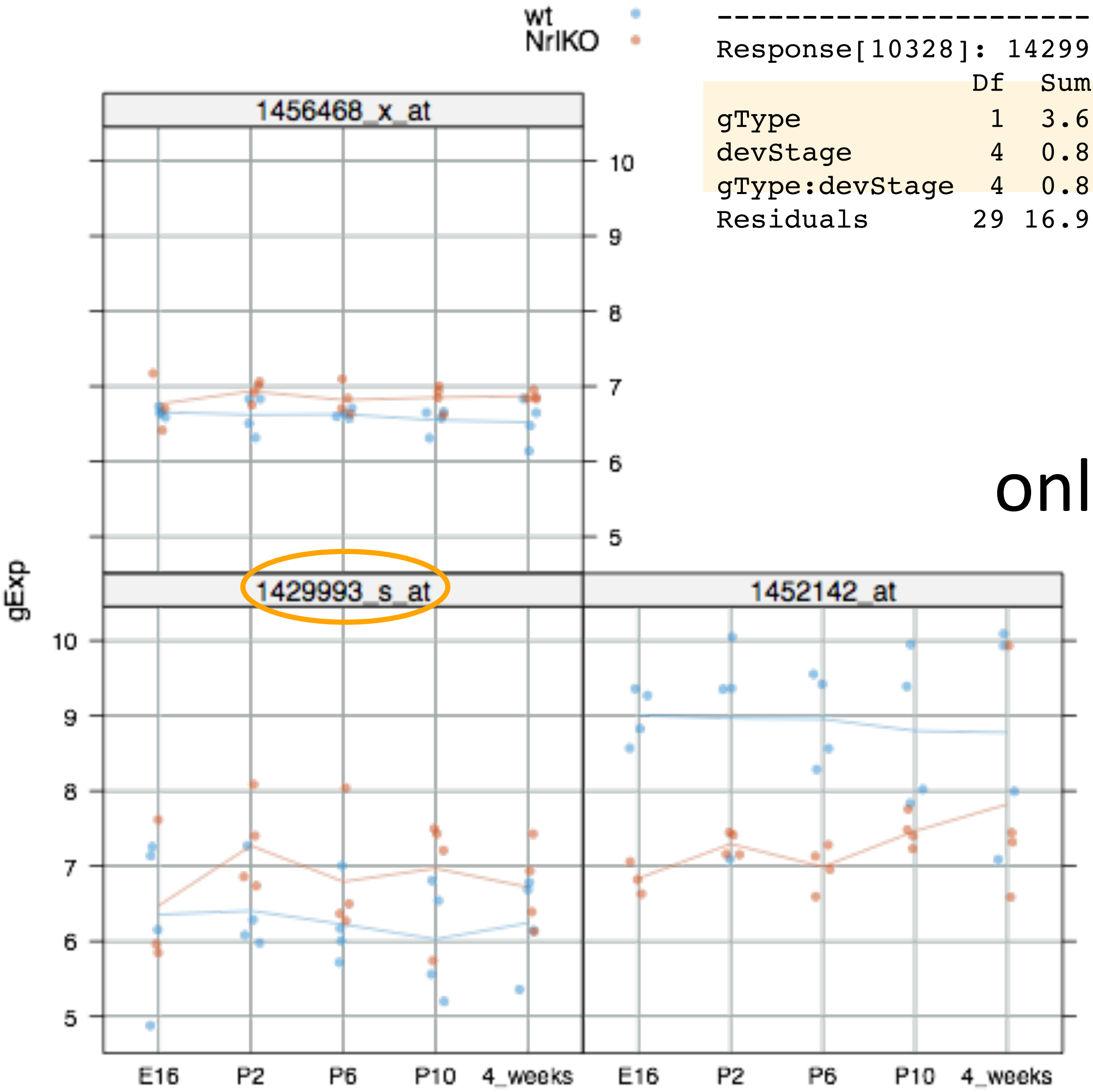
think about this:
no interaction
no knockout effect
YES developmental stage effects

```
Analysis of Variance Table

----------------------------
Response[26301]: 1455007_s_at
               Df Sum Sq Mean Sq F value    Pr(>F)    
gType           1 0.3209 0.32092  2.1120    0.1569    
devStage        4 7.7431 1.93578 12.7394 4.204e-06 ***
gType:devStage  4 0.1927 0.04818  0.3171    0.8642    
Residuals      29 4.4066 0.15195                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



only devStage matters

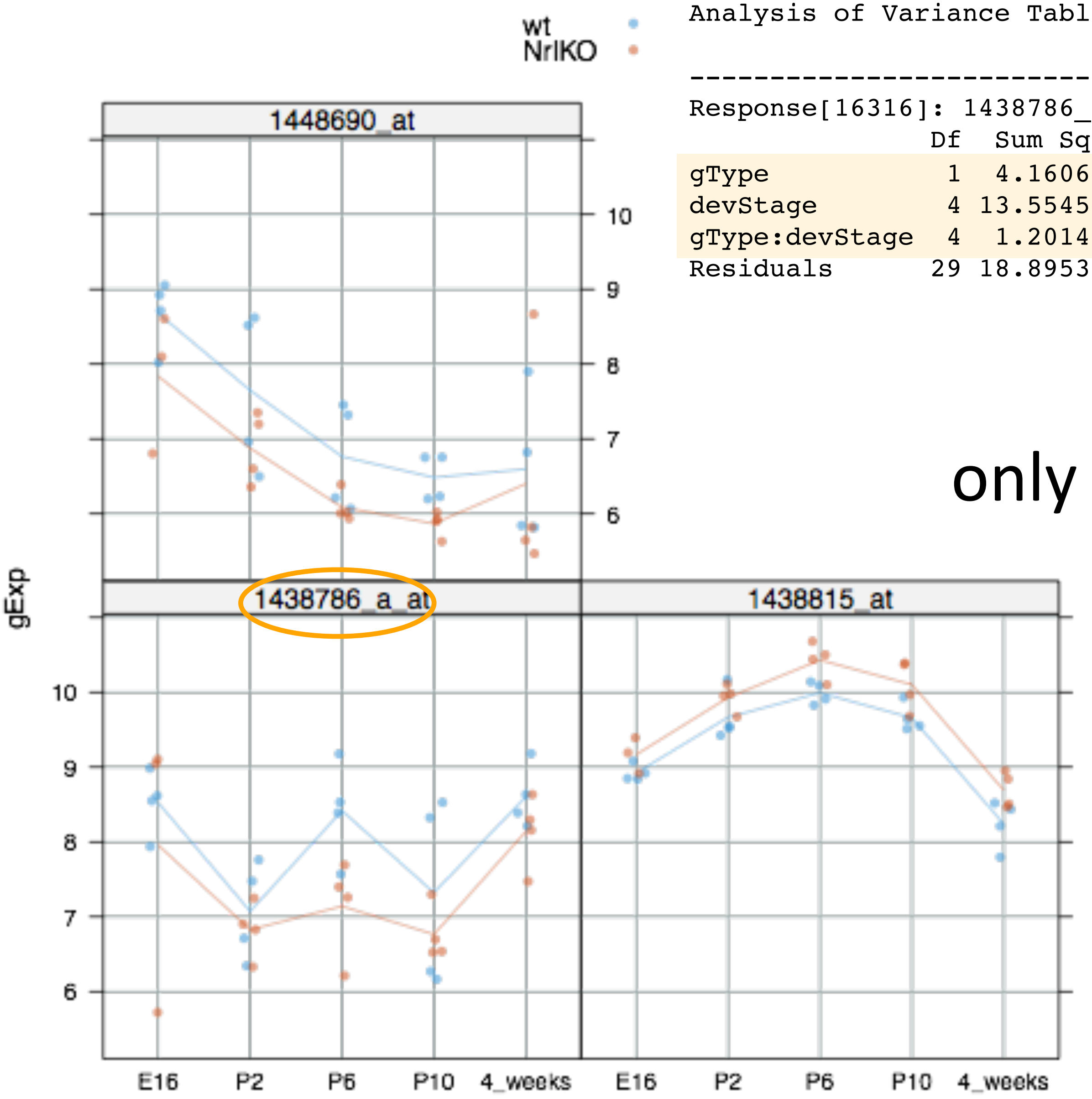
think about this:
no interaction
YES knockout effect
no developmental stage effects



```
Analysis of Variance Table

-----------------------------
Response[10328]: 1429993_s_at
               Df  Sum Sq Mean Sq F value  Pr(>F)
gType           1  3.6819  3.6819  6.3094 0.01783 *
devStage        4  0.8028  0.2007  0.3439 0.84603
gType:devStage  4  0.8034  0.2008  0.3442 0.84586
Residuals      29 16.9231  0.5836
```

only gType matters

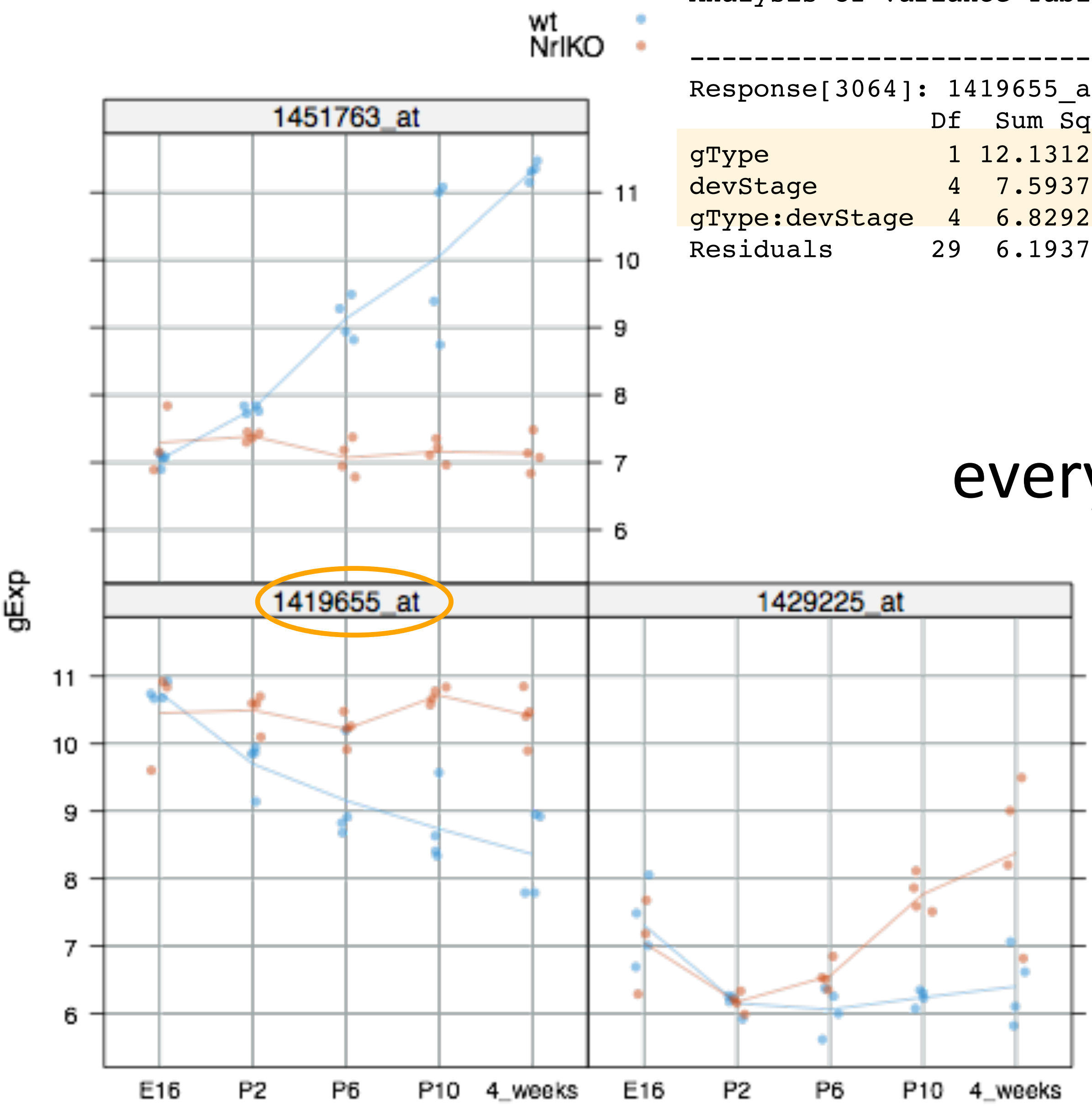
think about this:
no interaction
YES knockout effect
YES developmental stage effects



```
Analysis of Variance Table

------------------------------
Response[16316]: 1438786_a_at
                Df  Sum Sq Mean Sq F value   Pr(>F)
gType            1  4.1606  4.1606  6.3855 0.017216 *
devStage         4 13.5545  3.3886  5.2008 0.002774 **
gType:devStage   4  1.2014  0.3003  0.4610 0.763712
Residuals       29 18.8953  0.6516
```

only main effects

think about this:
YES interaction
YES knockout effect
YES developmental stage effects



```
Analysis of Variance Table

----------------------------
Response[3064]: 1419655_at
                Df  Sum Sq Mean Sq F value    Pr(>F)
gType            1 12.1312 12.1312 56.8008 2.623e-08 ***
devStage         4  7.5937  1.8984  8.8888 8.210e-05 ***
gType:devStage   4  6.8292  1.7073  7.9939 0.0001798 ***
Residuals       29  6.1937  0.2136
```

everything's going on

beginning to see the awkwardness of having a categorical variable with many levels (devStage)?

much nicer to have a quantitative variable and treat it that way!

let's make a quantitative version of devStage

first, let's discuss in abstract ... then we'll do for real

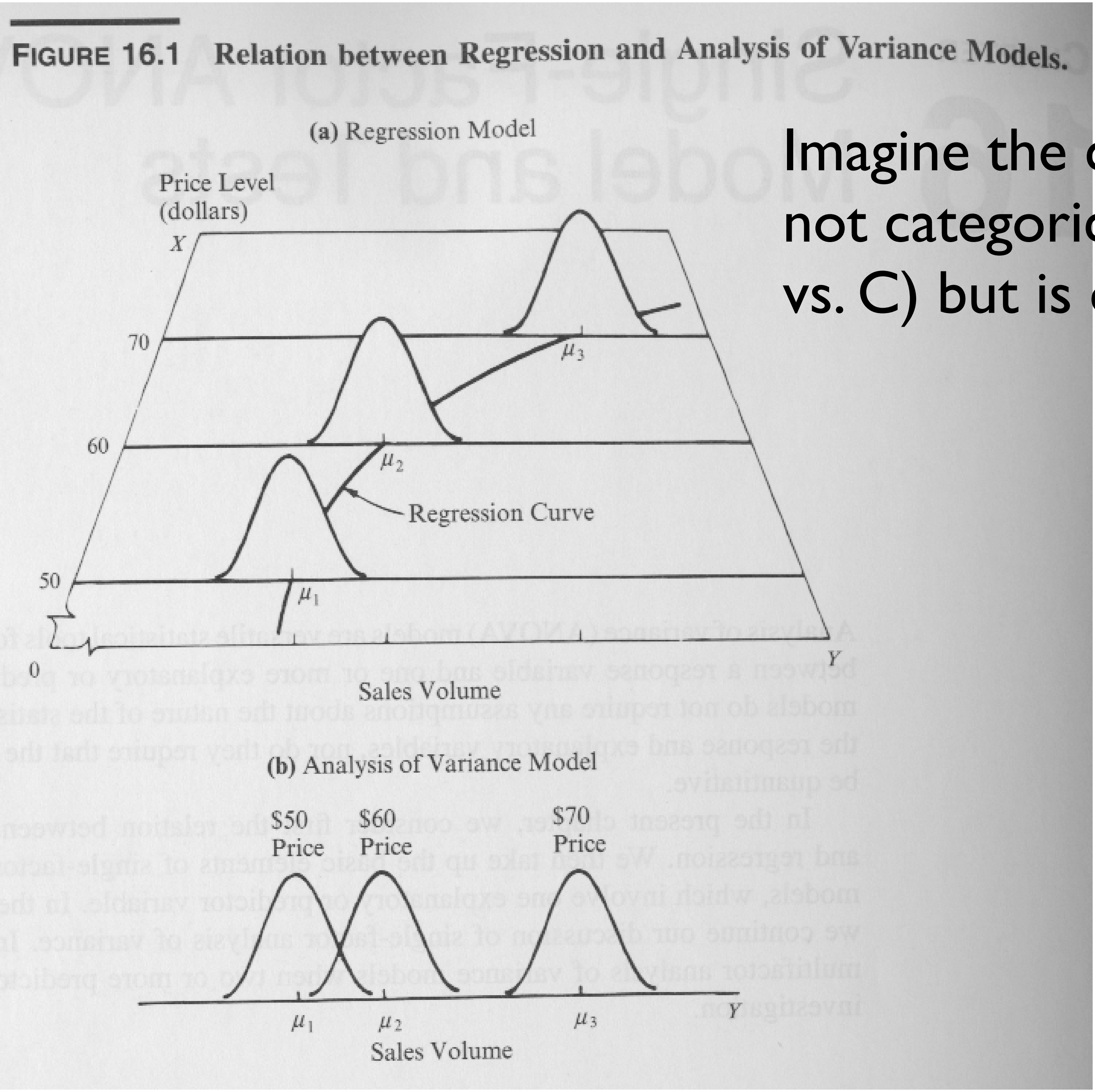
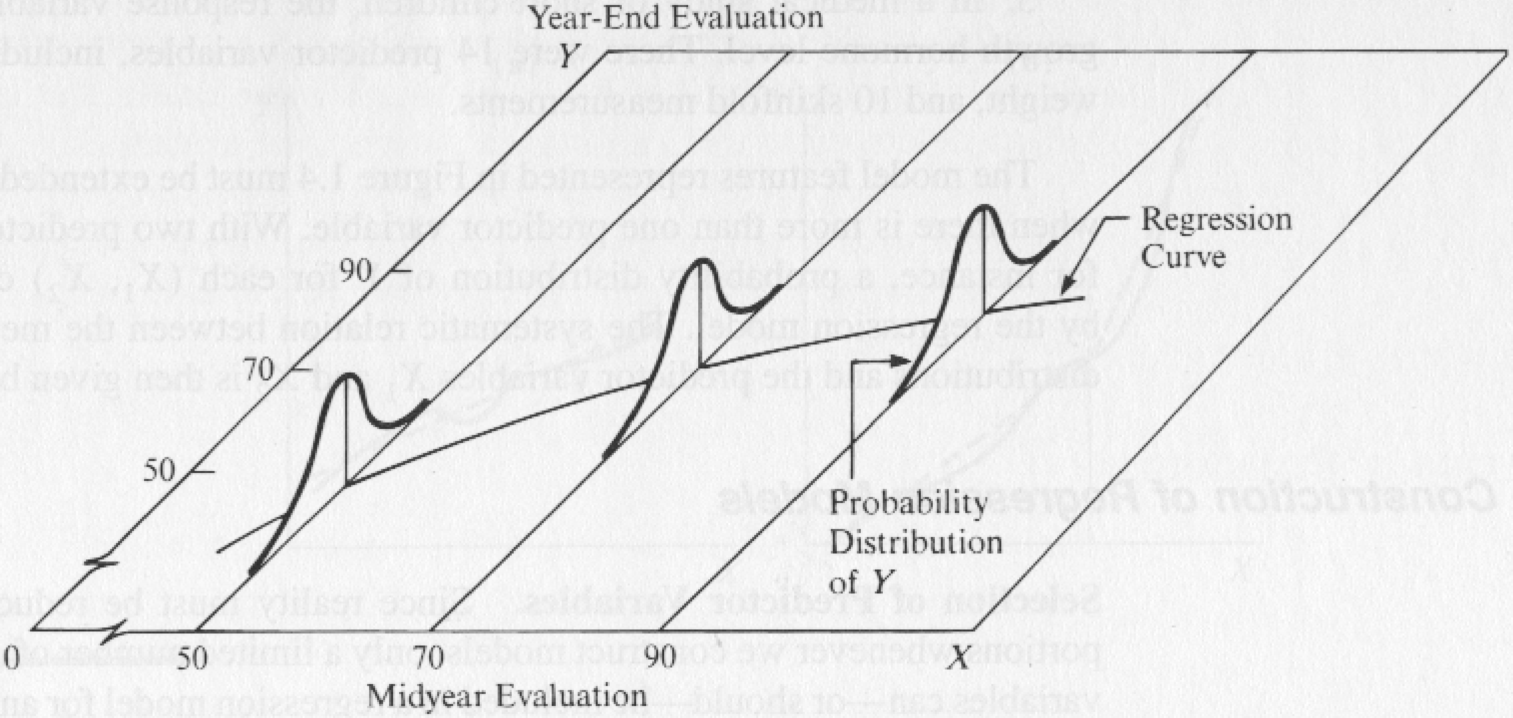


FIGURE 16.1 Relation between Regression and Analysis of Variance Models.

Imagine the covariate is not categorical (A vs. B vs. C) but is quantitative

FIGURE 1.4 Pictorial Representation of Regression Model.

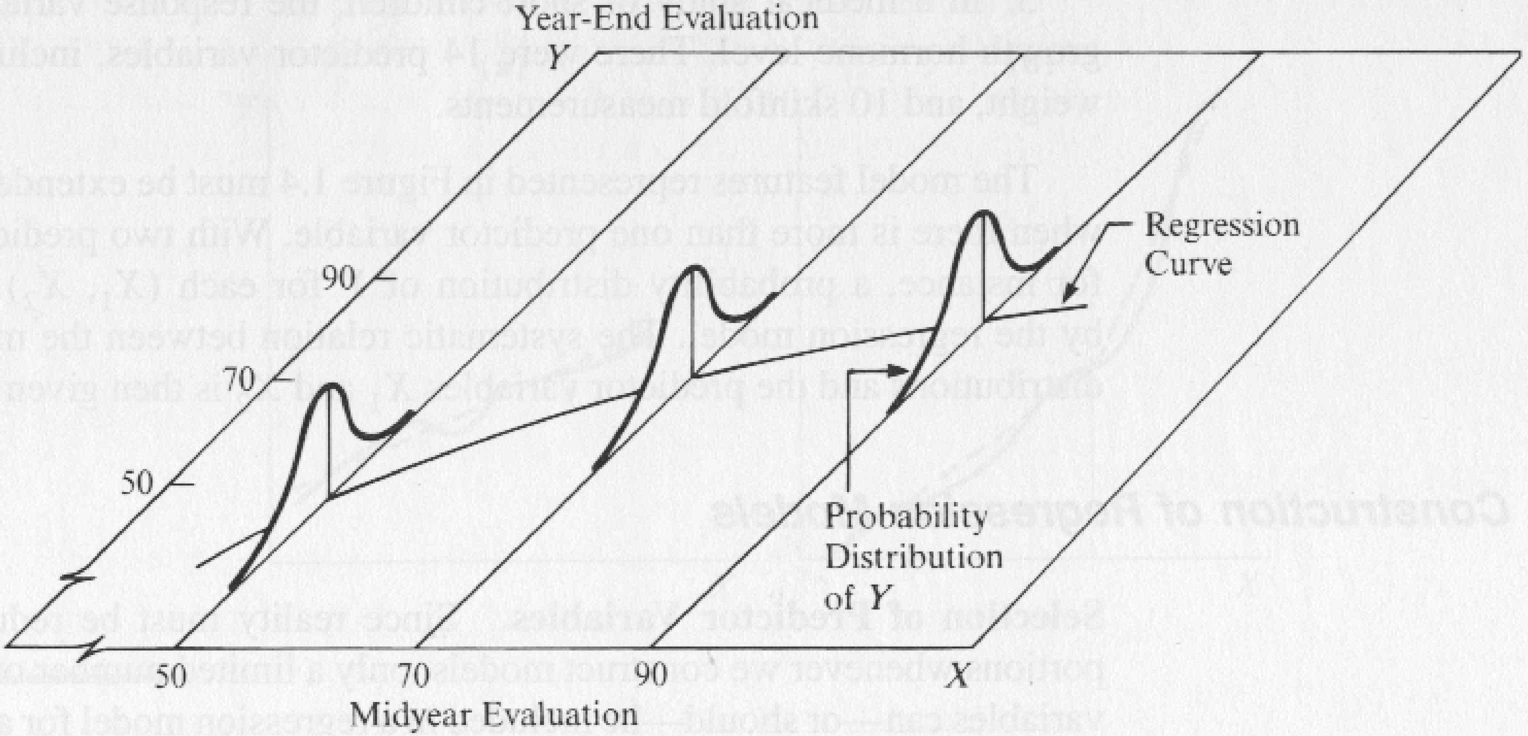


Rotated 90° rel to previous figure,to reflect how we usually view.

Covariate X on horizontal axis.

Response Y on vertical axis.

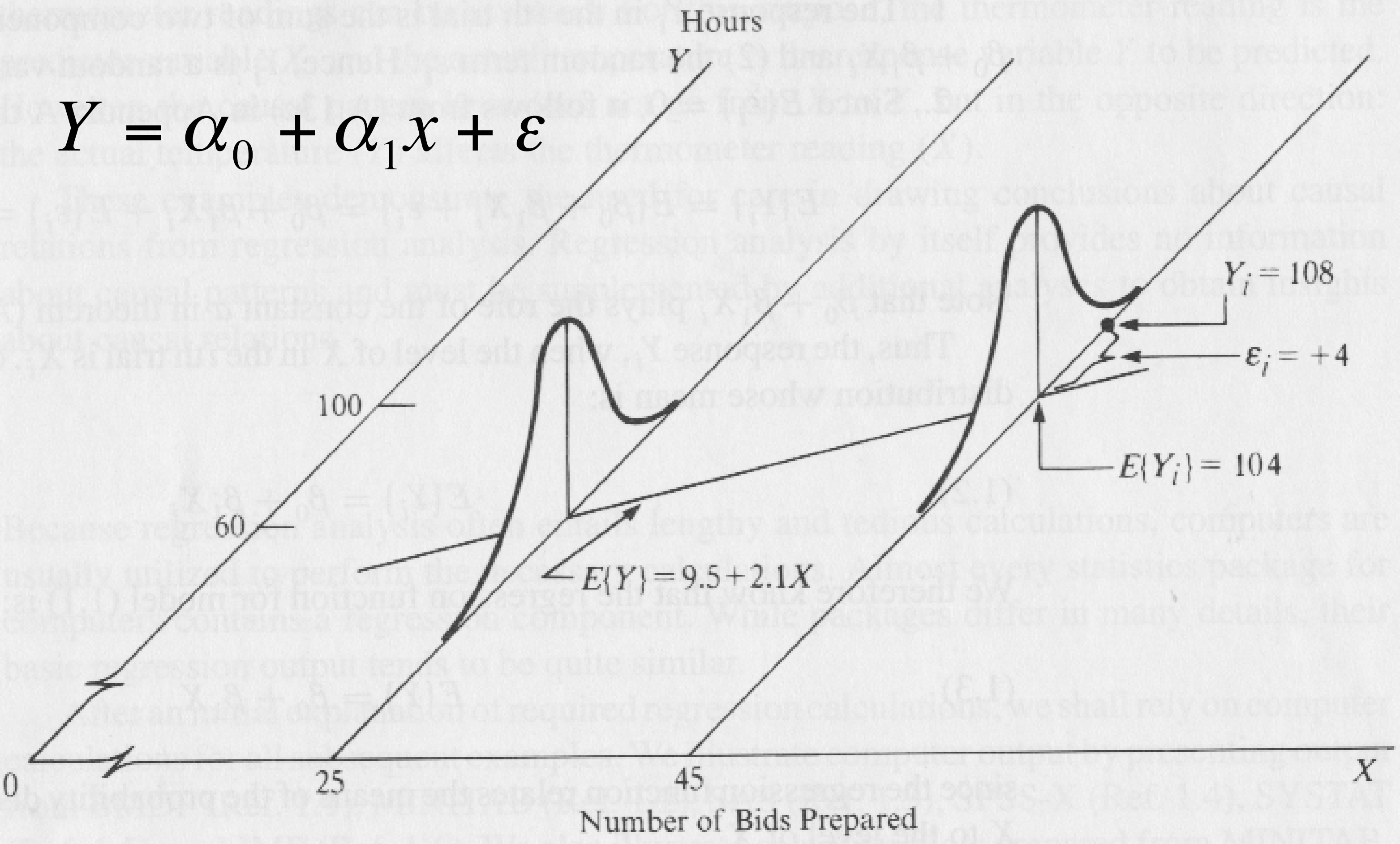
FIGURE 1.4 Pictorial Representation of Regression Model.



$$Y_{X=x} = f(x;\alpha) + \varepsilon_x, E(\varepsilon_x) = 0$$

FIGURE 1.6 Illustration of Simple Linear Regression Model (1.1).

$$Y = \alpha_0 + \alpha_1 x + \varepsilon$$



Regression function is *linear* … *linear model*.

Some regression models, in decreasing generality:

Nonparametric regression (smoothers like loess, splines)

Nonlinear (parametric) regression

Linear model, (multiple) linear regression
- if a mix of categorical and quantitative covariates, sometimes called analysis of covariance (ANCOVA)

Analysis of variance (ANOVA), i.e. linear model with a covariates categorical

Two-sample t test

inference in linear models

Plain vanilla linear model, matrix formulation

$$Y = X\alpha + \varepsilon$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \\ y_n \end{bmatrix} \qquad X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ \\ 1 & x_n \end{bmatrix} \qquad \alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \qquad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \\ \varepsilon_n \end{bmatrix}$$

Here’s what a design matrix would look like with 1 quantitative covariate.

$$Y = X\alpha + \varepsilon$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} = \begin{bmatrix} \alpha_0 \cdot 1 + \alpha_1 \cdot x_1 \\ \alpha_0 \cdot 1 + \alpha_1 \cdot x_2 \\ \vdots \\ \alpha_0 \cdot 1 + \alpha_1 \cdot x_n \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} = \begin{bmatrix} \alpha_0 + \alpha_1 x_1 + \varepsilon_1 \\ \alpha_0 + \alpha_1 x_2 + \varepsilon_2 \\ \vdots \\ \alpha_0 + \alpha_1 x_n + \varepsilon_n \end{bmatrix}$$

$$y_i = \alpha_0 + \alpha_1 x_i + \varepsilon_i$$

Here we are just fitting a line but using matrix notation to handle all *n* observations at once, more elegantly.

Big pay-offs ensue .....

how to estimate the alphas?

the story is the same whether you have quantitative and/or categorical covariate(s)

$$Y = X\alpha + \varepsilon$$

Estimation of the parameter α

Two viewpoints:

- maximum likelihood estimation, assuming ε_i are iid $N(0, \sigma^2)$
- "ordinary least squares" (OLS), i.e. minimizing the sum of the squared residuals

both lead to the same estimator of α:

$$\hat{\alpha} = (X^T X)^{-1} X^T y = \min^{-1} \sum (y_i - x_i \alpha)^2$$

Estimation of the parameter α

$$\hat{\alpha} = (X^T X)^{-1} X^T y = \min^{-1} \sum (y_i - x_i \alpha)^2$$

How one might derive this ...

- linear algebra: fitted value $X\hat{\alpha}$ must be the projection of the observed data vector Y onto the space spanned by the columns of X
- calculus: take the sum of squared residuals and minimize it, i.e. take first derivative(s), set equal to zero, and solve for $\hat{\alpha}$

Greatest Hits of Regression Results (normal iid errors)

$Y = X\alpha + \varepsilon$ regression model

$\hat{\alpha} = (X^T X)^{-1} X^T Y$ the MLE and OLS estimator of α

$\hat{Y} = X\hat{\alpha}$ the fitted or predicted values

$\hat{Y} = X(X^T X)^{-1} X^T Y = HY$ where $H = X(X^T X)^{-1} X^T$ is called the "hat matrix"

$\hat{\varepsilon} = Y - \hat{Y} = Y - X\hat{\alpha}$ the residuals (note NOT the same as the errors ε)

$\hat{\sigma}^2 = \dfrac{1}{n-p}\hat{\varepsilon}^T\hat{\varepsilon}$ the estimated error variance (p is the dimension of α)

$\hat{V}(\hat{\alpha}) = \hat{\sigma}^2 (X^T X)^{-1}$ the estimated covariance matrix of $\hat{\alpha}$

estimated standard errors for the estimated regression coefficients -- $\widehat{se}(\hat{\alpha}_j)$ --

are obtained by taking the square root of the diagonal elements of $\hat{V}(\hat{\alpha})$

Inference in Regression (normal iid errors)

$Y = X\alpha + \varepsilon$ regression model

$\hat{\alpha} = (X^T X)^{-1} X^T Y$ the MLE and OLS estimator of α

$\hat{\sigma}^2 = \frac{1}{n-p} \hat{\varepsilon}^T \hat{\varepsilon}$ the estimated error variance

$\hat{V}(\hat{\alpha}) = \hat{\sigma}^2 (X^T X)^{-1}$ the estimated covariance matrix of $\hat{\alpha}$

How test $H_0 : \alpha_j = 0$?

With a t-statistic. Under H_0, we have (at least approximately) that:

$$\frac{\hat{\alpha}_j}{\widehat{se}(\hat{\alpha}_j)} \sim t_{n-p}$$

so a p-value is obtained by computing a tail probability for the observed value of $\hat{\alpha}_j$ from a t_{n-p} distribution.

How to do inference on contrasts? (still assuming normal iid errors, one gene-at-a-time model)

$Y = X\alpha + \varepsilon$ regression model

$\hat{\alpha} = (X^T X)^{-1} X^T Y$

$\hat{\sigma}^2 = \frac{1}{n-p} \hat{\varepsilon}^T \hat{\varepsilon}$ the estimated error variance

$\hat{V}(\hat{\alpha}) = \hat{\sigma}^2 (X^T X)^{-1}$ the estimated covariance matrix of $\hat{\alpha}$

Consider the contrasts of interest:

$\mathrm{C}^T \alpha = \beta \quad \Rightarrow \quad \mathrm{C}^T \hat{\alpha} = \hat{\beta}$

Using results not developed in this class, we have that:

$\hat{V}(\hat{\beta}) = \mathrm{C}^T \hat{V}(\hat{\alpha}) C = \hat{\sigma}^2 \mathrm{C}^T (X^T X)^{-1} C$ is the estimated covariance matrix of $\hat{\beta}$

How to do inference on contrasts?

How test $H_0 : \beta_j = 0$?

With a t-statistic. Under H_0, we have (at least approximately) that:

$$\frac{\hat{\beta}_j}{\widehat{se}(\hat{\beta}_j)} \sim t_{n-p}$$

so a p-value is obtained by computing a tail probability for the observed value of $\hat{\beta}_j$ from a t_{n-p} distribution.