Statistical Methods for High Dimensional Biology

STAT/BIOF/GSAT 540

Lecture 6 – Anova & Linear models

Sara Mostafavi

January 22 2018

Slide credits: Drs. Jenny Bryan and Su-In Lee

Announcements

- Project group size and composition
- Website and lectures

Review of the data in hand:
Developing mouse retina – time course for the experiment

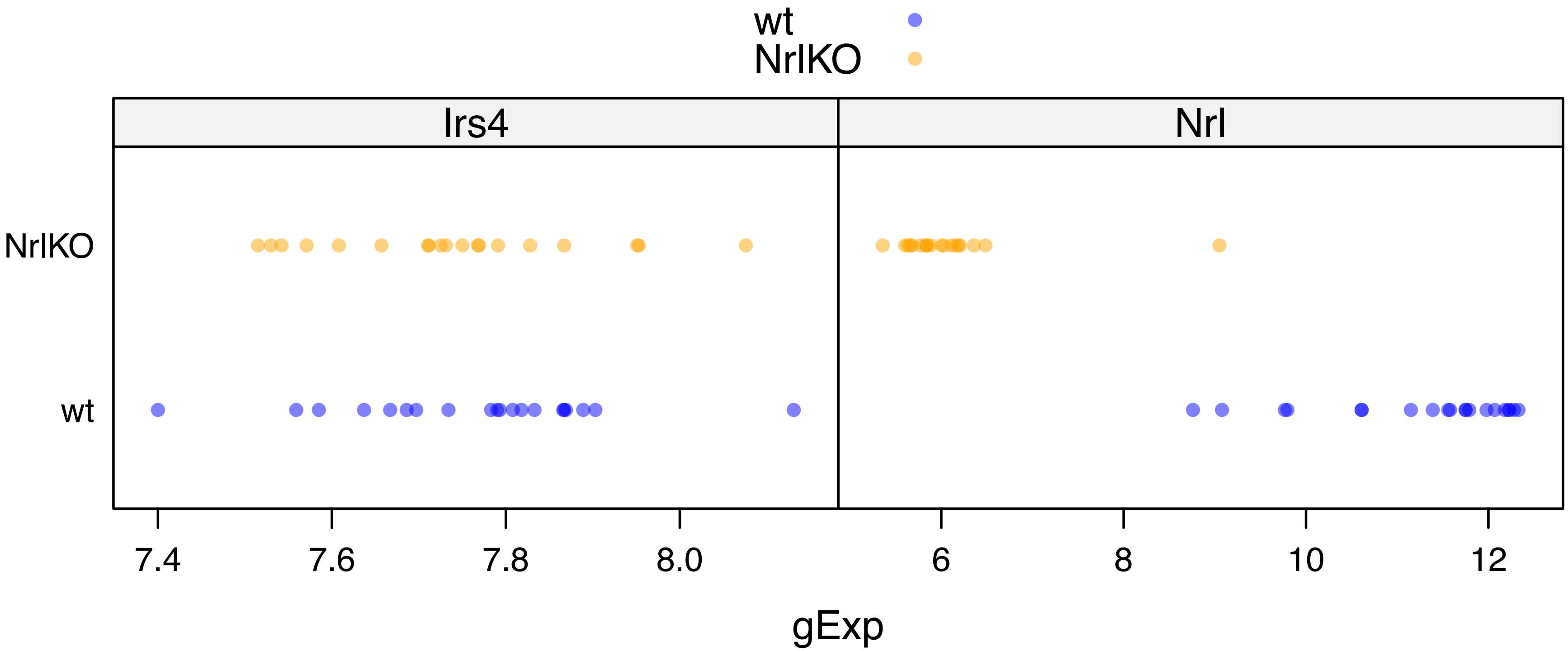
So sample collections:

4 developmental stages
2 genotypes: wild-type , Nrl KO
3-4 replicates for each combination

Experimental design

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

Do we think the orange's and blue's are generated by different underlying distributions?



Irs4 (insulin receptor substrate 4) was selected at random as a boring non differentially expressed gene; NrlKO ~= wt

Nrl (neural retina leucine zipper gene) is the gene that was knocked out in half the mice; obviously should be differentially expressed; NrlKO << wt

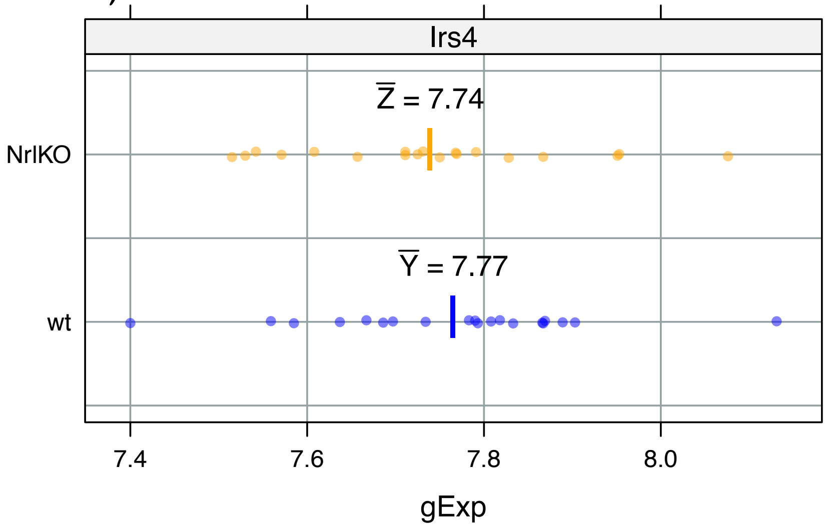
Comparing the mean of two groups

- T-test: special case of ANOVA, where the only difference is that with ANOVA you can compare more than two groups.

- ANOVA: special case of linear regression/ model, where the only difference is with linear models you can consider quantitative and categorical variables.

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

two sample t test



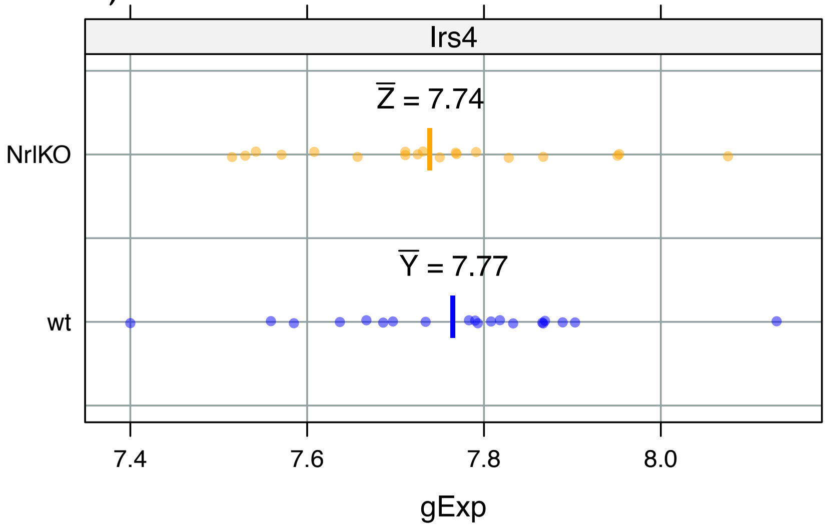
```
> summary(aov(gExp ~ gType, miniDat,
+              subset = gene == "Irs4"))
```

(one-way) analysis of variance
“ANOVA”

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

linear model
linear regression

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

    Two Sample t-test

data:  gExp by gType
t = 0.5286, df = 37, p-value = 0.6002
<snip, snip>
sample estimates:
   mean in group wt mean in group NrlKO
           7.765750            7.739684
```



```
> summary(aov(gExp ~ gType, miniDat,
+             subset = gene == "Irs4"))
            Df Sum Sq Mean Sq F value Pr(>F)
gType        1 0.0066 0.00662   0.279    0.6
Residuals   37 0.8764 0.02369
```

```
7.739684 - 7.765750 = -0.026066
```

```
-0.5286494 ^ 2 = 0.2794702
```

```
> summary(lm(gExp ~ gType, miniDat,
+            subset = gene == "Irs4"))
<snip, snip>
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.76575    0.03441 225.650   <2e-16 ***
gTypeNrlKO  -0.02607    0.04931  -0.529      0.6
<snip, snip>
F-statistic: 0.2795 on 1 and 37 DF,  p-value: 0.6002
```

These are not coincidences!

Linear regression

- Change of notation to be consistent with conventions used in linear regression

$$Y \sim F$$

$$Y = \mu + \varepsilon, \text{ where } \varepsilon \sim F, E(\varepsilon) = 0$$

- We're going to follow statistical convention for regression and use Y for a variable we observe and regard as a response (like before) and X will be associated with the variables we regard as predictors or explanatory variables, e.g. the distinction between wild type and knockouts.

- Generic problem: given a collection of variables we want to know whether the response/outcome variable Y depends on other factors $X_1,\ldots,X_n$
- Statistical model: defines a mathematical relationship between Y anx $X_1,\ldots,X_n$. The model “predicts” Y from X_i

Imagine we are studying the response Y (e.g., gene expression) in two or more groups, denoted by *j*:

$$Y_{ij} = \mu_j + \varepsilon_{ij}, \text{ where } \varepsilon_{ij} \sim F, E(\varepsilon_{ij}) = 0$$

Note how we allow for different expected values of Y for each treatment :

$$E(Y_{ij}) = \mu_{ij}$$

We assume that the noise has a common distribution across the groups.

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 …)

$$Y_{ij} = \mu_j + \varepsilon_{ij}, \text{ where } \varepsilon_{ij} \sim F, E(\varepsilon_{ij}) = 0$$

$$\begin{bmatrix} Y_{11} \\ \vdots \\ Y_{n_1 1} \\ Y_{12} \\ \vdots \\ Y_{n_2 2} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_1 \\ \mu_2 \\ \vdots \\ \mu_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{n_1 1} \\ \varepsilon_{12} \\ \vdots \\ \varepsilon_{n_2 2} \end{bmatrix}$$

Whenever the Y_{ij} is from group 1, I put in μ_1, and when Y_{ij} is from group 2, I put in μ_2.

$$Y_{ij} = \mu_j + \varepsilon_{ij}, \text{ where } \varepsilon_{ij} \sim F, E(\varepsilon_{ij}) = 0$$

$$\underset{\substack{\uparrow \\ \mathbf{Y}}}{\begin{bmatrix} Y_{11} \\ \vdots \\ Y_{n_1 1} \\ Y_{12} \\ \vdots \\ Y_{n_2 2} \end{bmatrix}} = \underset{\substack{\uparrow \\ \mathbf{X}}}{\begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix}} \underset{\substack{\uparrow \\ \boldsymbol{\alpha}}}{\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}} + \begin{bmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{n_1 1} \\ \varepsilon_{12} \\ \vdots \\ \varepsilon_{n_2 2} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_1 \\ \mu_2 \\ \vdots \\ \mu_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{n_1 1} \\ \varepsilon_{12} \\ \vdots \\ \varepsilon_{n_2 2} \end{bmatrix}$$

For statistical and computational reasons, easier to work with matrix formulation of the problem. X is called the design matrix (“feature matrix” in CS/ML)

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

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

The exact form of the design matrix X and the parameter alpha are not uniquely defined. The user has some control. The two objects are tightly related to each other. This will become much more clear in examples.

How do we do hypothesis testing with linear regression?

- Recall that for comparing two groups, we'd like to know

$$\begin{tikzcd}
\mu_1 = \mu_2 \arrow[r, phantom, "\Leftrightarrow"] & \mu_1 - \mu_2 = 0 \arrow[d] \\
\mu_1 - \mu_2 = \tau_2 & \tau_2 = 0
\end{tikzcd}$$

TOTALLY EQUIVALENT!

ANOVA-style, "cell means"

$$Y_{ij} = \mu_j + \varepsilon_{ij}$$

ANOVA-style, "ref + tx effects"

$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij}, (\tau_1 = 0)$$

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

$$\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \\ y_{n_2 2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \\ \varepsilon_{n_2 2} \end{bmatrix}$$

$$\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \\ y_{n_2 2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \tau_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \\ \varepsilon_{n_1 2} \end{bmatrix}$$

The design matrix specifies how the observed data relates to the regression parameters.

Cell-mean notation **“ref + treatment” notation**

$$y_{i,1} = \mu_1 \quad \Leftrightarrow \quad y_{i,1} = \theta$$

$$y_{i,2} = \mu_2 \quad \Leftrightarrow \quad y_{i,2} = \theta + \tau_2$$

$$\Big\downarrow$$

$$\mu_1 - \mu_2 = \theta - \theta + \tau_2 = \tau_2$$

How do we do hypothesis testing with linear regression?

- Recall that for comparing two groups, we'd like to know

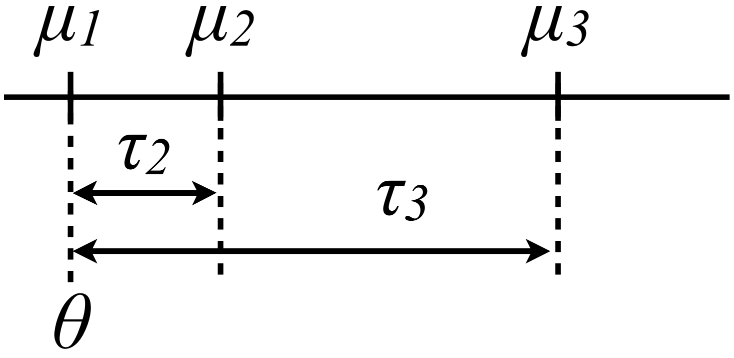
$$\mu_1 = \mu_2 \Leftrightarrow \mu_1 - \mu_2 = 0 \Leftrightarrow \tau_2 = 0$$

- With more than two groups, what would we like to test??

Note we can obtain one set of parameters from the others!

ANOVA-style, "cell means"

$$Y_{ij} = \mu_j + \varepsilon_{ij}$$



$$\begin{aligned} \mu_1 &= \theta \\ \mu_2 &= \theta + \tau_2 \\ \mu_3 &= \theta + \tau_3 \end{aligned} \qquad \begin{aligned} \theta &= \mu_1 \\ \tau_2 &= \mu_2 - \mu_1 \\ \tau_3 &= \mu_3 - \mu_1 \end{aligned}$$

$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij}, (\tau_1 = 0)$$

ANOVA-style, "ref + tx effects"

Let’s assume we have **three** groups

ANOVA-style, “cell means”

$$Y_{ij} = \mu_j + \varepsilon_{ij}$$

ANOVA-style, “ref + tx effects”

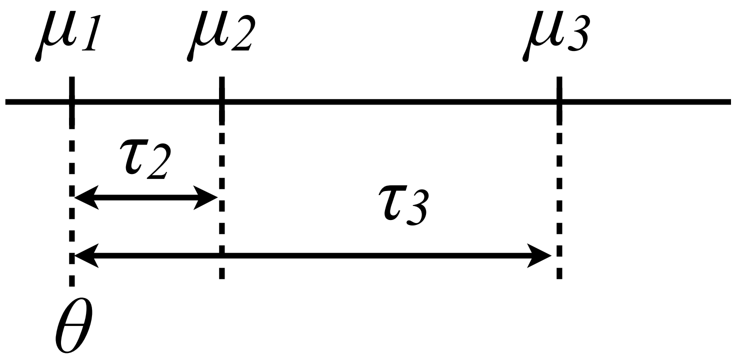
$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij}, (\tau_1 = 0)$$

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

$$\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \\ y_{n_3 3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \\ \varepsilon_{n_3 3} \end{bmatrix} \qquad \begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \\ y_{n_3 3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \\ \varepsilon_{n_3 3} \end{bmatrix}$$

The design matrix specifies how the observed data relates to the regression parameters.

We can do this neatly with matrix multiplication! The matrices C below are sometimes called "contrast matrices".



ANOVA-style, "cell means"

$$Y_{ij} = \mu_j + \varepsilon_{ij}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

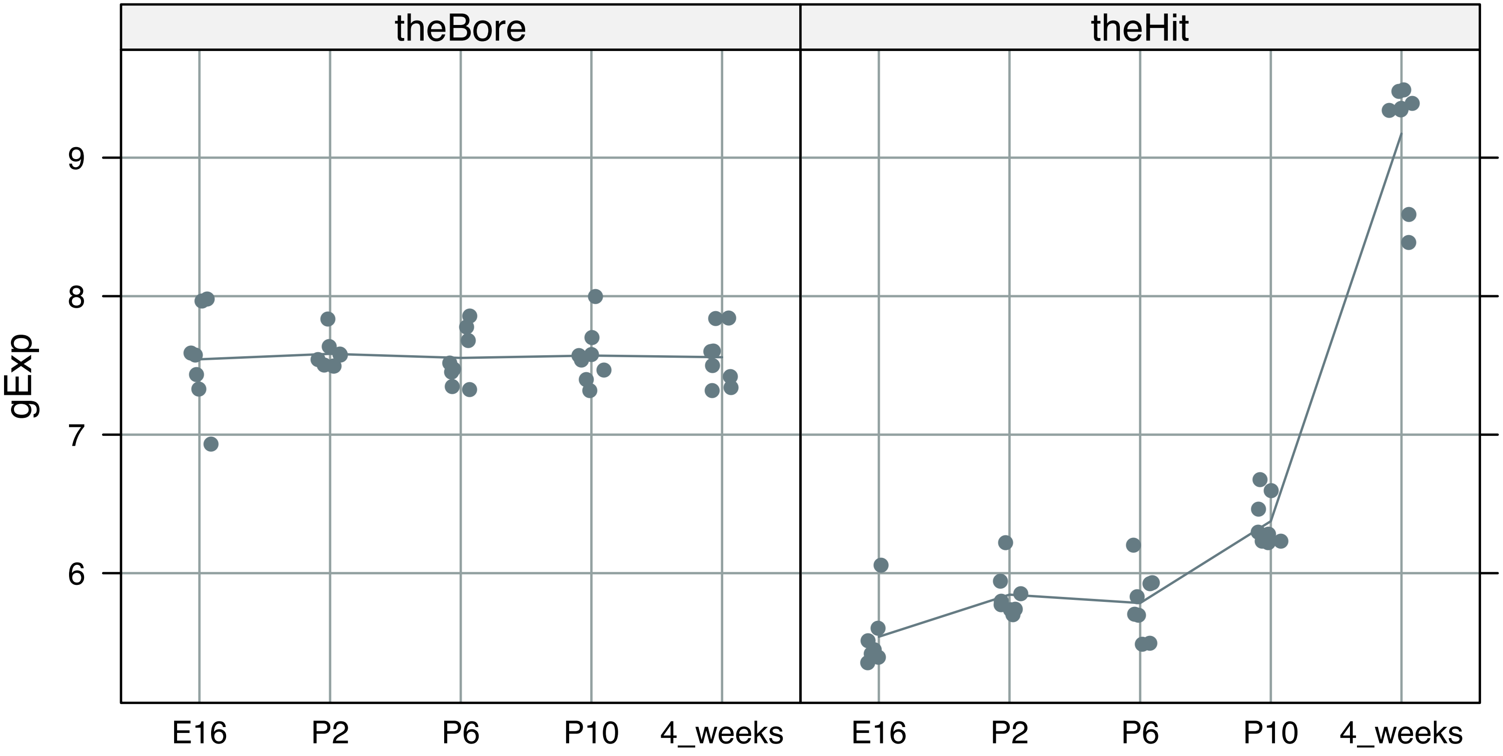
$$C^T \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \end{bmatrix} = \mu \qquad C^T \mu = \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij}, (\tau_1 = 0)$$

ANOVA-style, "ref + tx effects"

Let's look at some data: do you think devStage has an effect on gene expression?

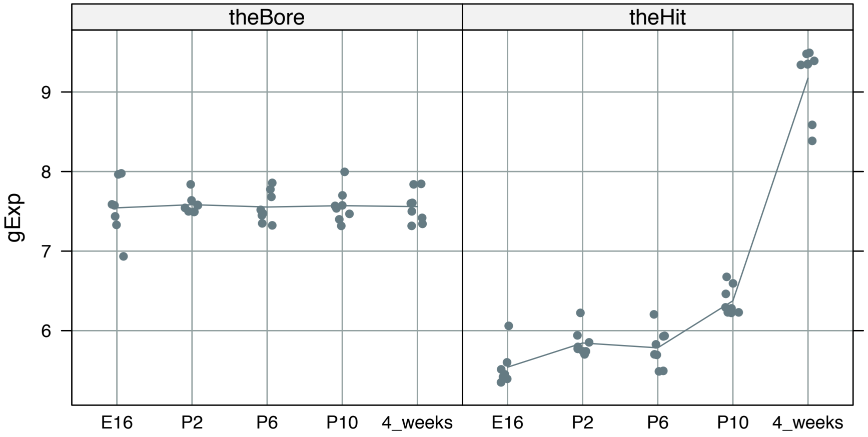
(side question: do you feel uncomfortable with how I asked the question?)



What’s our null hypothesis ?

E16 P2 P6 P10 4_weeks

$$\mu_{E16} = \mu_{P2} = \mu_{P6} = \mu_{P10} = \mu_{4weeks}$$

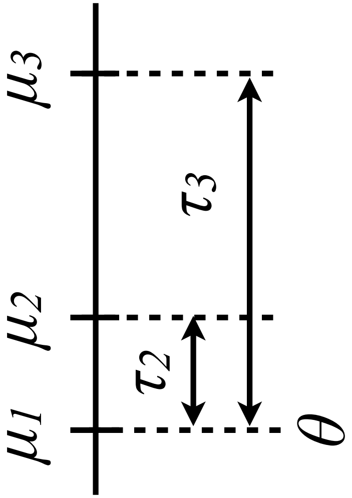


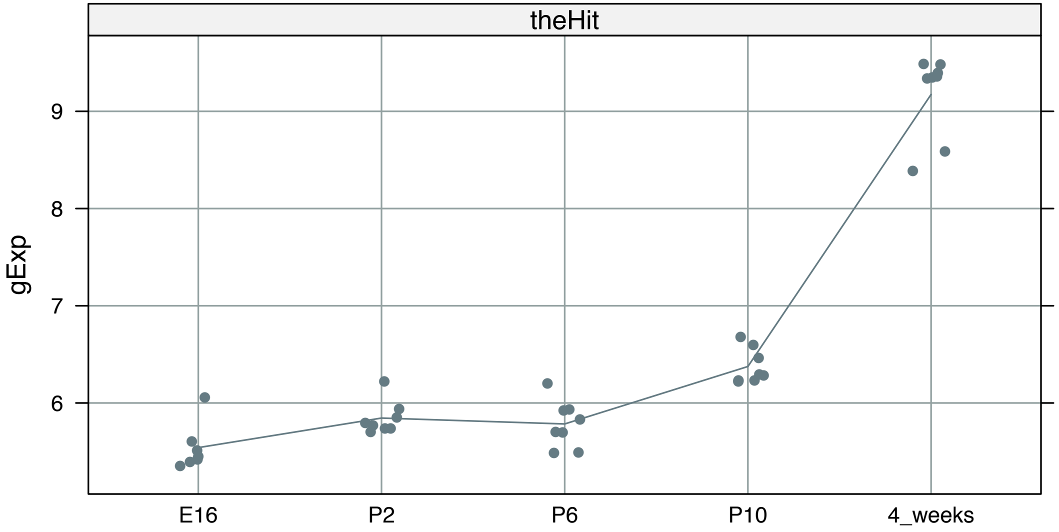
```
> with(miniDat,
+      tapply(gExp, list(devStage, gene), mean))
         theBore   theHit
E16     7.544143 5.540857
P2      7.583500 5.844875
P6      7.554000 5.784250
P10     7.571000 6.375125
4_weeks 7.559000 9.173375
```

```
> data.frame(cellMeans = theHitAvgs,
+            txEffects = theHitAvgs - theHitAvgs[1])
         cellMeans txEffects
E16       5.540857 0.0000000
P2        5.844875 0.3040179
P6        5.784250 0.2433929
P10       6.375125 0.8342679
4_weeks   9.173375 3.6325179
```

the mu's = "cell means"
.... estimated by sample avg @ each devStage

(theta, the tau's) = ref + tx effects
.... estimated by (E16 avg, other avgs - E16 avg)

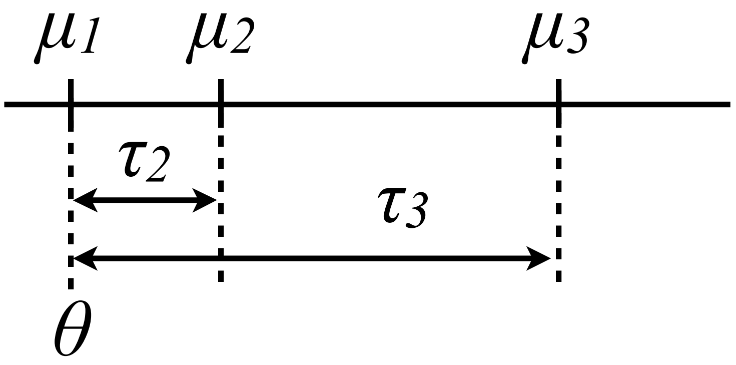




$$Y = X\alpha + \varepsilon$$

$$\alpha = (\theta, \tau_{P2}, \tau_{P6}, \tau_{P10}, \tau_{4\_weeks})$$

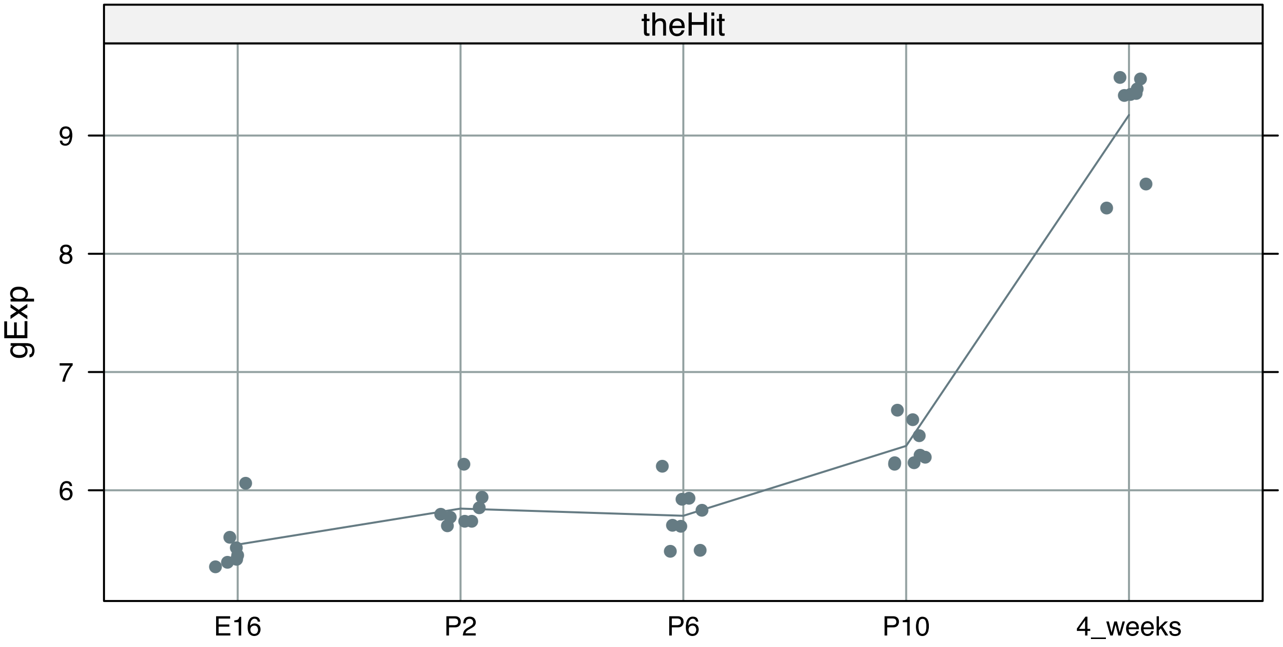
```
        cellMeans txEffects
E16      5.540857 0.0000000
P2       5.844875 0.3040179
P6       5.784250 0.2433929
P10      6.375125 0.8342679
4_weeks  9.173375 3.6325179
```



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

$$\alpha = (\theta, \tau_{P2}, \tau_{P6}, \tau_{P10}, \tau_{4\_weeks})$$

```
> hitFit <- lm(gExp ~ devStage, miniDat, gene == "theHit")

> summary(hitFit)$coef
                 Estimate Std. Error   t value     Pr(>|t|)
(Intercept)     5.5408571  0.1021381 54.248698 1.307554e-34
devStageP2      0.3040179  0.1398583  2.173756 3.678022e-02
devStageP6      0.2433929  0.1398583  1.740282 9.085489e-02
devStageP10     0.8342679  0.1398583  5.965093 9.559065e-07
devStage4_weeks 3.6325179  0.1398583 25.972843 5.266481e-24
```



```
> summary(hitFit)
Call:
lm(formula = gExp ~ devStage, <blah, blah>)
<snip, snip>
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       5.5409     0.1021  54.249  < 2e-16 ***
devStageP2        0.3040     0.1399   2.174   0.0368 *
devStageP6        0.2434     0.1399   1.740   0.0909 .
devStageP10       0.8343     0.1399   5.965 9.56e-07 ***
devStage4_weeks   3.6325     0.1399  25.973  < 2e-16 ***
---
<snip, snip>
F-statistic: 243.4 on 4 and 34 DF,  p-value: < 2.2e-16
```

what if we -- how would we -- force R to parametrize the model differently, e.g. using “cell means”?

```
> hitFitCellMeans <- lm(gExp ~ 0 + devStage, miniDat, gene == "theHit")

> summary(hitFitCellMeans)

Call:
lm(formula = gExp ~ 0 + devStage, <blah, blah>)

<snip, snip>

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
devStageE16       5.54086    0.10214   54.25   <2e-16 ***
devStageP2        5.84488    0.09554   61.18   <2e-16 ***
devStageP6        5.78425    0.09554   60.54   <2e-16 ***
devStageP10       6.37512    0.09554   66.73   <2e-16 ***
devStage4_weeks   9.17337    0.09554   96.02   <2e-16 ***
---
<snip, snip>
Residual standard error: 0.2702 on 34 degrees of freedom
F-statistic:  4804 on 5 and 34 DF,  p-value: < 2.2e-16
```

parameter estimates = estimated means for each devStage = sample averages

Yay for interpretability!

```
        theHitAvgs
E16       5.540857
P2        5.844875
P6        5.784250
P10       6.375125
4_weeks   9.173375
```

what if we -- how would we -- force R to parametrize the model differently, e.g. using “cell means”?

```
> hitFitCellMeans <- lm(gExp ~ 0 + devStage, miniDat, gene == "theHit")

> summary(hitFitCellMeans)

Call:
lm(formula = gExp ~ 0 + devStage, <blah, blah>)

<snip, snip>

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
devStageE16      5.54086    0.10214   54.25   <2e-16 ***
devStageP2       5.84488    0.09554   61.18   <2e-16 ***
devStageP6       5.78425    0.09554   60.54   <2e-16 ***
devStageP10      6.37512    0.09554   66.73   <2e-16 ***
devStage4_weeks  9.17337    0.09554   96.02   <2e-16 ***
---
<snip, snip>
Residual standard error: 0.2702 on 34 degrees of freedom
F-statistic:  4804 on 5 and 34 DF,  p-value: < 2.2e-16
```

BUT what null hypotheses do these p-values correspond to????

```
        theHitAvgs
E16       5.540857
P2        5.844875
P6        5.784250
P10       6.375125
4_weeks   9.173375
```

what if we -- how would we -- force R to parametrize the model differently, e.g. using “cell means”?

```
> hitFitCellMeans <- lm(gExp ~ 0 + devStage, miniDat, gene == "theHit")

> summary(hitFitCellMeans)

Call:
lm(formula = gExp ~ 0 + devStage, <blah, blah>)

<snip, snip>

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
devStageE16      5.54086    0.10214   54.25   <2e-16 ***
devStageP2       5.84488    0.09554   61.18   <2e-16 ***
devStageP6       5.78425    0.09554   60.54   <2e-16 ***
devStageP10      6.37512    0.09554   66.73   <2e-16 ***
devStage4_weeks  9.17337    0.09554   96.02   <2e-16 ***
---
<snip, snip>
Residual standard error: 0.2702 on 34 degrees of freedom
F-statistic:  4804 on 5 and 34 DF,  p-value: < 2.2e-16
```

These p-values are for these tests:

$$H_0 : \mu_j = 0$$

Probably not what you’re really interested in! Boo.

```
        theHitAvgs
E16       5.540857
P2        5.844875
P6        5.784250
P10       6.375125
4_weeks   9.173375
```

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

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"

F-test and overall significance of one or more covariates

- The t-stat in linear regression allows us to test simple hypotheses:

$$\mathrm{H}_0 : \tau_i = 0$$
$$\mathrm{H}_A : \tau_i \neq 0$$

- But when we have multiple covariates/factors, we often like to test more complex hypotheses:

$$\mathrm{H}_0 : \tau_2 = \tau_3 = \ldots = 0$$

AND statement

$$\mathrm{H}_A : \tau_i \neq 0 \quad \text{for *some* } i$$

OR statement

- F-test allows us to test such compound tests

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

$$\alpha = (\theta, \tau_{P2}, \tau_{P6}, \tau_{P10}, \tau_{4\_weeks})$$

$H_0 : \tau_j = 0$

vs

$H_0 : \tau_j \neq 0$

for each j individually

$H_0 : \tau_j = 0$ AND statement

vs

$H_0 : \tau_j \neq 0$ OR statement

for all j at the same time

```
> summary(hitFit)
Call:
lm(formula = gExp ~ devStage, <blah, blah>)
<snip, snip>
Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)       5.5409     0.1021  54.249  < 2e-16 ***
devStageP2        0.3040     0.1399   2.174   0.0368 *  
devStageP6        0.2434     0.1399   1.740   0.0909 .  
devStageP10       0.8343     0.1399   5.965 9.56e-07 ***
devStage4_weeks   3.6325     0.1399  25.973  < 2e-16 ***
---
<snip, snip>
F-statistic: 243.4 on 4 and 34 DF,  p-value: < 2.2e-16
```

Regression residuals

- The goal of any model is to explain (*fit*) the observed data. How well does the model achieve this aim?

Our linear regression model: $y_{ij} = \theta + \tau_j + \varepsilon_{ij}$

the response (dependent variable) is modeled by a *linear* function of independent variables (given by the design matrix)

The model residual tells us how "good" our model fits the data

Regression Error (in theory) $\varepsilon_{ij} = y_{ij} - \theta + \tau_j$

Regression Residual Error (in practice) $\hat{\varepsilon}_{ij} = y_{ij} - \hat{\theta} + \hat{\tau}_j$

Residual variance and the utility of the model

- The goal of any model is to explain (*fit*) the observed data. How well does the model achieve this aim?

Our model of the data

$$y_{ij} = \theta + \tau_j + \varepsilon_{ij}$$

How good does the model fit out data:

$$\sum_{ij}(\hat{y}_{ij}-y_{ij})^2 = \sum_{ij}(y_{ij}-(\hat{\theta}+\hat{\tau}_j))^2$$

Residuals Sum of Squares

Small (restricted) model $$y_{ij} = \theta + \varepsilon_{ij} \quad \text{for all } i, j$$

Big (unrestricted) model $$y_{ij} = \theta + \tau_j + \varepsilon_{ij}$$

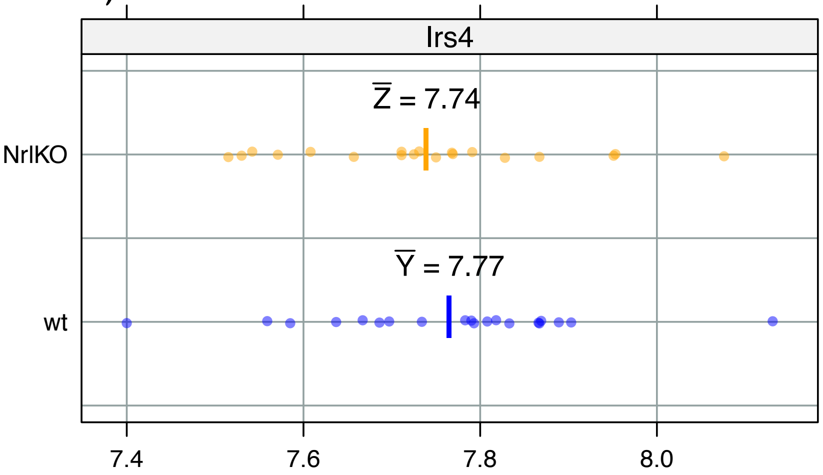
Crucial question: is the residual sum of squares (i.e., error) for restricted model (RSS_r) substantially larger than residual sum of squares for the full model (RSS_F)?

Test statistics for **F-test**:

$$F = \frac{(RSS_r - RSS_f)/(p_f - p_r)}{RSS_f/(n - p_f)} \qquad \sim F_{(p_{f-}p_f,n-p_f)}$$

Due to RA Fisher, F statistic follows an F distribution with degrees of freedom p_{f-r}, n-df_f

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

    Two Sample t-test

data:  gExp by gType
t = 0.5286, df = 37, p-value = 0.6002
<snip, snip>
sample estimates:
   mean in group wt mean in group NrlKO
           7.765750            7.739684
```



Equivalence between t-stat (squared) and F-stat when we only have 2 groups

```
-0.5286494 ^ 2 = 0.2794702
```

```
> summary(lm(gExp ~ gType, miniDat,
+            subset = gene == "Irs4"))
<snip, snip>
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.76575    0.03441 225.650   <2e-16 ***
gTypeNrlKO  -0.02607    0.04931  -0.529      0.6
<snip, snip>
F-statistic: 0.2795 on 1 and 37 DF,  p-value: 0.6002
```

These are not coincidences!

R^2 and regression residuals

R^2 (Coefficient of determination): proportion of variance in the dependent variable that is predictable from the independent variables. Provides a measure of how well our response/outcome are predicted by the model.

Total sum of squares (SS$_T$)

$$\sum_i (y_i - \overline{y})^2$$

Residual sum of squares (RSS)

$$\sum_i (\hat{y}_i - y_i)^2 = \sum_i \varepsilon_i^2$$

R^2

$$1 - \frac{RSS}{SS_T}$$

(Variance explained by the model)

Assumption of regression

1. The relationship between y (dependent variable) and x (independent variable) is linear.

2. The residuals do not vary with x. *(you'll hear more about this later in the course)*

3. The residuals are independent: the value of one residual is not influenced by the value of another (i.e., IID sample).

4. The **residuals** are normally distributed.

Increasing the complexity of our linear regression model ….

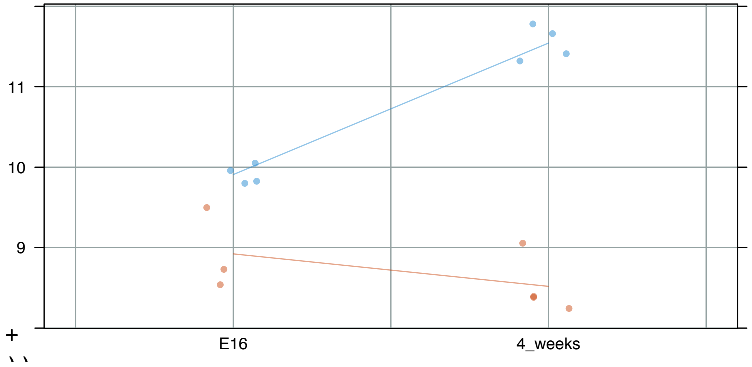
What if you have two categorical variable: genotype and time
(we will simplify the example by only consider two time points E16 vs 4wk)

What question do we want to ask?

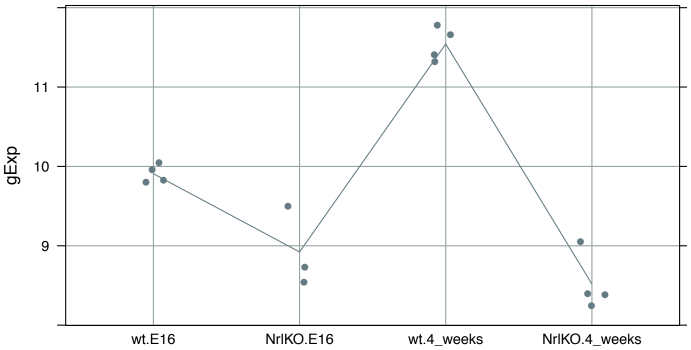
- Does the effect of one variable (factor) depends on the other

-> Interaction test (aka two-way anova in the context of categorical covariates)

Two-way anova/2-2 factorial design / interaction test

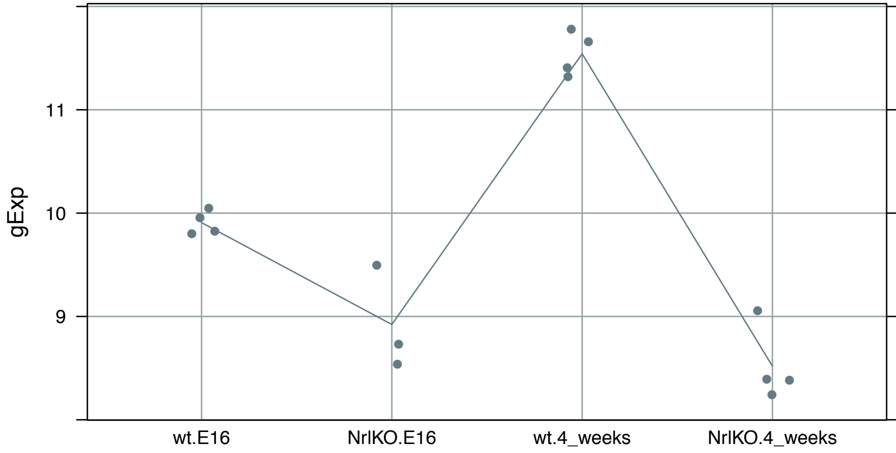


One-way anova – "four groups"



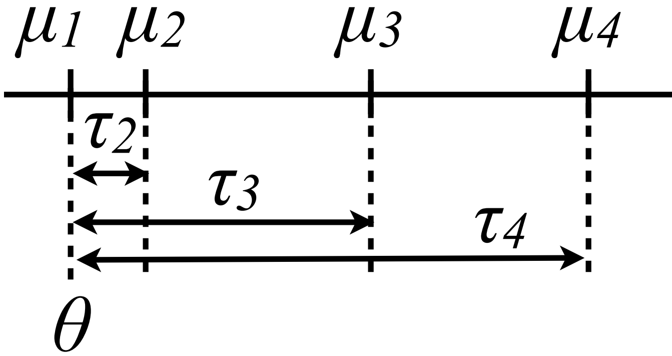
What if you don't use an interaction term, and just model everything linearly? ("4 group problem")

```
> 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
```

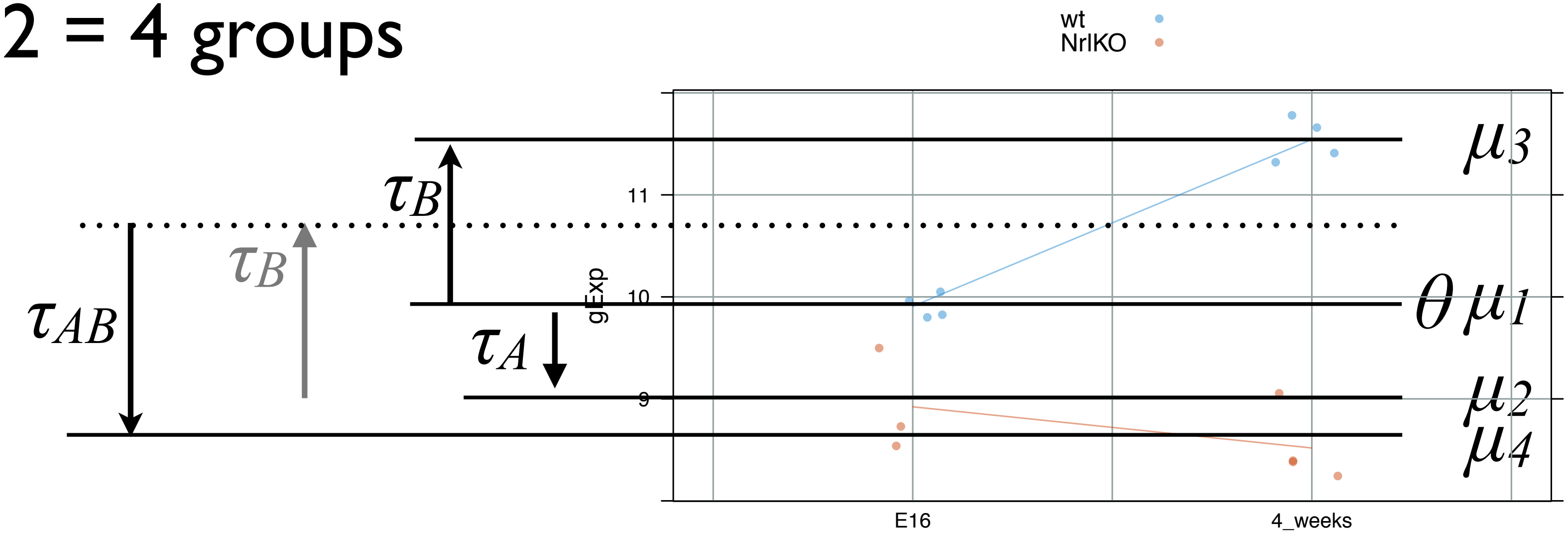


```
> summary(grpFit)
lm(formula = gExp ~ grp, data = miniDat)
<snip, snip>
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         9.9080     0.1575  62.911 2.03e-15 ***
grpNrlKO.E16       -0.9857     0.2406  -4.097  0.00177 **
grpwt.4_weeks       1.6345     0.2227   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
```

$$H_0 : \tau_j = 0$$



2 * 2 = 4 groups

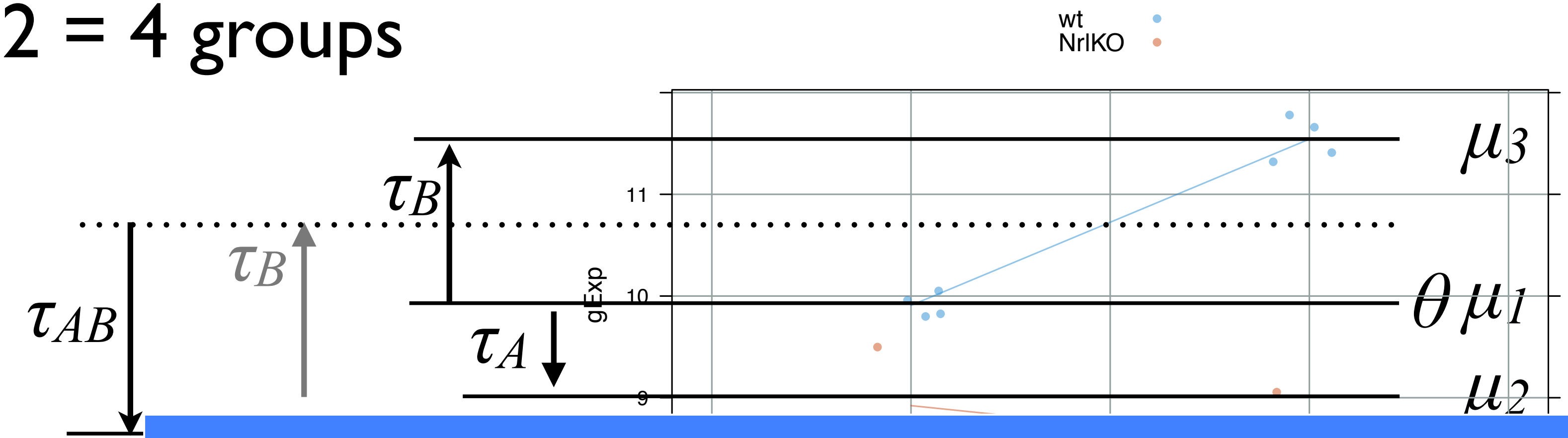


$$Y = X\alpha + \varepsilon$$

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

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

2 * 2 = 4 groups



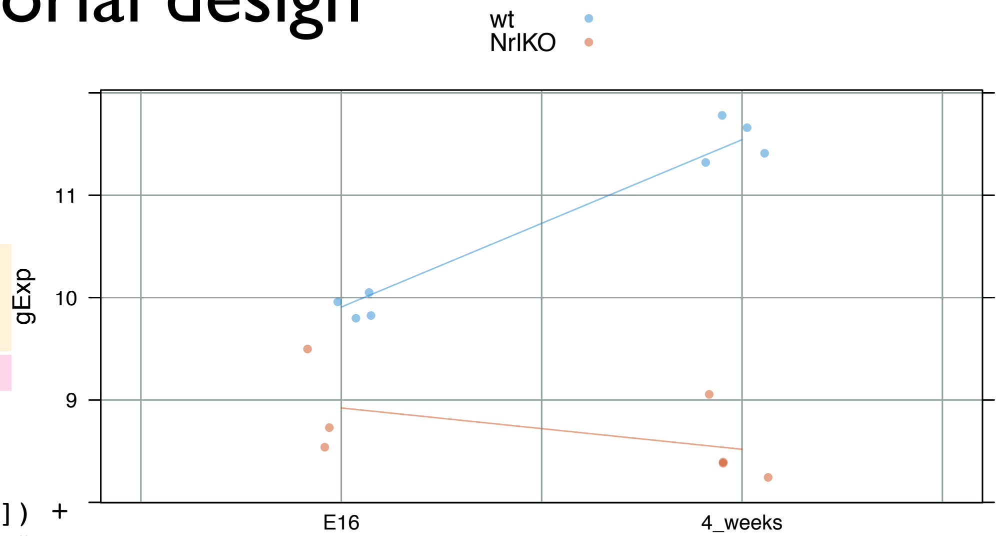
Terminology: main effect vs interaction effect

$Y =$

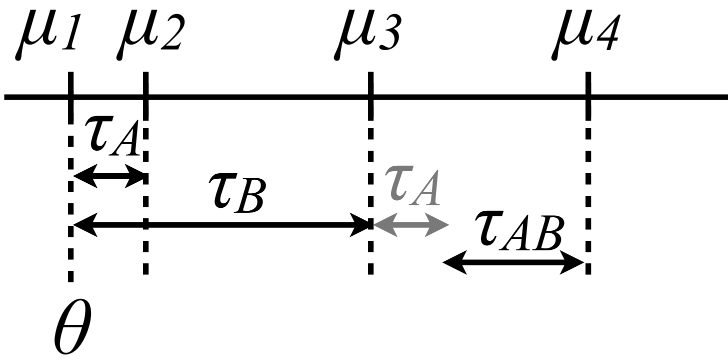
$$\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \\ y_{n_4 4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots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

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

“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 ***
```

$$H_0 : \tau_A = 0$$

$$H_0 : \tau_B = 0$$

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

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 “parameterize” a model and why you might do that

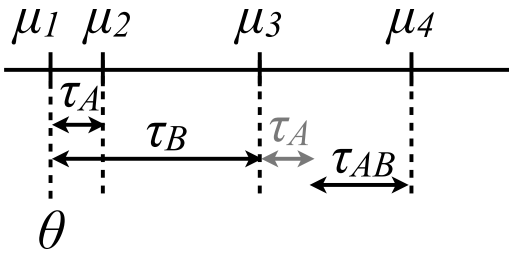
let’s look at a handful of genes/probesets to get a feel for all the ways a gene could be interesting or boring now ....

approaching with 2x2 factorial mindset

Let's through some example genes to get a sense of what an interaction effect looks like

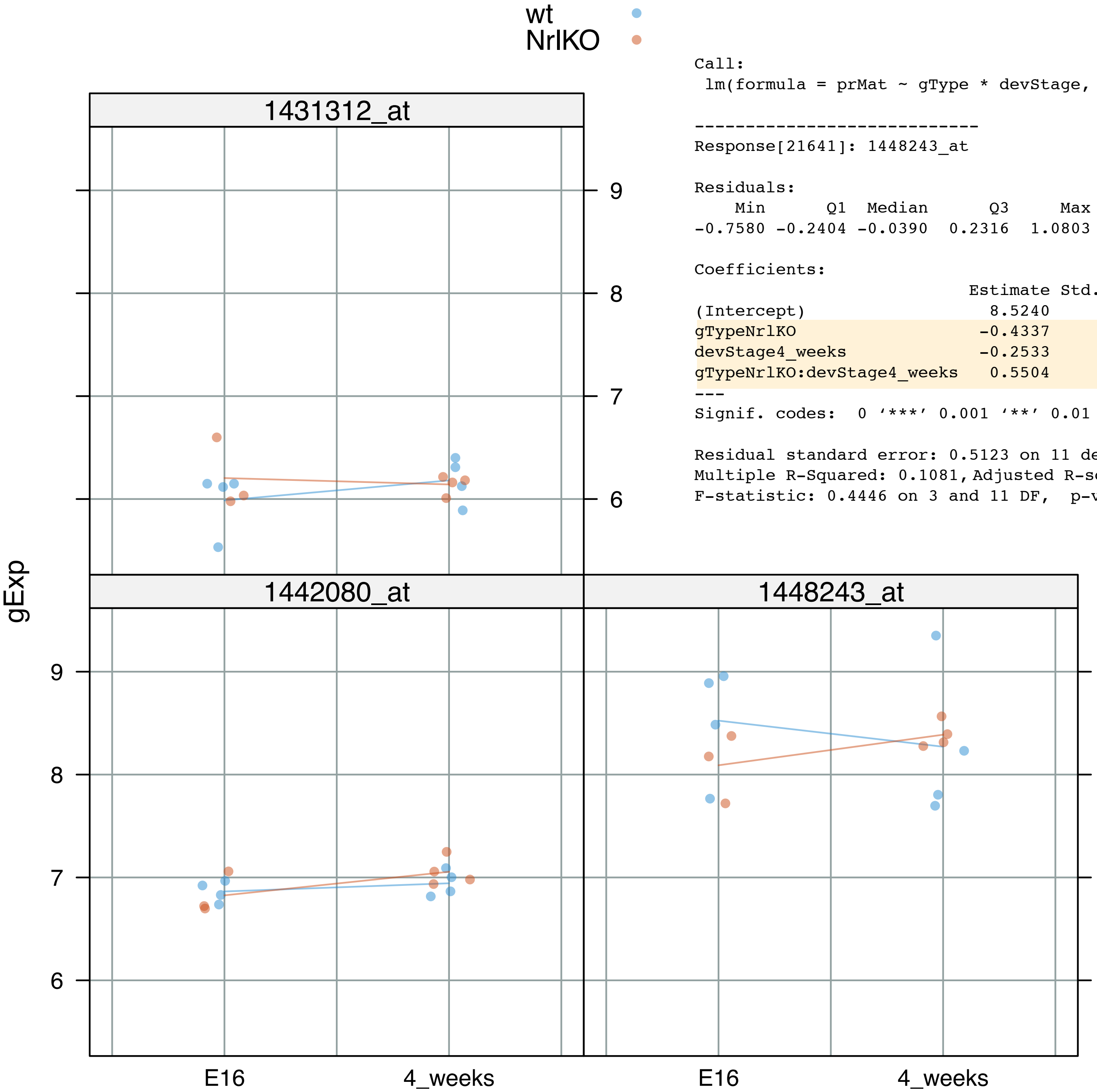
We have three parameters we'd like to interpret:

Main effect: genotype
Main effect: age
Interaction: genotype x age



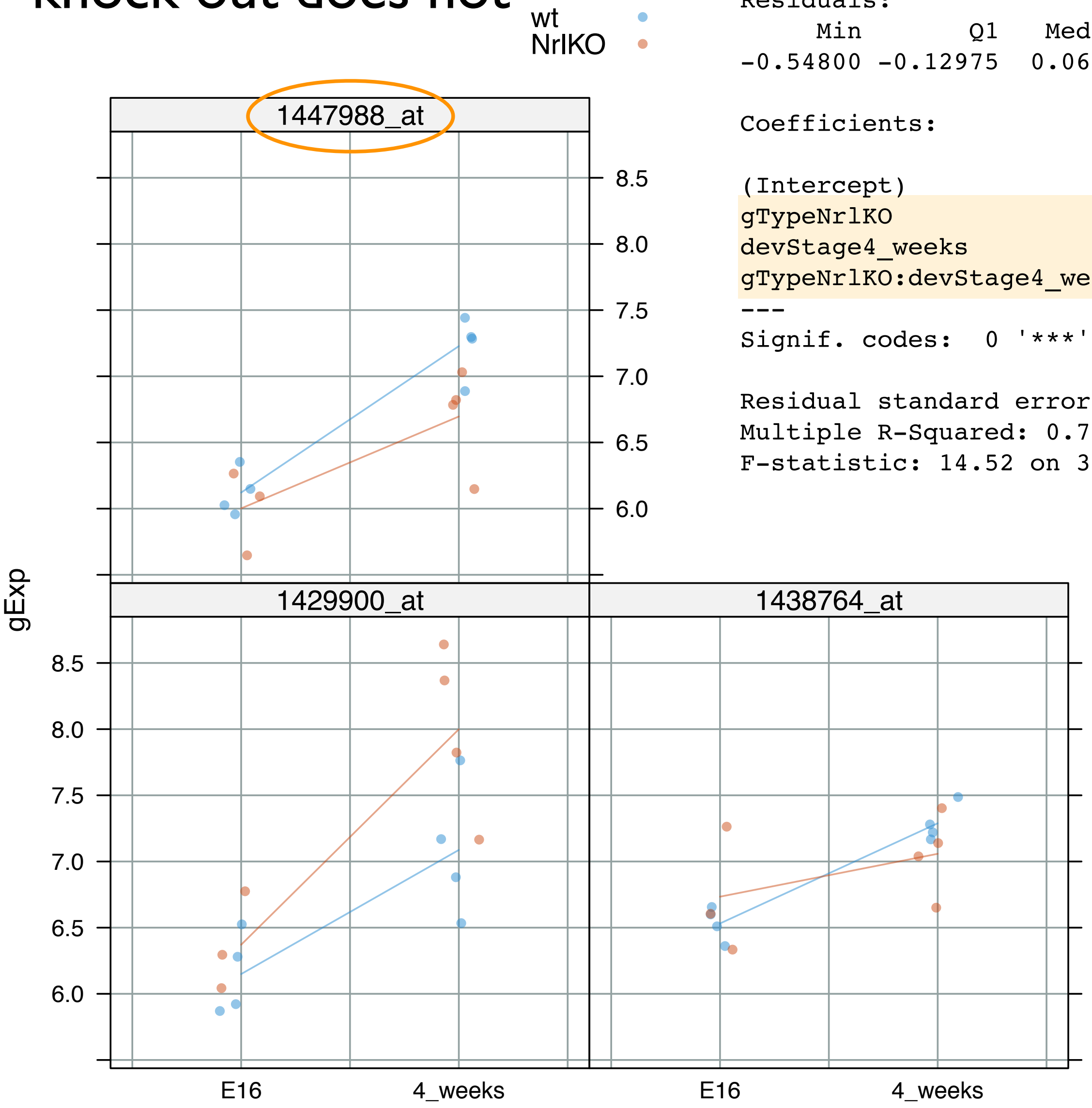
What are the possible conclusions:

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!



```
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
```

developmental stage matters, but gene knock out does not



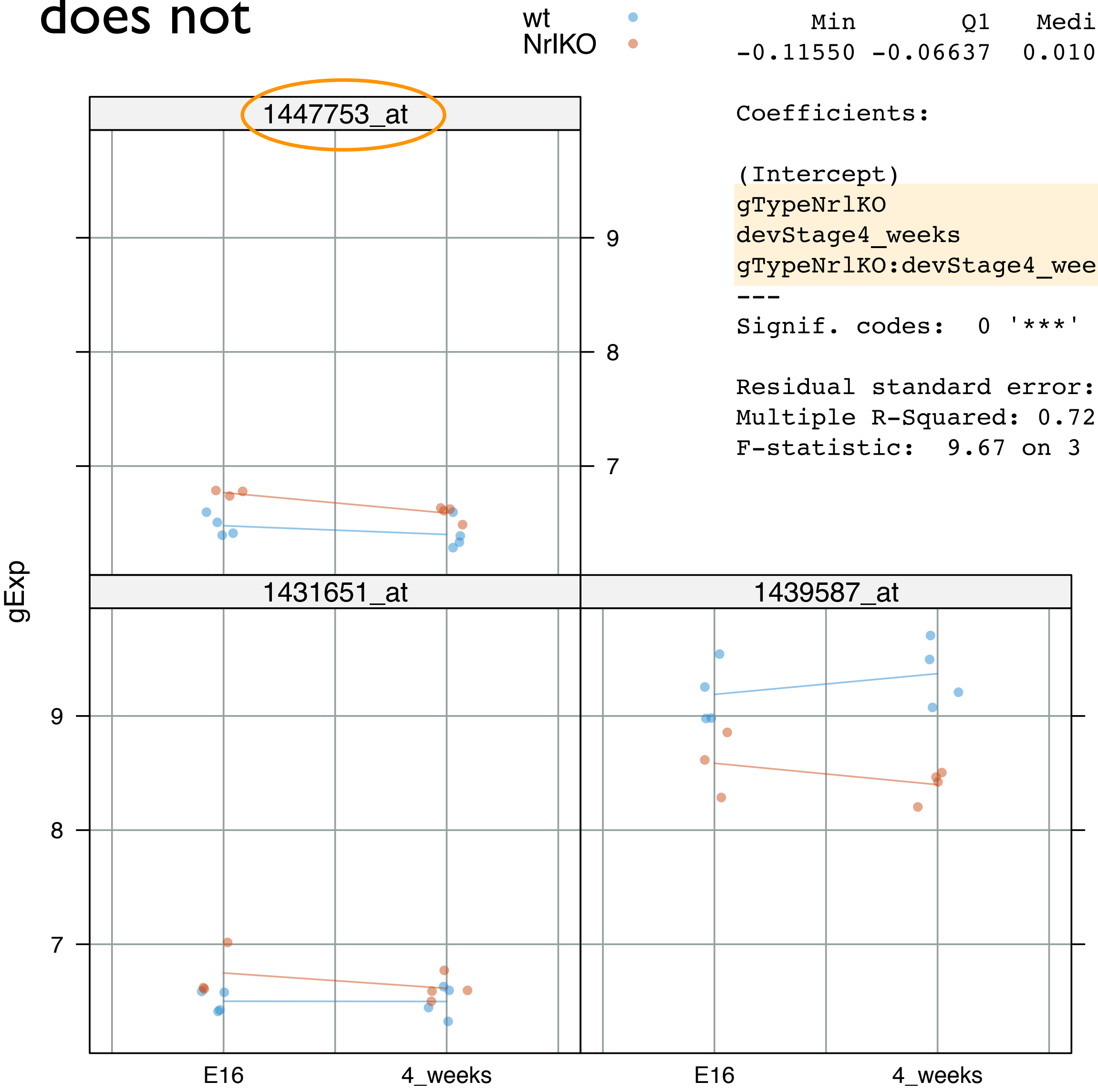
```
Call:
 lm(formula = prMatSimple ~ gType * devStage)

-----------------------------
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
```

$$H_0 : \tau_{\Delta Nrl} = 0$$ ✓

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

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

gene knock out matters, but developmental stage does not



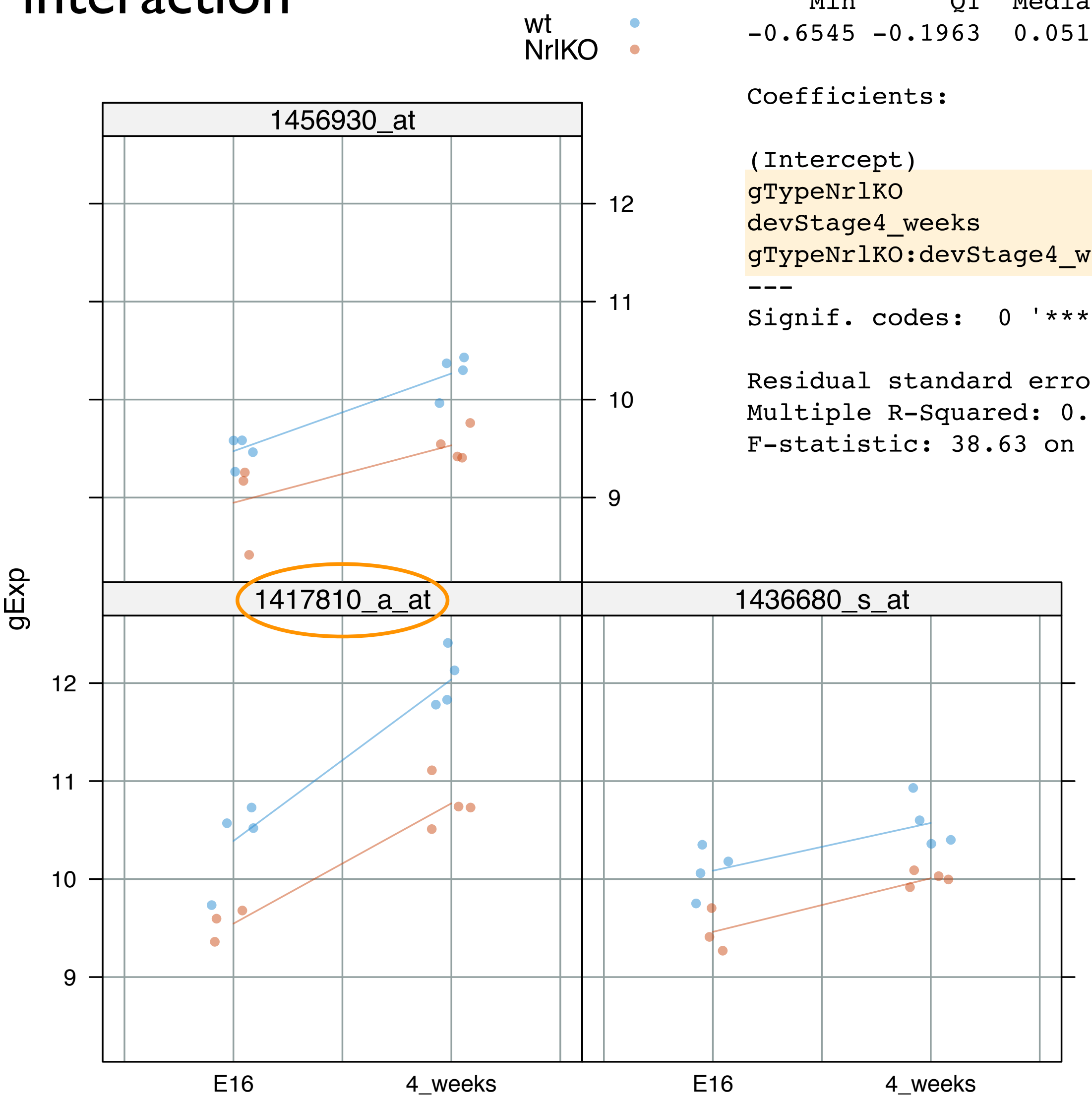
```
Call:
 lm(formula = prMatSimple ~ gType * devStage)

----------------------------
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
```

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

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

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

gene knock out & developmental stage matter, but no interaction



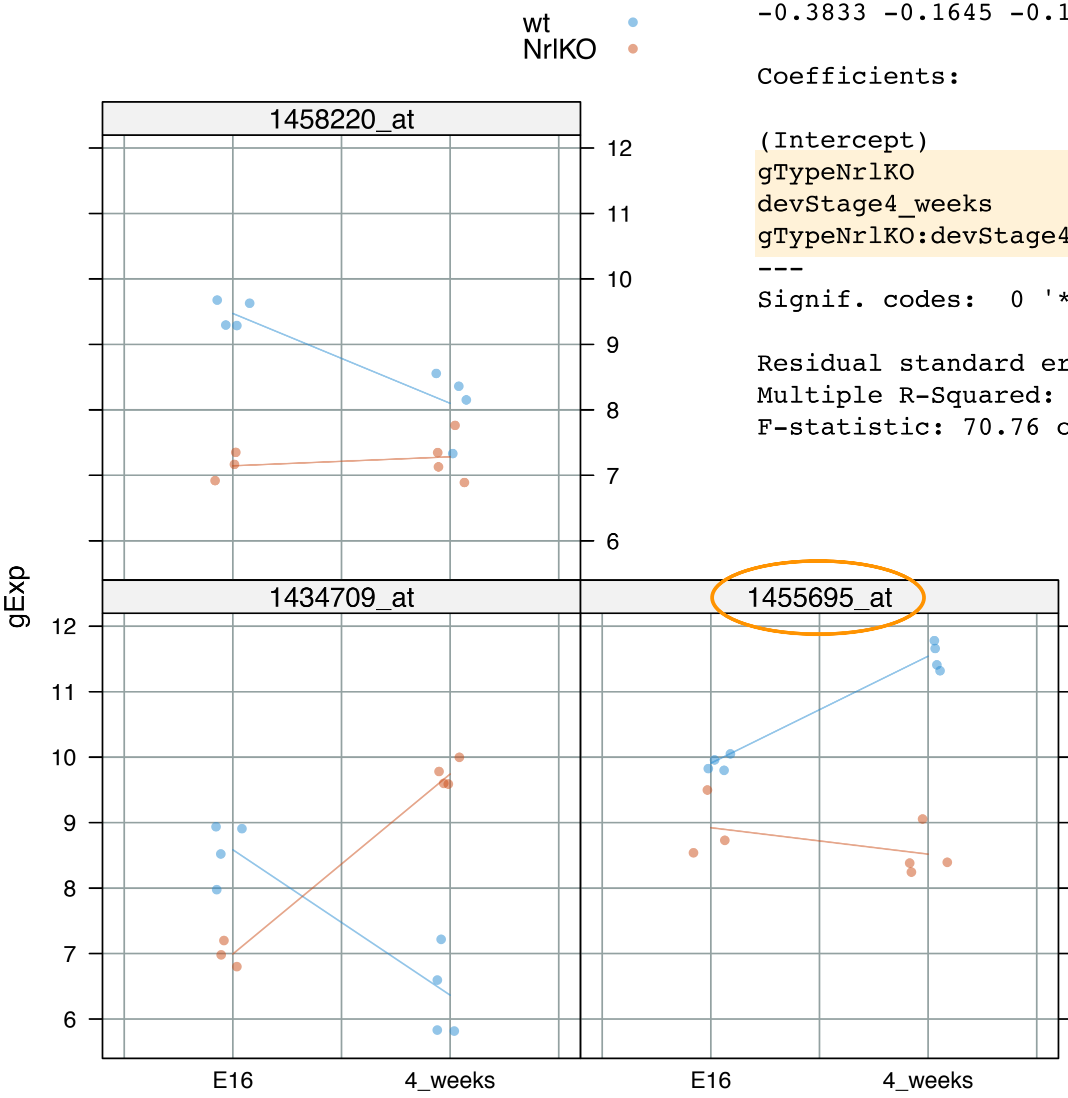
```
Call:
 lm(formula = prMatSimple ~ gType * devStage)

----------------------------
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
```

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

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

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

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 4weeks**

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?

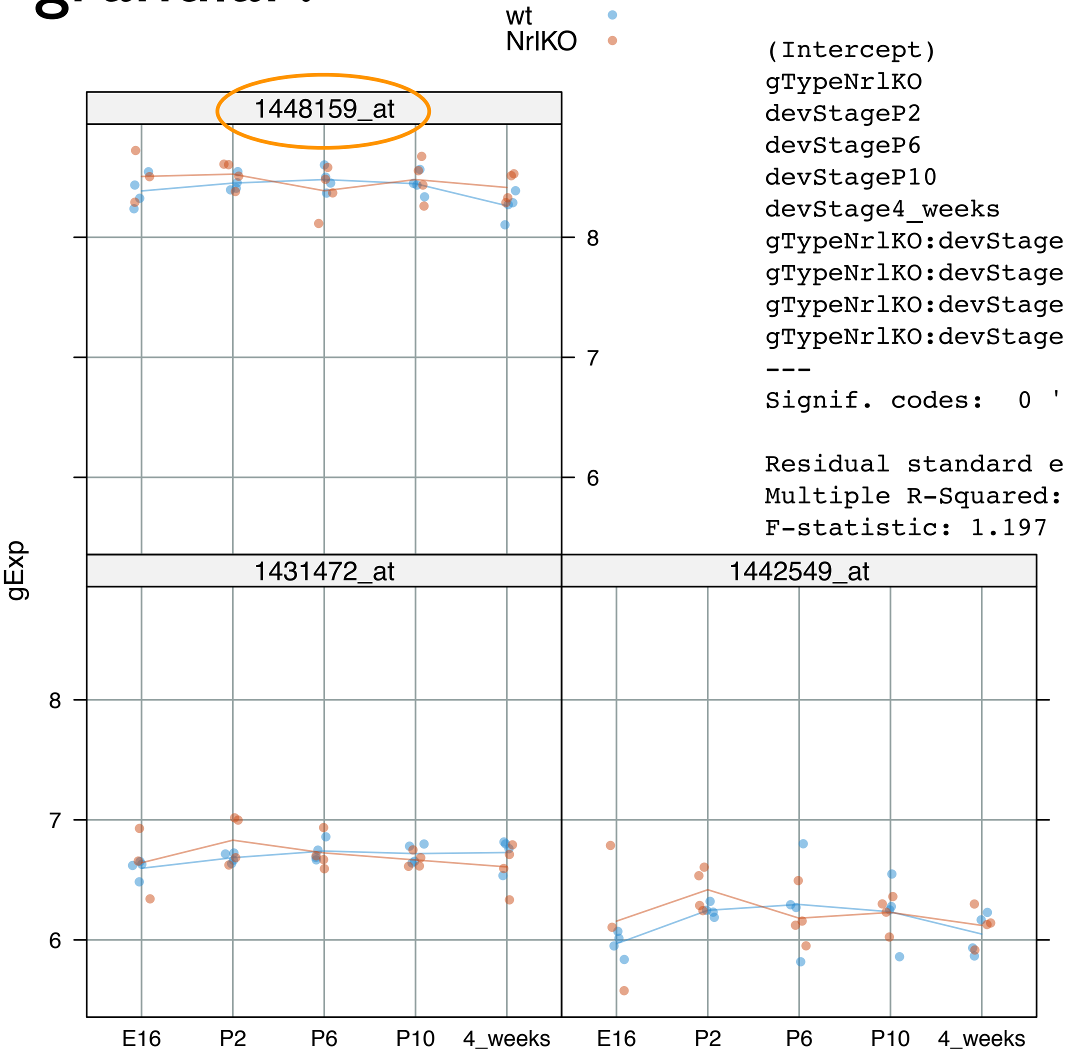
“two-way ANOVA” or ... just a linear model!

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

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?

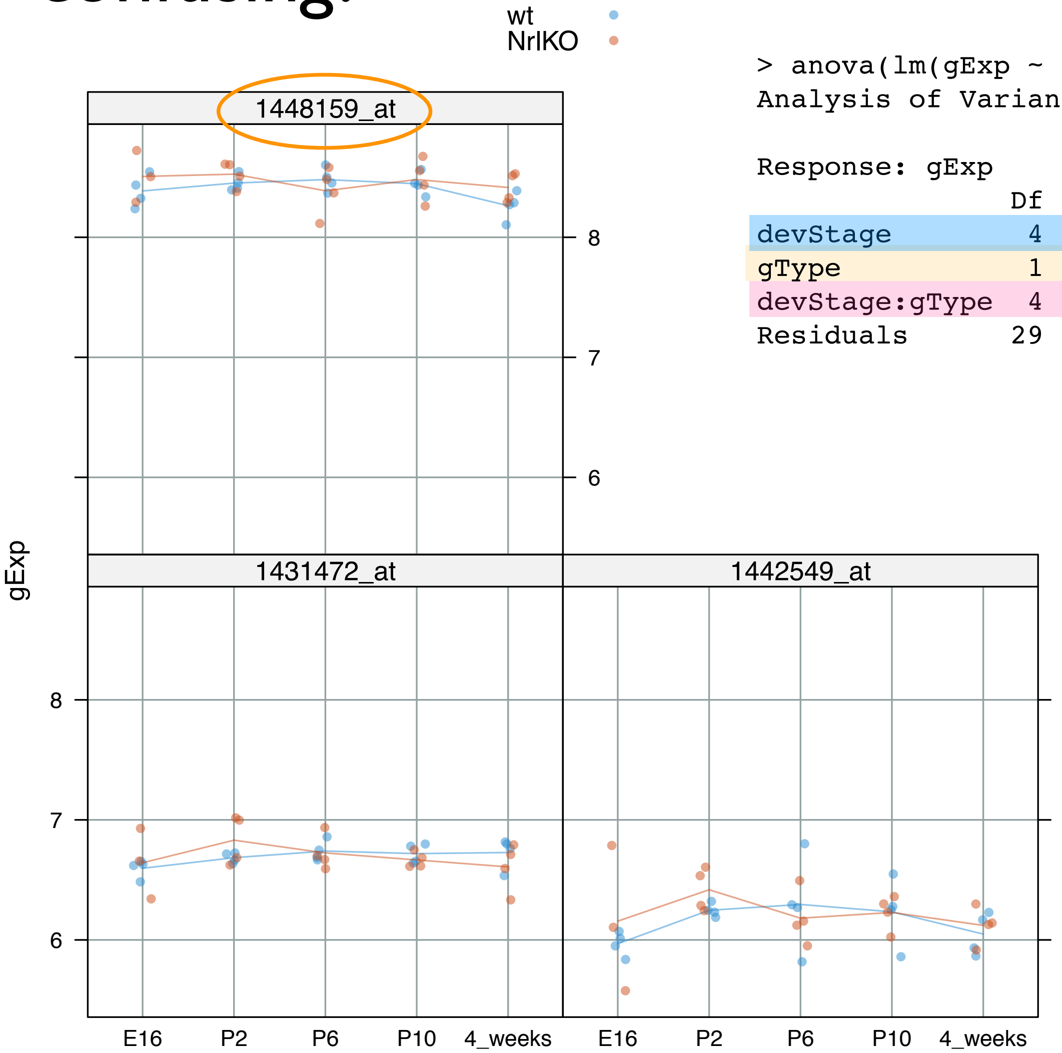


```
Call:
 lm(formula = prMat ~ gType * devStage)

----------------------------
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 :(

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}

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

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

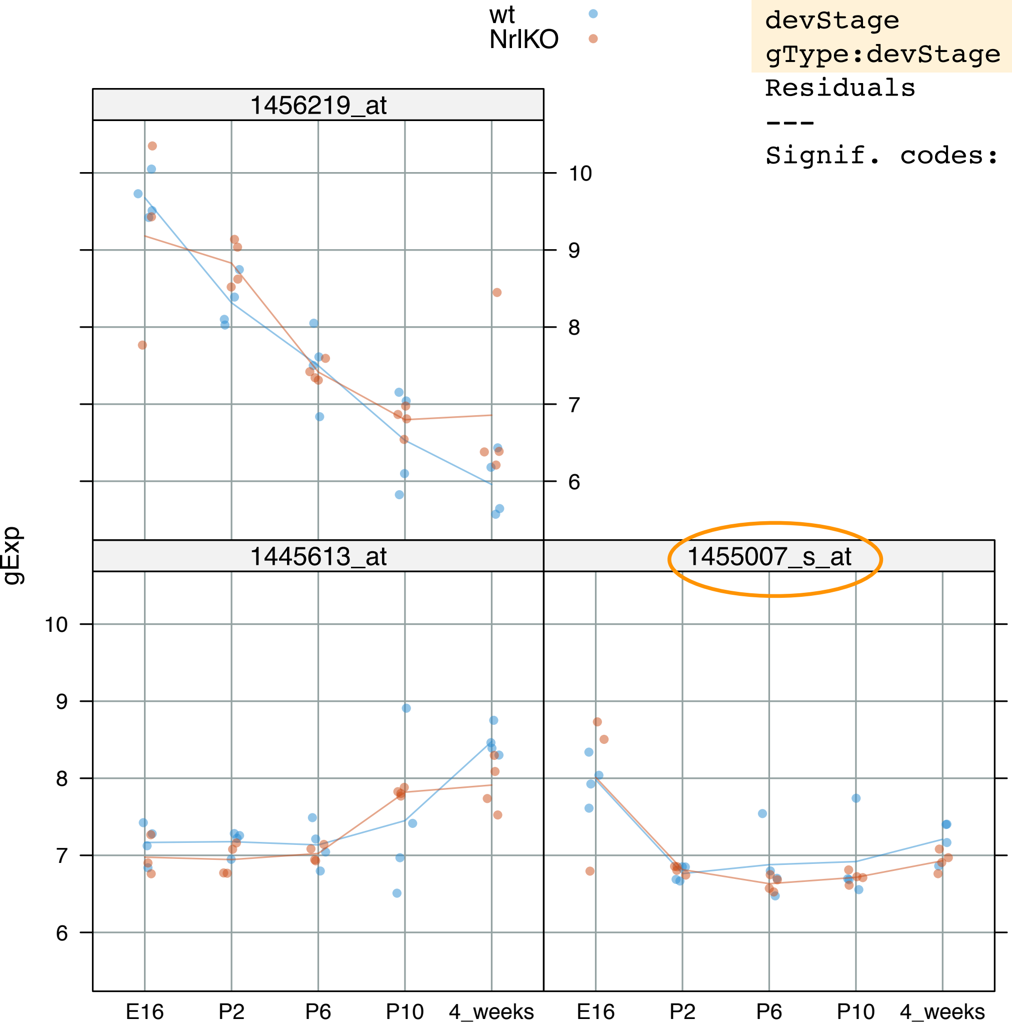
by definition:

$p_{small} < p_{big}$

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

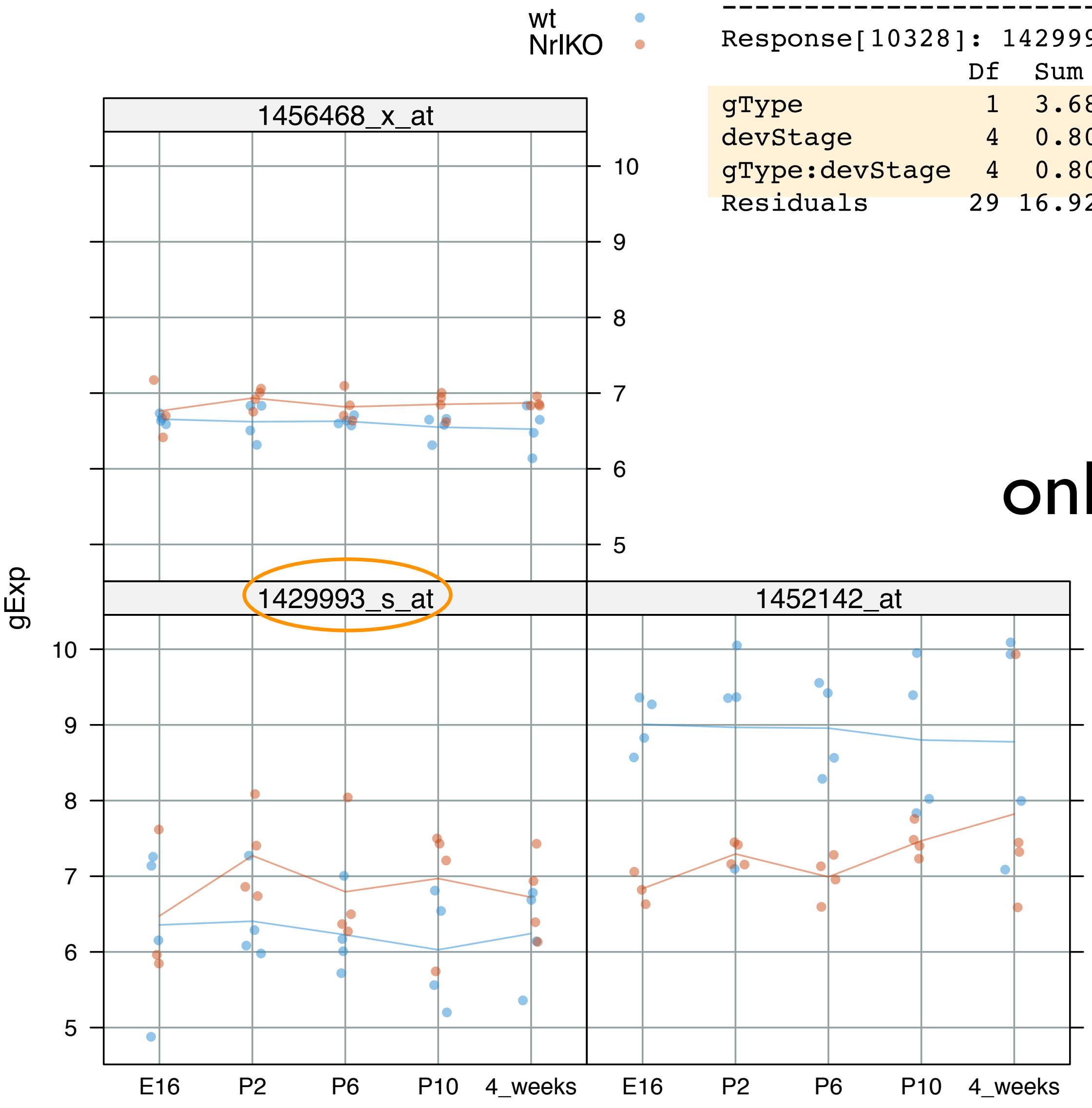
$$F = \frac{\left(\dfrac{RSS_{small} - RSS_{big}}{p_{big} - p_{small}} \right)}{\dfrac{RSS_{big}}{n - p_{big}}} \sim_{H_0} F_{(p_{big} - p_{small}, n - p_{big})}$$

```
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
```

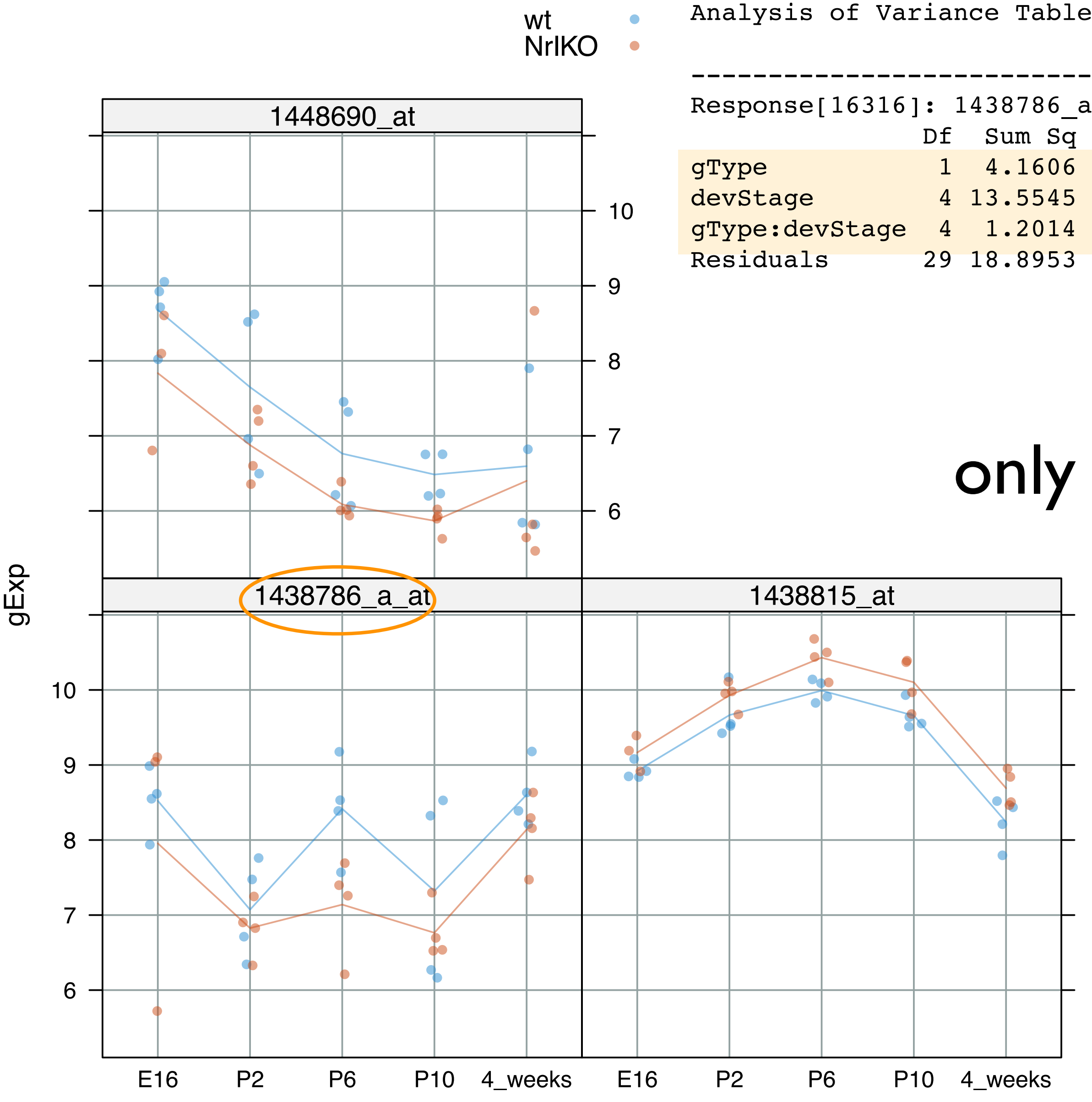


only devStage matters

```
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                  
```

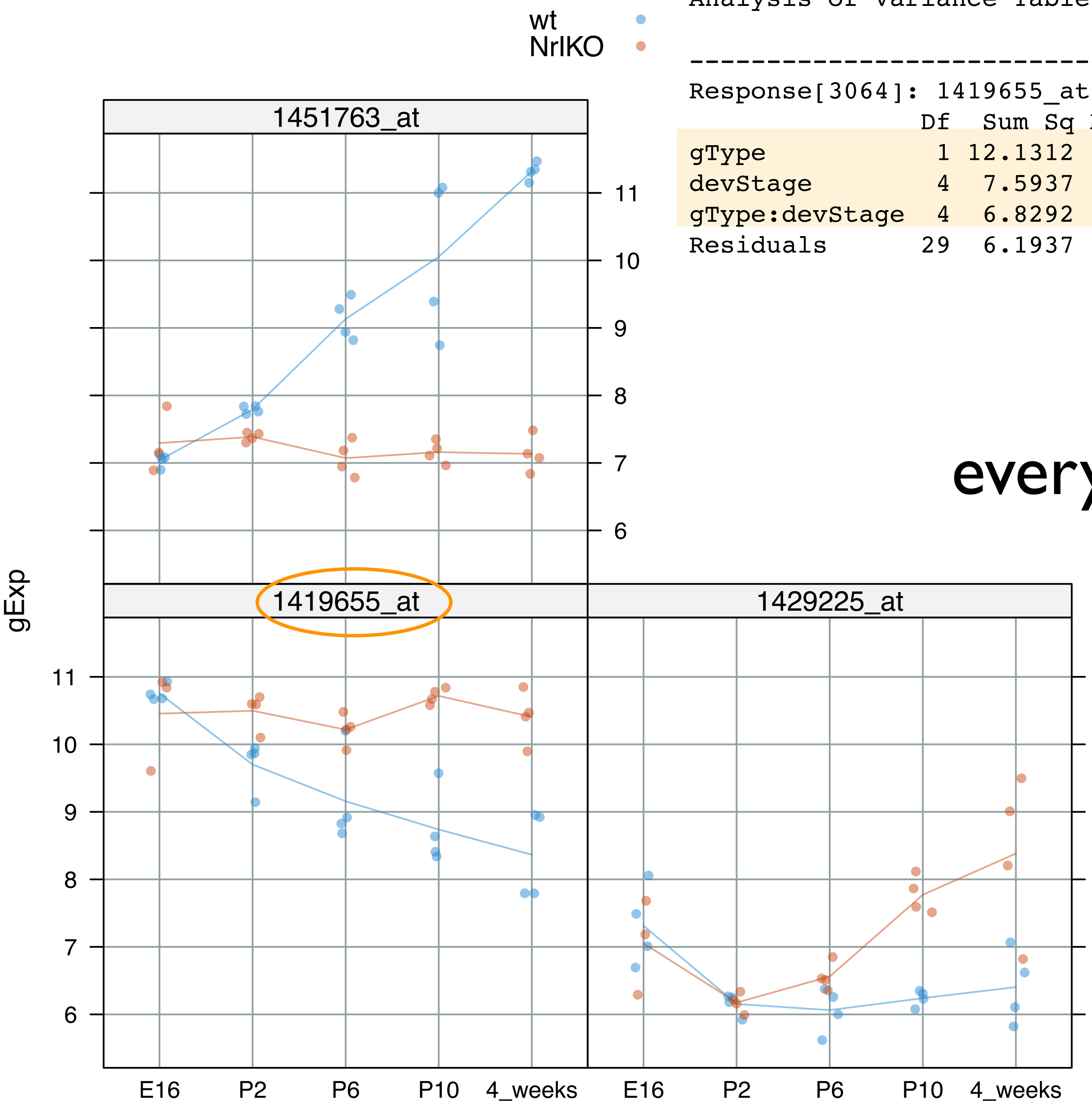


only gType matters



```
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

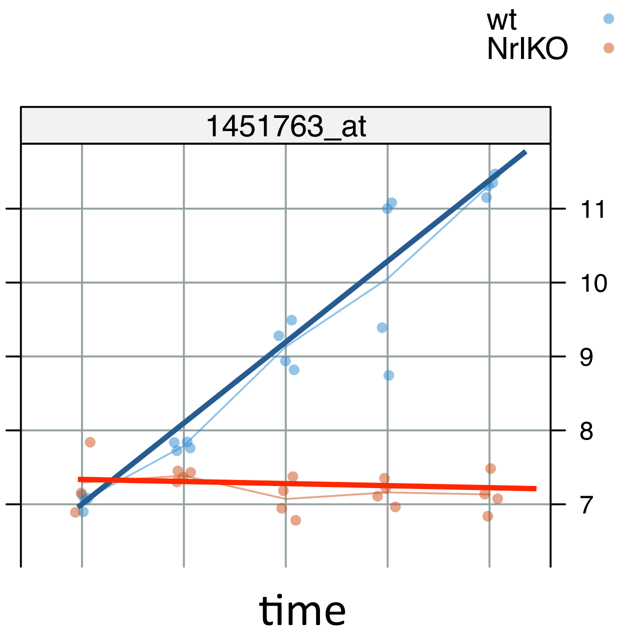


```
Analysis of Variance Table

----------------------------
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

Seems awkward to model a categorical variables with many levels (e.g., devStage).

We are estimating many parameters with little data. Isn't there a better way to model this type of data?? YES – treat your variables as quantitative when possible



Are the slopes of the two lines different from zero and from each other?