Linear models and ANOVA

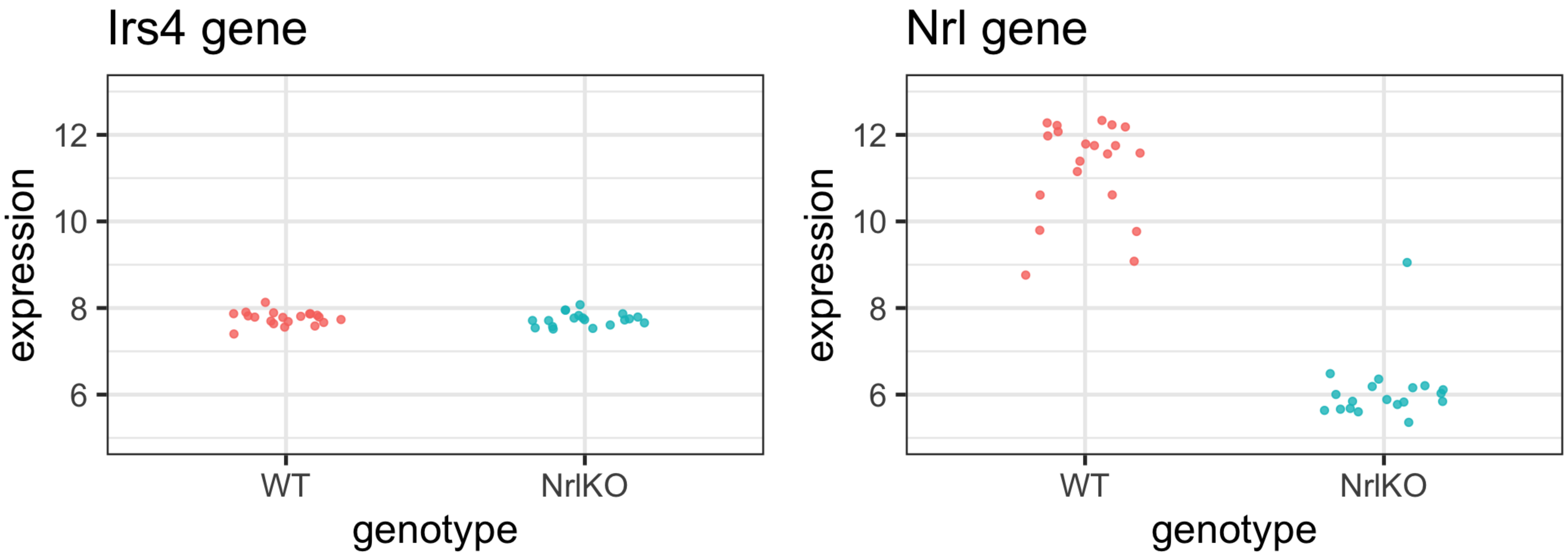
Keegan Korthauer

January 26, 2023

Recap: Are these genes different in NrlKO vs WT?

H_0: the expression level of gene g is the same in both genotypes

Is there **enough** evidence in the data to reject H_0?



Learn about a population from a random sample

Population (Unknown)

$$Y \sim F,\ Z \sim G$$

$$E[Y] = \mu_Y,\ E[Z] = \mu_Z$$

$$Var[Y] = \sigma^2_Y,\ Var[Z] = \sigma^2_Z$$

$$H_0 : \mu_Y = \mu_Z$$

$$H_A : \mu_Y \neq \mu_Z$$

Sample (Observed, with randomness)

$$(Y_1, Y_2, \ldots, Y_{n_Y}) \text{ and } (Z_1, Z_2, \ldots, Z_{n_Z})$$

$$\hat{\mu}_Y = \bar{Y} = \frac{\sum_{i=1}^{n_Y} Y_i}{n_Y}$$

$$\hat{\sigma}^2_Y = S^2_Y = \frac{1}{n_Y}\sum_{i=1}^{n_Y}(Y_i - \bar{Y})^2$$

(with similar quantities for $Z : \bar{Z}$ and S^2_Z)

$$T = \frac{\bar{Y} - \bar{Z}}{\sqrt{\widehat{Var}(\bar{Y} - \bar{Z})}}$$

$\bar{Y}, \bar{Z}, S^2_Y, S^2_Z$ and T are examples of **statistics** computed from the sample

Summary: Hypothesis testing

1. Formulate scientific hypothesis as a **statistical hypothesis** $(H_0 \text{ vs } H_A)$
2. Define a **test statistic** to test H_0 and compute its **observed value**. For example:

- 2-sample *t*-test
- Welch’s *t*-test (unequal variance)
- Wilcoxon rank-sum test
- Kolmogorov-Smirnov test

3. Compute the probability of seeing a test statistic as extreme as that observed, under the **null sampling distribution** (p-value)
4. Make a decision about the **significance** of the results, based on a pre-specified significance level (α)

We can run these tests in R

Example: use the `t.test` function to test H_0 using a 2-sample *t*-test with equal variance:

```r
filter(twoGenes, gene == "Irs4") %>%
  t.test(expression ~ genotype, data = ., var.equal = TRUE)
```

```
	Two Sample t-test

data:  expression by genotype
t = 0.52854, df = 37, p-value = 0.6003
alternative hypothesis: true difference in means between group WT and group NrlKO is not equal to 0
95 percent confidence interval:
 -0.07384018  0.12595821
sample estimates:
   mean in group WT mean in group NrlKO 
           7.765671            7.739612 
```

Discussion recap

- What test should I use?
  - What test(s) might be appropriate if your sample size is just barely large enough to invoke CLT, but you also have suspected outliers?
  - If more than one test is appropriate (e.g. *t*-test, Wilcoxon, and KS), which should we report?
  - What should you do if methods that are equally appropriate and defensible give very different answers?
- What is generally more important for results interpretation: the effect size or the p-value?

Today’s Learning Objectives

1. Compare means of different groups (2 or more) using a **linear regression model**
2. Use ‘indicator’ variables to represent the levels of a qualitative explanatory variable
3. Write a linear model using matrix notation and understand which matrix is built by R
4. Distinguish between **single** and **joint** hypothesis tests (e.g. t-tests vs F-tests)

3 ways to test H$_0$: $\mu_1 = \mu_2$

t-test ANOVA linear regresion

2-sample t-test (with equal variance)

```r
filter(twoGenes, gene == "Irs4") %>%
  t.test(expression ~ genotype, data = ., var.equal = TRUE)
```

```

	Two Sample t-test

data:  expression by genotype
t = 0.52854, df = 37, p-value = 0.6003
alternative hypothesis: true difference in means between group WT and group NrlKO is not equal to 0
95 percent confidence interval:
 -0.07384018  0.12595821
sample estimates:
   mean in group WT mean in group NrlKO 
           7.765671            7.739612 
```

1. but you can change that!

These are not coincidences!

t-test ANOVA linear regresion

2-sample t-test (with equal variance)

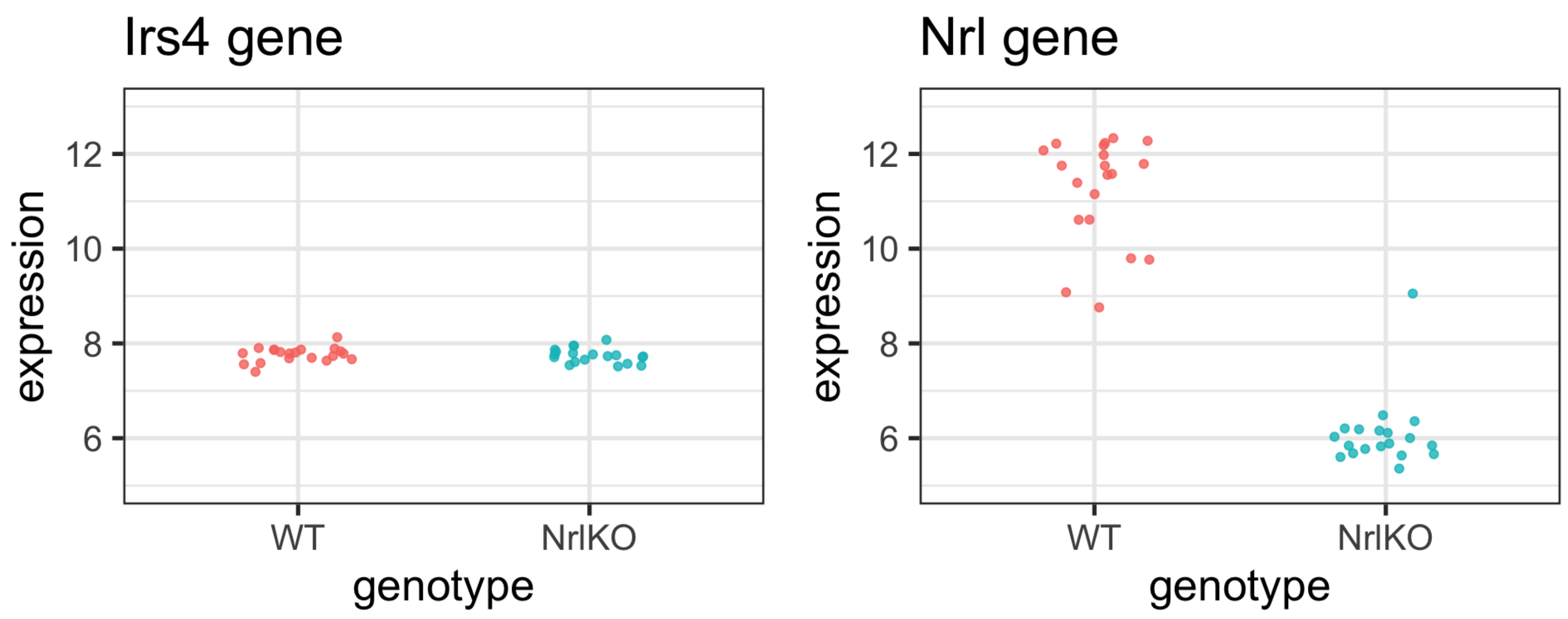
```
$`t statistic`
        t 
0.5285386 

$`p-value`
[1] 0.6002819

$`mean difference`
[1] 0.02605902

$`(t statistic)^2`
       t 
0.279353 
```

Key Question

Why are these giving us the same results?

1. Note that the t statistic squared is equal to the ANOVA F statistic

t-test vs *line*ar regression: where’s the *line*[1]?



Key Question

Why can we run a t-test with a **linear** regression model?

[1] Note that the *x*-axis in these plots is not numerical, thus a line in this space does not have any mathematical meaning

From *t*-test to linear regression

Let’s change the notation to give a common framework to all methods

$$Y \sim G;\; E[Y] = \mu_Y$$

$$\downarrow$$

$$Y = \mu_Y + \varepsilon_Y;\; \varepsilon_Y \sim G;\; E[\varepsilon_Y] = 0$$

Why is this equivalent?

$$E[Y] = E[\mu_Y + \varepsilon_Y] = \mu_Y + E[\varepsilon_Y] = \mu_Y$$

We are just rewriting Y here

From *t*-test to linear regression

Let’s change the notation to give a common framework to all methods

$$Y \sim G;\ \ E[Y] = \mu_Y$$

↓

$$Y = \mu_Y + \varepsilon_Y;\ \ \varepsilon_Y \sim G;\ \ E[\varepsilon_Y] = 0$$

We can use indices to accommodate multiple groups, i.e.,

$$Y_{ij} = \mu_j + \varepsilon_{ij};\quad \varepsilon_{ij} \sim G_j;\quad E[\varepsilon_{ij}] = 0;$$

where $j = \{\mathrm{WT}, \mathrm{NrlKO}\}$ (or $j = \{1, 2\}$) identifies the groups; and $i = 1, \ldots, n_j$ identifies the observations within each group

For example: Y_{11} is the first observation in group 1 or WT

This is called the cell-means model

Using data from the model

$$Y_{ij} = \mu_j + \varepsilon_{ij}; \quad \varepsilon_{ij} \sim G; \quad E[\varepsilon_{ij}] = 0;$$

where j indexes groups (e.g. WT vs NrlKO) and i indexes samples within group, the goal is to test $H_0 : \mu_1 = \mu_2$

Note

In the **cell-means** model parameterization, we have a parameter $E[Y_{ij}] = \mu_j$ that represents the population mean of each group (in our example: genotype)

Important

We assume a common distribution G for all groups (equal variance assumption)

Why the name? ‘Cell’ here refers to a cell of a table - e.g. make a table of means by group, and μ_j represents the population value for each cell j in the table

Recall: sample mean estimator of population mean

- For each group j, the **population** mean is given by $E[Y_{ij}] = \mu_j$
- A natural *estimator* of the population mean μ_j is the **sample** mean $\hat{\mu}_j = \bar{Y} = \frac{\sum_{i=1}^{n_j} Y_{ij}}{n_j}$
- Recall that the `t.test` function calculates these for us in R

But why does the `lm` function report different estimates?

`t.test` `lm`

```r
# t.test
filter(twoGenes, gene == "Irs4") %>%
  t.test(expression ~ genotype, data = ., var.equal = TRUE)
```

```
	Two Sample t-test

data:  expression by genotype
t = 0.52854, df = 37, p-value = 0.6003
alternative hypothesis: true difference in means between group WT and group NrlKO is not equal to 0
95 percent confidence interval:
 -0.07384018  0.12595821
sample estimates:
   mean in group WT mean in group NrlKO 
           7.765671            7.739612 
```

- `(Intercept)` estimate from `lm` is the **sample mean** of WT group
- `genotypeNrlKO` estimate from `lm` is **not** the sample mean of the NrlKO group… what is it?

Parameterization: how to write the model?

- By default, the `lm` function does not use the cell-means parameterization
- Usually, the goal is to *compare* the means, not to study each in isolation

Let’s let $\theta = \mu_1$ and rewrite $\mu_j = \theta + \tau_j$, and plug into **cell-means** (μ_j) model:

$$Y_{ij} = \mu_j + \varepsilon_{ij}; \quad \varepsilon_{ij} \sim G; \quad E[\varepsilon_{ij}] = 0;$$

$$\downarrow$$

This gives us the **reference-treatment effect** (θ, τ_j) model:

$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij}; \quad \tau_1 = 0, \quad \varepsilon_{ij} \sim G; \quad E[\varepsilon_{ij}] = 0;$$

Reference-treatment effect parameterization

Reference-treatment effect (θ, τ_j) model:

$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij}; \quad \tau_1 = 0, \quad \varepsilon_{ij} \sim G; \quad E[\varepsilon_{ij}] = 0;$$

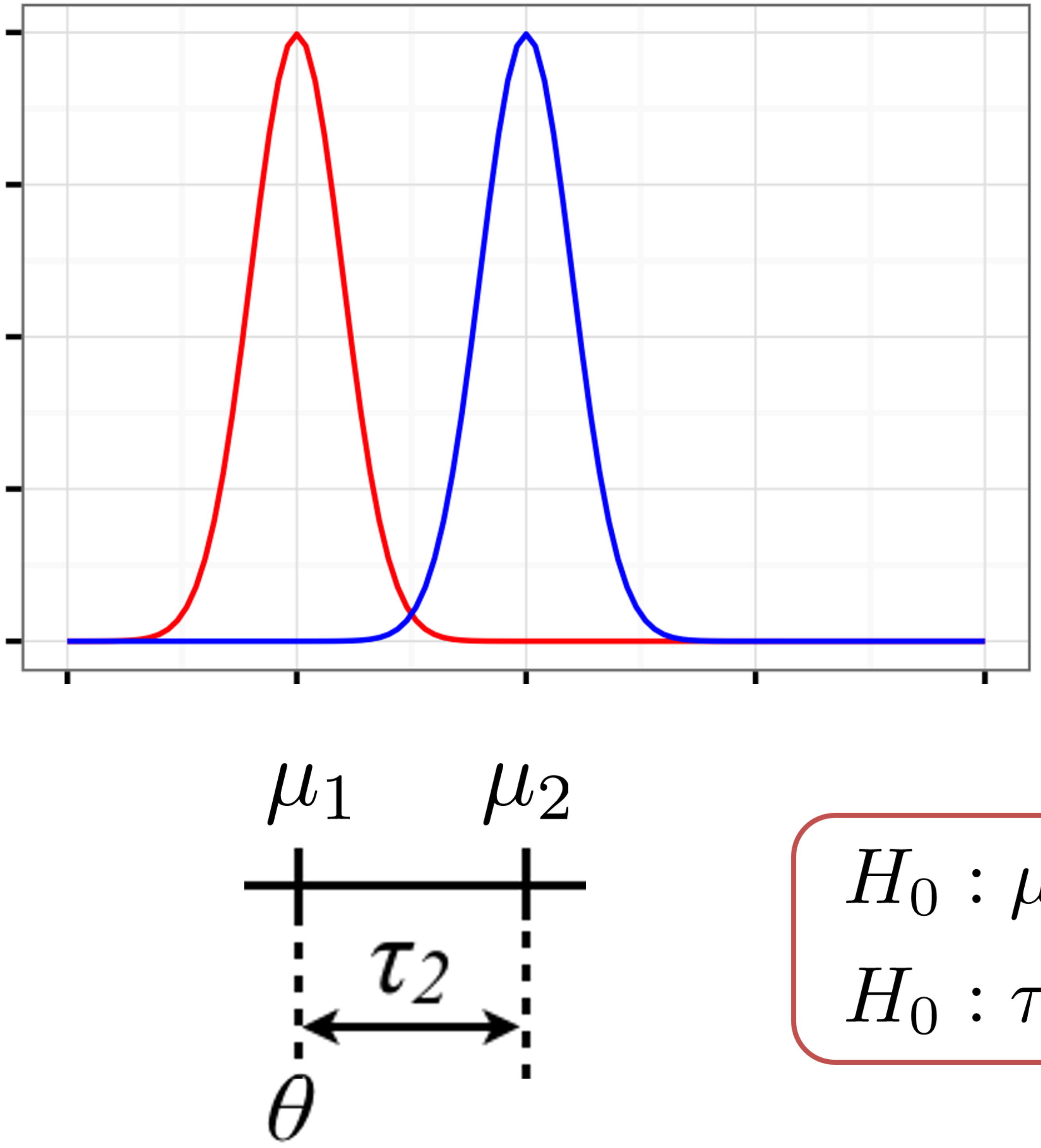
- Note that for each group, the population mean is given by $E[Y_{ij}] = \theta + \tau_j = \mu_j$, and $\tau_2 = \mu_2 - \mu_1 = E[Y_{i2}] - E[Y_{i1}]$ *compares* the means
- τ_1 must be set to zero, since group 1 is the *reference* group

ⓘ **Note**

In the **reference-treatment effect** model parameterization, we have the following parameters:

- θ represents the population mean of the reference group (in our example: WT)
- τ_j represents the difference in the population mean of group j compared to the reference (in our example: NrlKO - WT)

Relation between parameterizations



$$H_0 : \mu_1 = \mu_2$$

$$H_0 : \tau_2 = 0$$

`lm` output

- the sample mean of the WT group (**reference**): $\hat{\theta}$
- the difference in sample mean of NrlKO and WT groups (**treatment effect**): $\hat{\tau}_2$

Irs4 Nrl

▶ Code

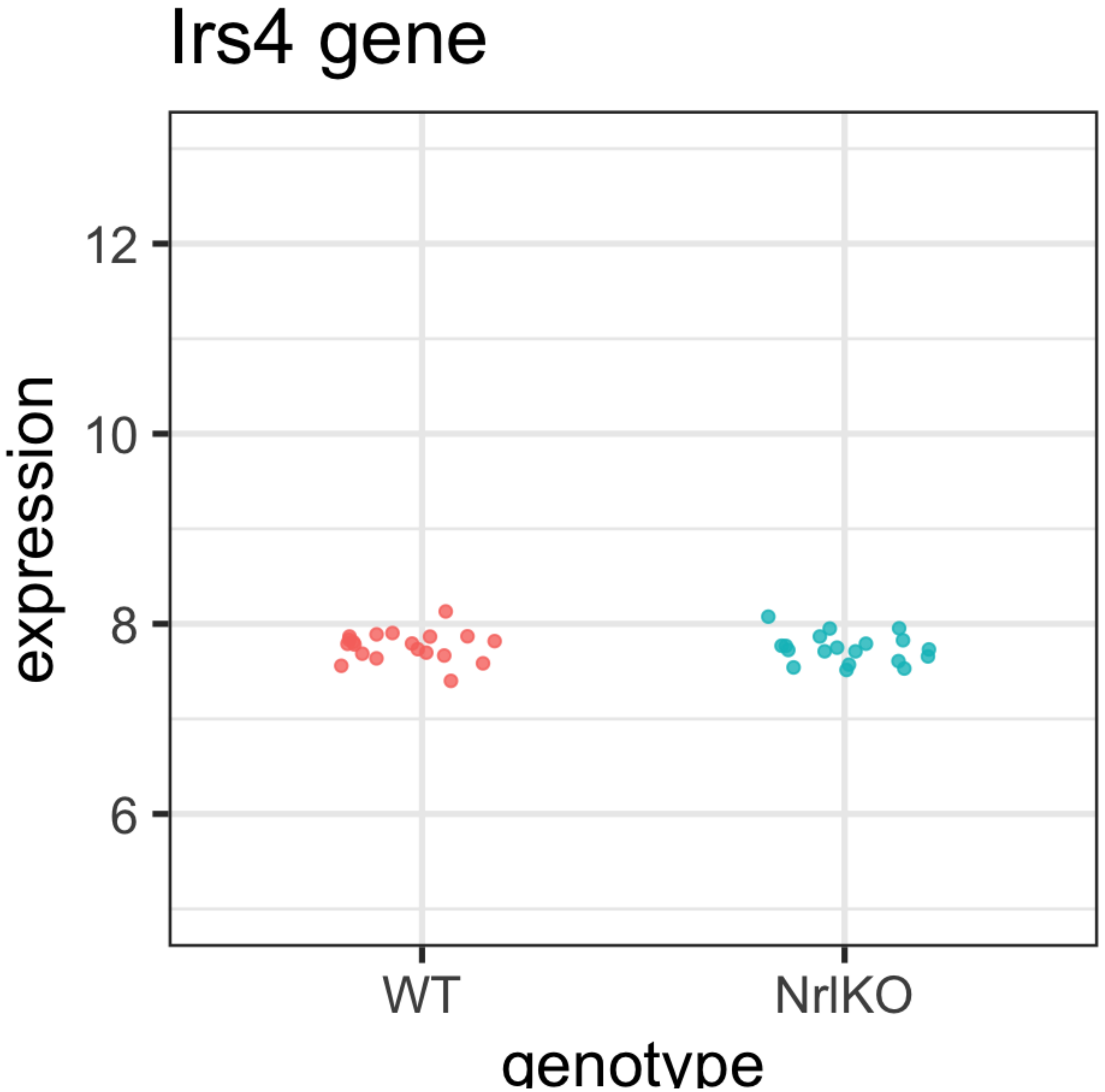
```
# A tibble: 1 × 3
     WT NrlKO diffExp
  <dbl> <dbl>   <dbl>
1  7.77  7.74 -0.0261
```

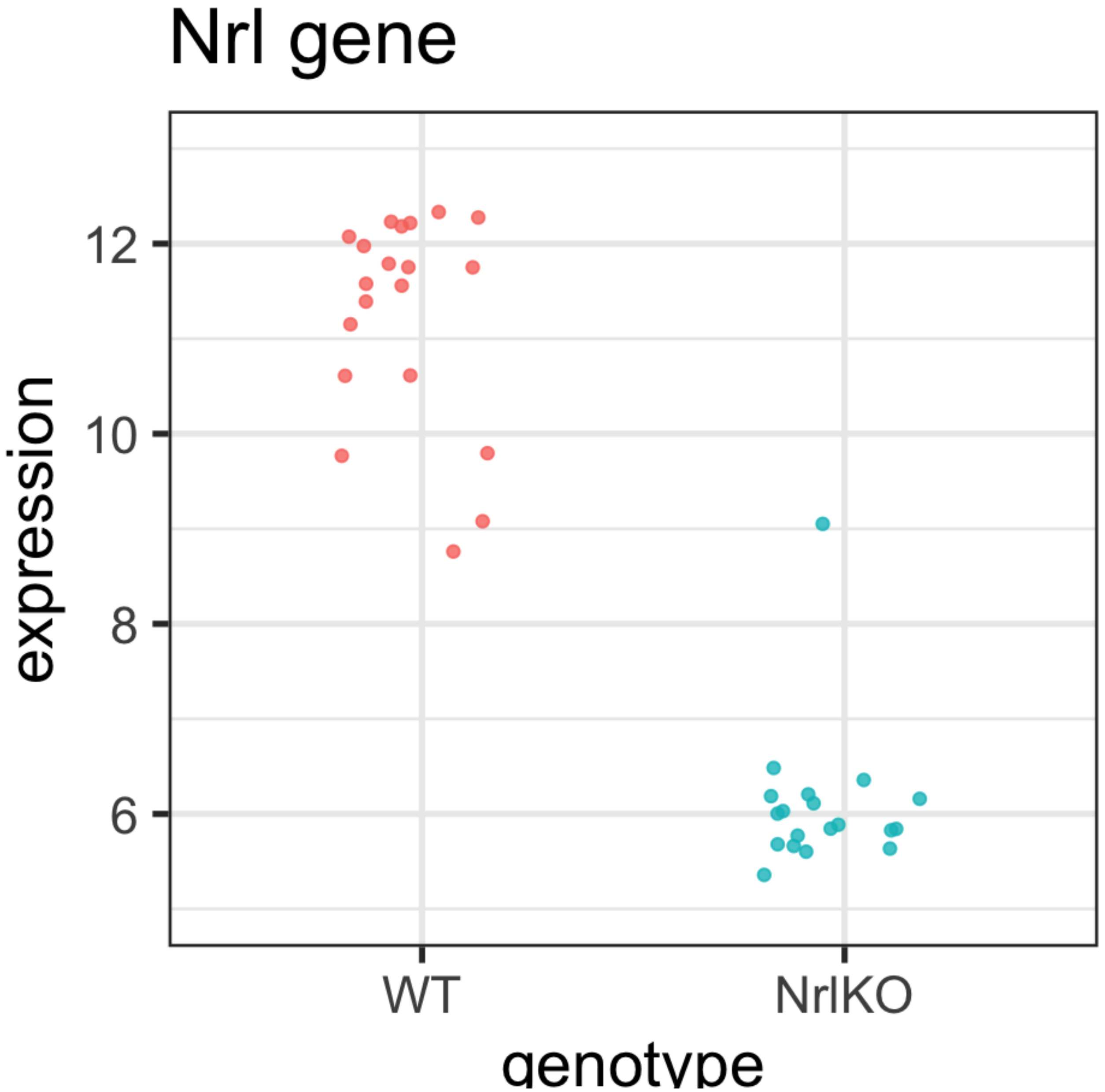
▶ Code

```
# A tibble: 2 × 5
  term          estimate std.error statistic  p.value
  <chr>            <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)     7.77      0.0344   226.    1.10e-59
2 genotypeNrlKO  -0.0261    0.0493    -0.529 6.00e- 1
```

We still haven’t answered our question… where’s the line??

$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij}; \quad \tau_1 = 0, \quad \varepsilon_{ij} \sim G; \quad E[\varepsilon_{ij}] = 0;$$


Irs4 gene


Nrl gene

Indicator variables

Let’s re-write our model using **indicator** (aka ‘dummy’) variables:

$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij} \quad \text{where} \quad \tau_1 = 0; \quad \varepsilon_{ij} \sim G; \quad E[\varepsilon_{ij}] = 0;$$

$$\downarrow$$

$$Y_{ij} = \theta + \tau x_{ij} + \varepsilon_{ij} \quad \text{where} \quad x_{ij} = \begin{cases} 1 \text{ if } j = 2 \\ 0 \text{ otherwise} \end{cases}$$

ⓘ **Note**

Note that $Y_{i1} = \theta + \varepsilon_{i1}$, because $x_{i1} = 0$ and $Y_{i2} = \theta + \tau + \varepsilon_{i2}$, because $x_{i2} = 1$ (for all i)

The second form is written as a *linear* ($y = a + bx + \varepsilon$) regression model, with a (**indicator**) explanatory variable x_{ij}

t-test with a linear model

ⓘ Note

Using indicator variables to model our categorical variable `genotype`, we can perform a **2-sample t-test** with a linear model

$$Y_{ij} = \theta + \tau x_{ij} + \varepsilon_{ij} \text{ where } x_{ij} = \begin{cases} 1 \text{ if } j = 2 \\ 0 \text{ if } j = 1 \end{cases}$$

- The standalone t-test is carried out on $H_0 : \mu_1 = \mu_2$
- The t-test in the linear model is carried out on $H_0 : \tau = 0$, where τ is the difference in population means (here NrlKO - WT)
- Recall that $\tau = \mu_2 - \mu_1$ - this is why these are equivalent tests!

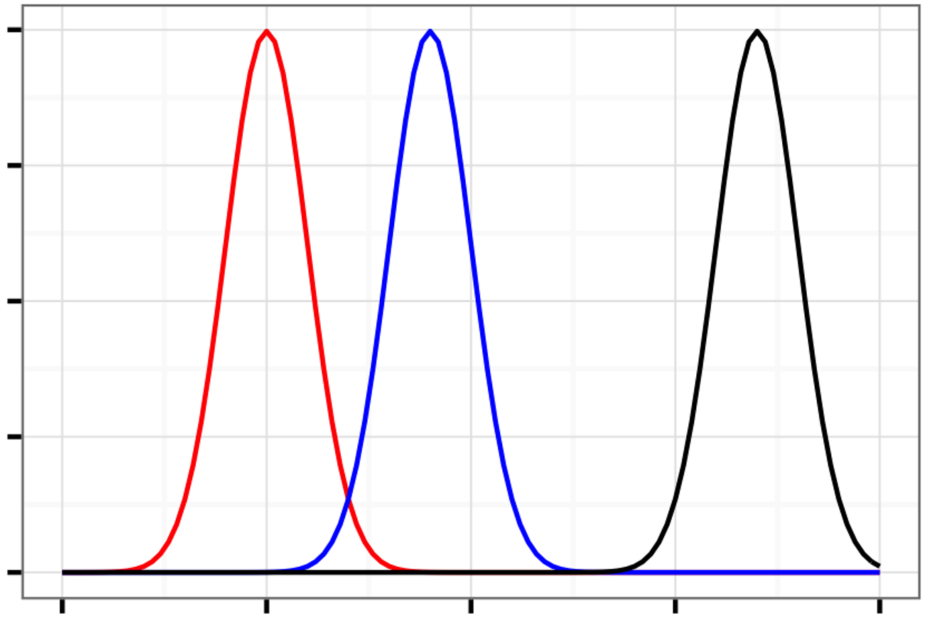
Beyond 2-group comparisons

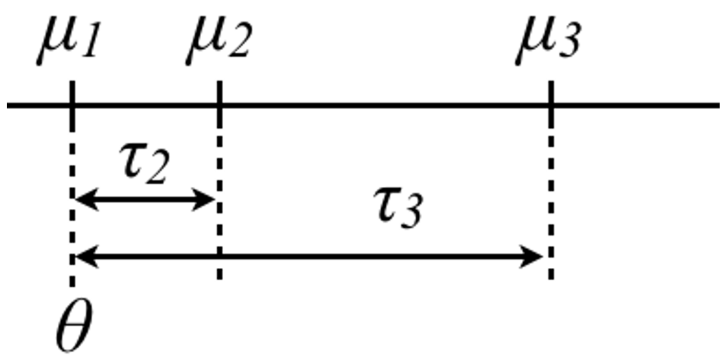
"cell-means"

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

"reference-treatments"

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





More than 2 groups!

Note

Indicator variables can be used to model one *or more* categorical variables, each with 2 *or more* levels!

2-sample *t*-test using a linear model

$$Y_{ij} = \theta + \tau x_{ij} + \varepsilon_{ij} \quad \text{where} \quad x_{ij} = \begin{cases} 1 \text{ if } j = 2 \\ 0 \text{ if } j = 1 \end{cases}$$

1-way ANOVA with many levels[1] using a linear model - e.g for 3 groups:

$$Y_{ij} = \theta + \tau_2 x_{ij2} + \tau_3 x_{ij3} + \varepsilon_{ij} \quad \text{where } x_{ij2} = \begin{cases} 1 \text{ if } j = 2 \\ 0 \text{ otherwise} \end{cases} \quad \text{and} \quad x_{ij3} = \begin{cases} 1 \text{ if } j = 3 \\ 0 \text{ otherwise} \end{cases}$$

Important

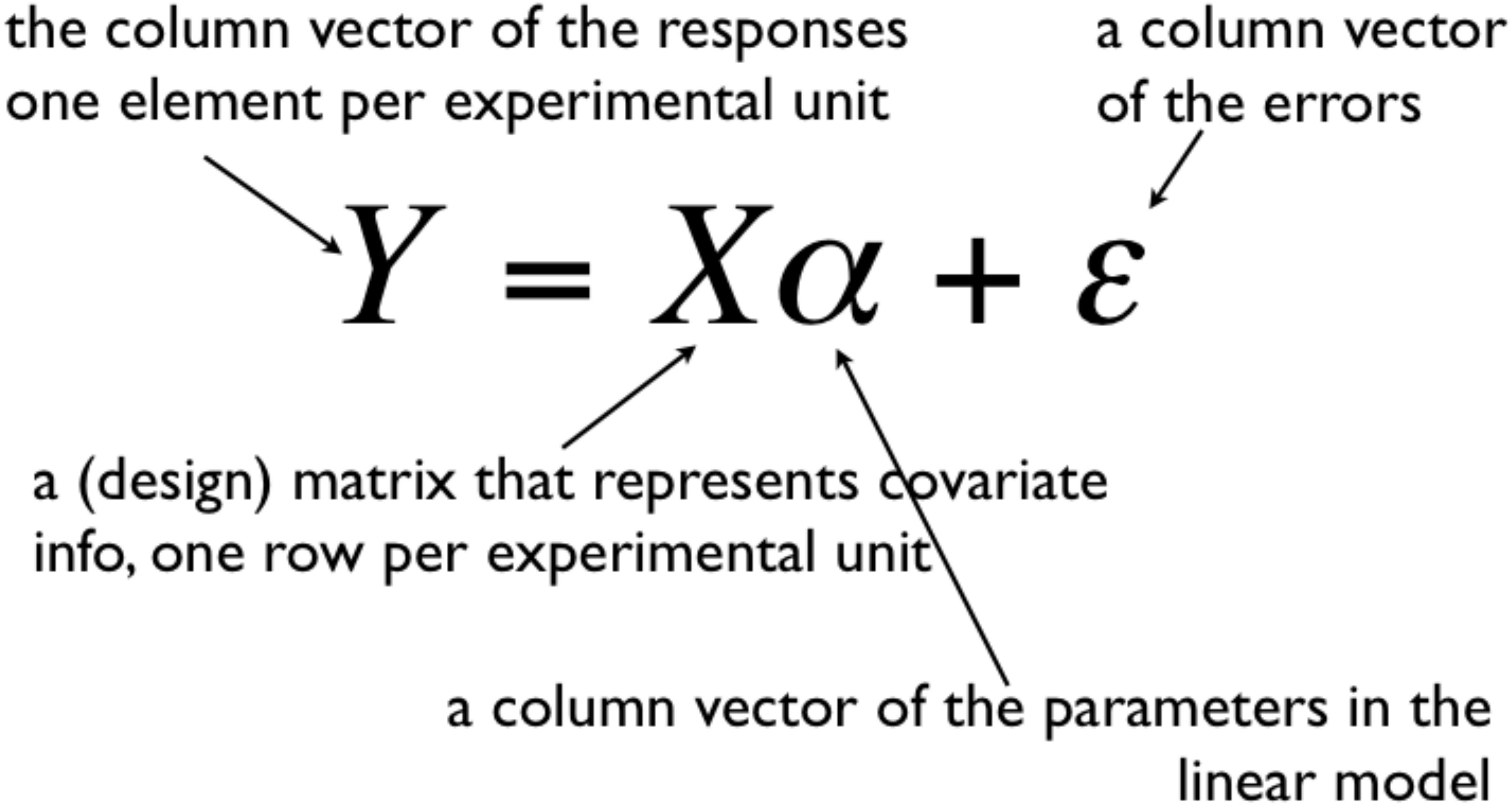
This equivalence is why R can estimate all of them with `lm()`

[1] in general: yet *another* parameterization can be used to present ANOVA

Connections

- The **t-test** is a special case of **ANOVA**, but with ANOVA you can compare **more than two groups** and **more than one factor**.
- **ANOVA** is a special case of **linear regression**, but with linear regression you can include **quantitative variables** in the model.
- **Linear regression** provides a unifying framework to model the association between a response and **many quantitative and qualitative variables**.
- **In R** all three can be computed using the `lm()` function.

Linear models using matrix notation



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

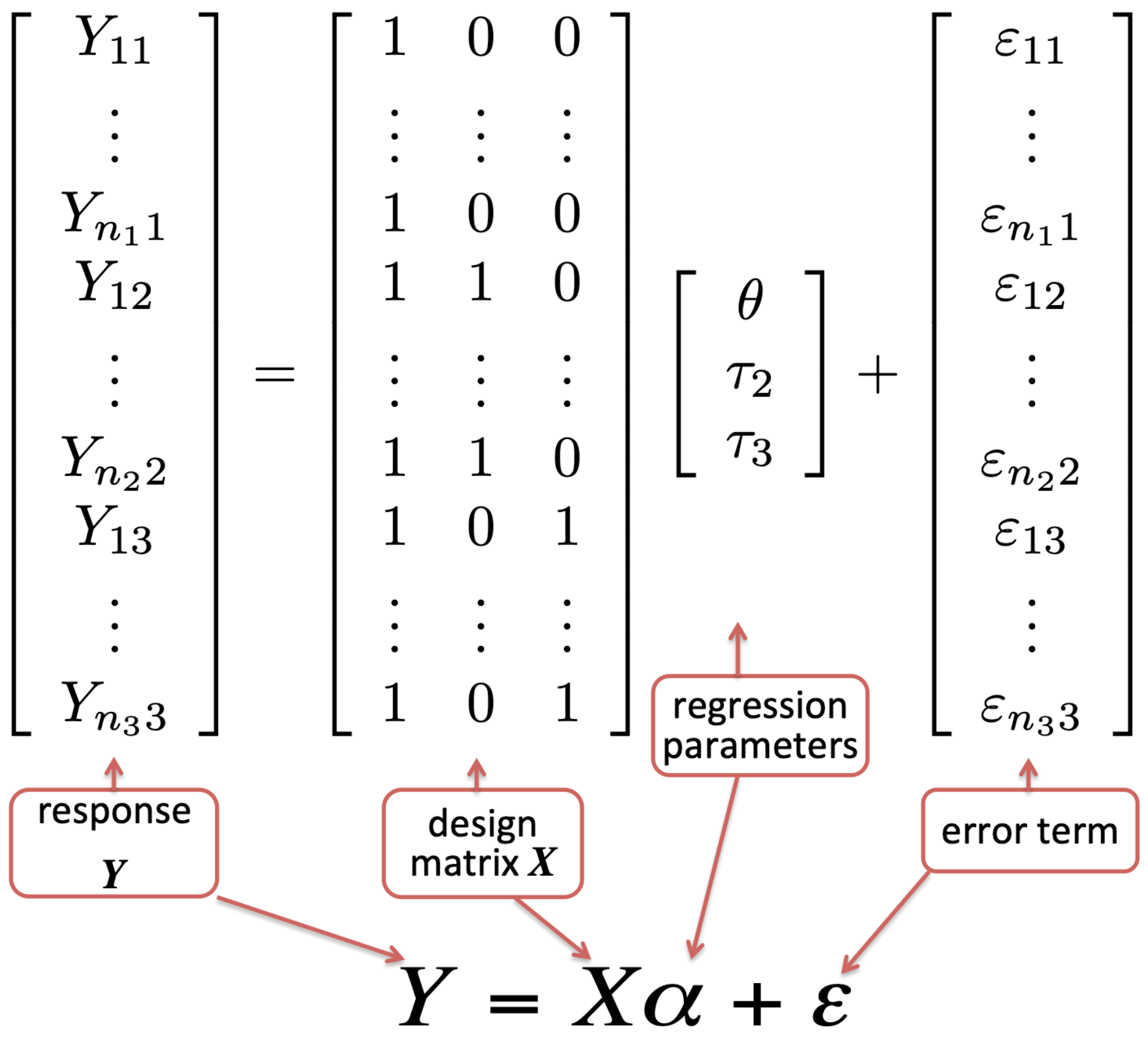
a column vector of the parameters in the linear model

It will become handy to write our model using matrix notation

Design matrix

Let's form a **design matrix** (X) for a 3-group comparison

$$Y_{ij} = \theta + \tau_2 x_{ij2} + \tau_3 x_{ij3} + \varepsilon_{ij}$$



$$\begin{bmatrix} Y_{11} \\ \vdots \\ Y_{n_1 1} \\ Y_{12} \\ \vdots \\ Y_{n_2 2} \\ Y_{13} \\ \vdots \\ Y_{n_3 3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{n_1 1} \\ \varepsilon_{12} \\ \vdots \\ \varepsilon_{n_2 2} \\ \varepsilon_{13} \\ \vdots \\ \varepsilon_{n_3 3} \end{bmatrix}$$

response $\boldsymbol{Y}$

design matrix $\boldsymbol{X}$

regression parameters

error term

$$\boldsymbol{Y} = \boldsymbol{X\alpha} + \boldsymbol{\varepsilon}$$

First column in X for reference treatment parameterization is all 1s

Second & third columns contain x_{ij2} and x_{ij3}:

- $x_{i12} = x_{i13} = 0$ for the reference group
- $x_{i22} = 1$ for the 2nd group
- $x_{i33} = 1$ for the 3rd group

$$\left[\begin{array}{c} Y_{11} \\ \hline \vdots \\ Y_{n_1 1} \\ Y_{12} \\ \hline \vdots \\ Y_{n_2 2} \\ Y_{13} \\ \hline \vdots \\ Y_{n_3 3} \end{array}\right] = \left[\begin{array}{ccc} 1 & 0 & 0 \\ \hline \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ \hline \vdots & \vdots & \vdots \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ \hline \vdots & \vdots & \vdots \\ 1 & 0 & 1 \end{array}\right] \left[\begin{array}{c} \theta \\ \tau_2 \\ \tau_3 \end{array}\right] + \left[\begin{array}{c} \varepsilon_{11} \\ \hline \vdots \\ \varepsilon_{n_1 1} \\ \varepsilon_{12} \\ \hline \vdots \\ \varepsilon_{n_2 2} \\ \varepsilon_{13} \\ \hline \vdots \\ \varepsilon_{n_3 3} \end{array}\right]$$

$$Y_{i1} = 1 \times \theta + 0 \times \tau_2 + 0 \times \tau_3 + \varepsilon_{i1} = \theta + \varepsilon_{i1}$$

$$Y_{i2} = 1 \times \theta + 1 \times \tau_2 + 0 \times \tau_3 + \varepsilon_{i2} = \theta + \tau_2 + \varepsilon_{i2}$$

$$Y_{i3} = 1 \times \theta + 0 \times \tau_2 + 1 \times \tau_3 + \varepsilon_{i3} = \theta + \tau_3 + \varepsilon_{i3}$$

$$Y_{ij} = \theta + \tau_2 x_{ij2} + \tau_3 x_{ij3} + \varepsilon_{ij}$$

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

$$\begin{bmatrix} Y_{11} \\ Y_{21} \\ \vdots \\ Y_{n_3 3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \varepsilon_{n_3 3} \end{bmatrix}$$

Reference group: μ_1

$\mu_2 - \mu_1$

$\mu_3 - \mu_1$

The model is still written with a reference-treatment parameterization (difference of means)

$$E[Y_{i1}] = \theta$$

$$E[Y_{i2}] = \theta + \tau_2 \quad \rightarrow \tau_2 = E[Y_{i2}] - E[Y_{i1}] = \mu_2 - \mu_1$$

$$E[Y_{i3}] = \theta + \tau_3 \quad \rightarrow \tau_3 = E[Y_{i3}] - E[Y_{i1}] = \mu_3 - \mu_1$$

Linear[1] regression can include *quantitative* & *qualitative* covariates

$$Y = \boxed{X}\alpha + \varepsilon$$

This gives us a VERY FLEXIBLE framework!!

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1.22 \\ 1 & 2.02 \\ 1 & 1.42 \\ \vdots & \vdots \\ 1 & 1.89 \\ 1 & 2.01 \\ \vdots & \vdots \\ 1 & 1.56 \\ 1 & 2.17 \\ 1 & 1.51 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1.22 & 0 \\ 1 & 0 & 2.02 & 0 \\ 1 & 0 & 1.42 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1.89 & 0 \\ 1 & 1 & 2.01 & 2.01 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1.56 & 1.56 \\ 1 & 1 & 2.17 & 2.17 \\ 1 & 1 & 1.51 & 1.51 \end{bmatrix}$$

1 categorical covariate

2 categorical covariates

1 continuous covariate

2 continuous
1 categorical

AND MANY MORE .....

Tip: ?model.matrix

[1] but you can change that!

How it works in practice using `lm()` in R

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

↓

```r
lm(y ~ x, data = yourData)
```

`y ~ x`: formula

`y`: numeric

`x`: numeric and/or factor

`yourData`: `data.frame` (or `tibble`) in which `x` and `y` are to be found

By default, R uses the reference-treatment parameterization[1]

[1] but you can change that!

`factor` class in R

Mathematically, the design matrix X in $Y = X\alpha + \varepsilon$ needs to be a numeric matrix

Important

- If your data contains categorical variables (e.g., `genotype`), you need to set them as **factors**
  - especially important if your categorical variables are encoded numerically!!
  - `lm` will automatically treat character variables as factors)
- R creates appropriate indicator variables (numeric) for factors!

```r
str(twoGenes$genotype)
```

```
Factor w/ 2 levels "WT","NrlKO": 2 2 2 2 2 2 2 2 2 2 ...
```

Under the hood, R creates a numeric X

```r
# create design matrix
mm <- model.matrix( ~ genotype, data = twoGenes)

# show first 3 and last 3 rows of model.matrix
head(mm, 3)
```

```
  (Intercept) genotypeNrlKO
1           1             1
2           1             1
3           1             1
```

```r
tail(mm, 3)
```

```
   (Intercept) genotypeNrlKO
76           1             0
77           1             0
78           1             0
```

```r
# show first 3 and last 3 values of genotype
twoGenes %>%
  slice(c(1:3, (n()-3):n())) %>%
  pull(genotype)
```

```
[1] NrlKO NrlKO NrlKO WT    WT    WT    WT   
Levels: WT NrlKO
```

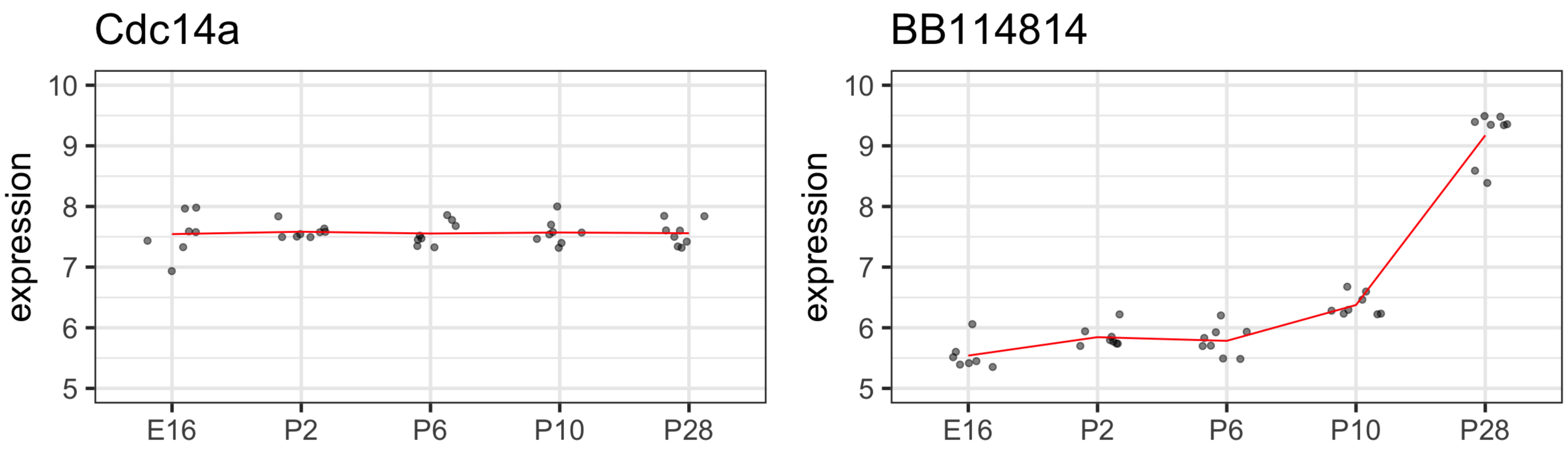
Beyond 2-group comparisons in our case study

Biological question

Is the expression of gene X the same at all developmental stages?

$$H_0 : \mu_{E16} = \mu_{P2} = \mu_{P6} = \mu_{P10} = \mu_{P28}$$

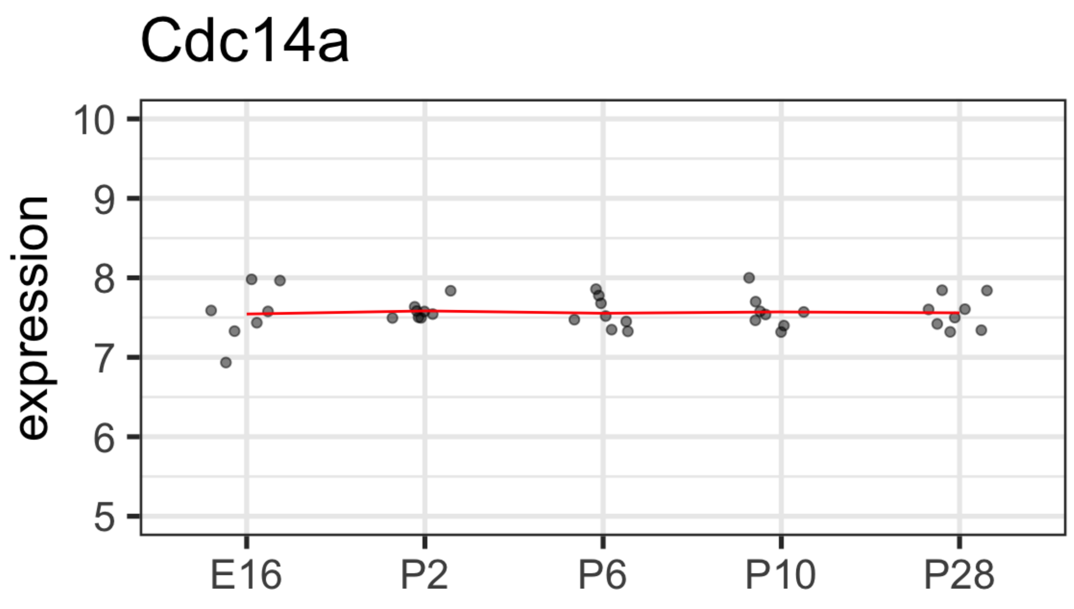
Let’s look at another two genes for some variety

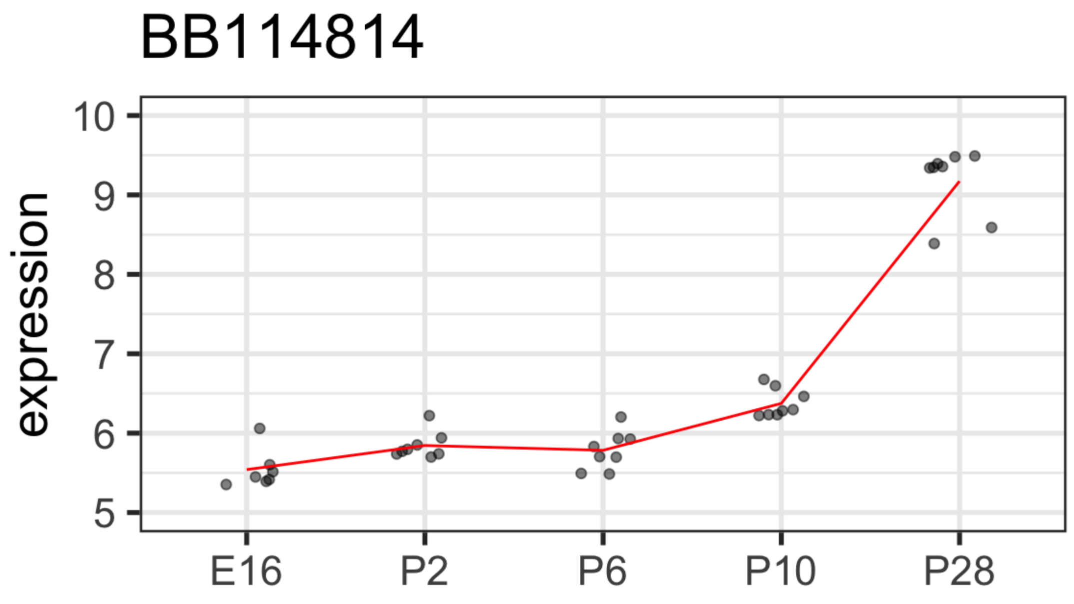


The sample means: $\hat{\mu}_{E16}, \; \hat{\mu}_{P2}, \; \hat{\mu}_{P6}, \; \hat{\mu}_{P10}, \; \hat{\mu}_{P28}$

```
twoGenes %>%
  group_by(gene, dev_stage) %>%
  summarize(meanExpr = mean(expression)) %>%
  pivot_wider(values_from = meanExpr, names_from = gene
```

```
# A tibble: 5 × 3
  dev_stage BB114814 Cdc14a
  <fct>        <dbl>  <dbl>
1 E16           5.54   7.54
2 P2            5.84   7.58
3 P6            5.78   7.55
4 P10           6.38   7.57
5 P28           9.17   7.56
```





BB114814 gene with notable time effect

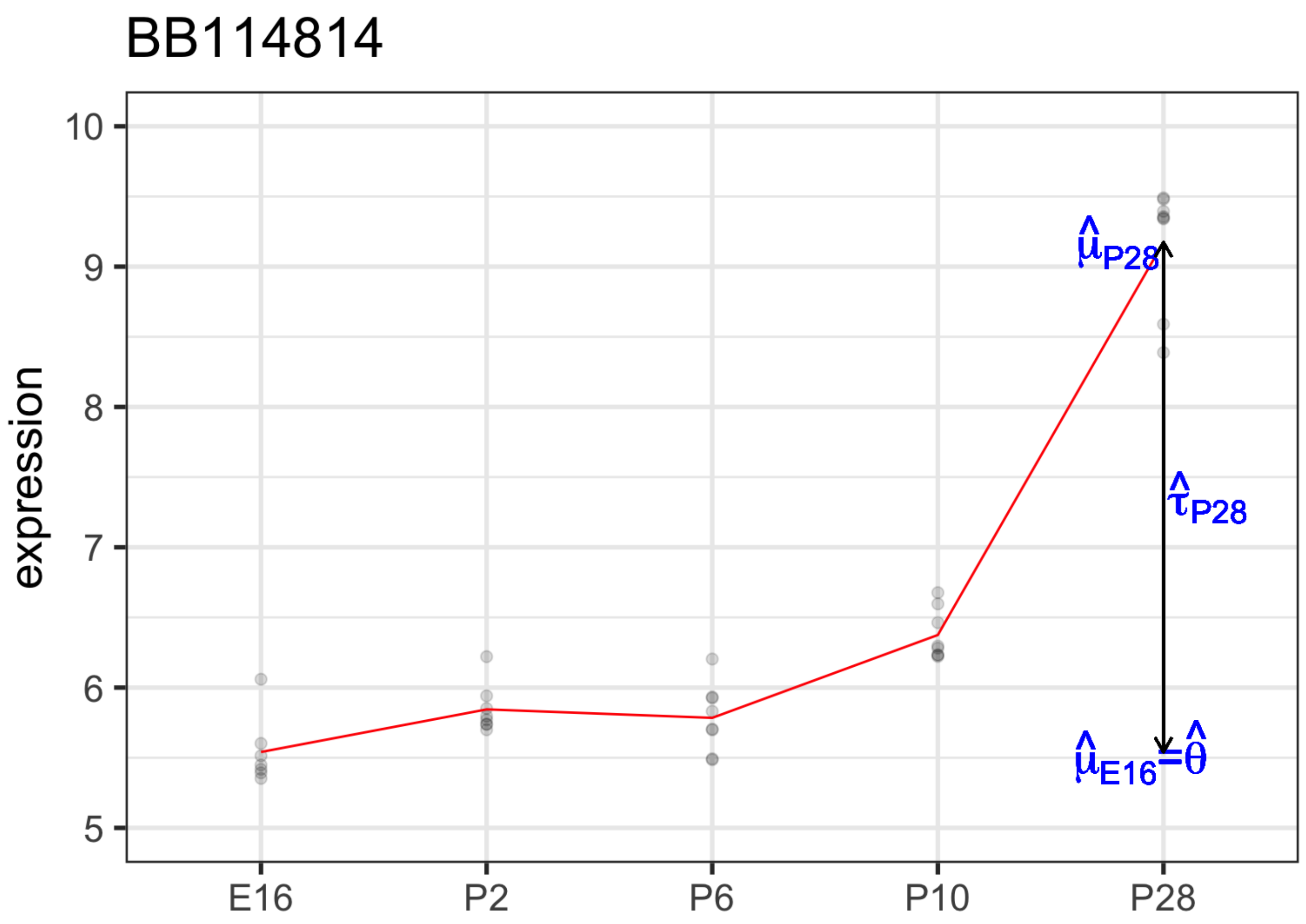
```r
twoGenes %>% filter(gene == "BB114814") %>%
  group_by(dev_stage) %>%
  summarize(cellMeans = mean(expression)) %>%
  mutate(timeEffect = cellMeans - cellMeans[1])
```

```
# A tibble: 5 × 3
  dev_stage cellMeans timeEffect
  <fct>         <dbl>      <dbl>
1 E16            5.54      0
2 P2             5.84      0.304
3 P6             5.78      0.243
4 P10            6.38      0.834
5 P28            9.17      3.63
```

“Effect” here is relative to reference/baseline (E16)

BB114814 gene with notable time effect

```
# A tibble: 5 × 3
  dev_stage cellMeans timeEffect
  <fct>         <dbl>      <dbl>
1 E16            5.54      0
2 P2             5.84      0.304
3 P6             5.78      0.243
4 P10            6.38      0.834
5 P28            9.17      3.63
```



Check your understanding

Can you guess the size of the X matrix needed to test for any time differences? How many indicator variables do we need?

Gene BB114814 with notable time effect

We need ___ indicator variables to estimate and test ___ time differences (between ___ time points):

Mathematically:

$$Y_{ij} = \theta + \tau_{P2} x_{ijP2} + \tau_{P6} x_{ijP6} + \tau_{P10} x_{ijP10} + \tau_{P28} x_{ijP28} + \varepsilon_{ij}$$

Notation: x_{ijk}:

- i indexes for the observation/sample within group
- j indexes the group (here: level of `dev_stage`)
- k is the name of the indicator variable

Under the hood, R creates a numeric X

```
str(twoGenes)
```

```
tibble [78 × 5] (S3: tbl_df/tbl/data.frame)
 $ gene      : chr [1:78] "BB114814" "BB114814" "BB114814" "BB114814" ...
 $ sample_id : chr [1:78] "GSM92610" "GSM92611" "GSM92612" "GSM92613" ...
 $ expression: num [1:78] 8.59 8.39 9.34 9.49 5.39 ...
 $ dev_stage : Factor w/ 5 levels "E16","P2","P6",..: 5 5 5 5 1 1 1 4 4 4 ...
 $ genotype  : Factor w/ 2 levels "WT","NrlKO": 2 2 2 2 2 2 2 2 2 2 ...
```

```
model.matrix( ~ dev_stage, data = twoGenes)
```

```
   (Intercept) dev_stageP2 dev_stageP6 dev_stageP10 dev_stageP28
1            1           0           0            0            1
2            1           0           0            0            1
3            1           0           0            0            1
4            1           0           0            0            1
5            1           0           0            0            0
6            1           0           0            0            0
7            1           0           0            0            0
8            1           0           0            1            0
9            1           0           0            1            0
10           1           0           0            1            0
11           1           0           0            1            0
12           1           1           0            0            0
13           1           1           0            0            0
14           1           1           0            0            0
```

Hypothesis tests in `lm` output

```
# A tibble: 5 × 3
  dev_stage cellMeans timeEffect
  <fct>         <dbl>      <dbl>
1 E16            5.54      0
2 P2             5.84      0.304
3 P6             5.78      0.243
4 P10            6.38      0.834
5 P28            9.17      3.63
```

```r
twoGenes %>% filter(gene == "BB114814") %>%
  lm(expression ~ dev_stage, data = .) %>% tidy()
```

```
# A tibble: 5 × 5
  term         estimate std.error statistic  p.value
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)     5.54      0.102     54.2  1.31e-34
2 dev_stageP2     0.304     0.140      2.17 3.69e- 2
3 dev_stageP6     0.243     0.140      1.74 9.11e- 2
4 dev_stageP10    0.834     0.140      5.96 9.62e- 7
5 dev_stageP28    3.63      0.140     26.0  5.30e-24
```

$H_0 : \theta = 0$ or $H_0 : \mu_{E16} = 0$

Estimate: $\hat{\theta} = \hat{\mu}_{E16} = \bar{Y}_{\cdot E16}$

> we are not usually interested in testing this hypothesis: baseline mean = 0

Hypothesis tests in `lm` output

```
# A tibble: 5 × 3
  dev_stage cellMeans timeEffect
  <fct>         <dbl>      <dbl>
1 E16            5.54      0
2 P2             5.84      0.304
3 P6             5.78      0.243
4 P10            6.38      0.834
5 P28            9.17      3.63
```

```r
twoGenes %>% filter(gene == "BB114814") %>%
  lm(expression ~ dev_stage, data = .) %>% tidy()
```

```
# A tibble: 5 × 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    5.54      0.102     54.2  1.31e-34
2 dev_stageP2    0.304     0.140      2.17 3.69e- 2
3 dev_stageP6    0.243     0.140      1.74 9.11e- 2
4 dev_stageP10   0.834     0.140      5.96 9.62e- 7
5 dev_stageP28   3.63      0.140     26.0  5.30e-24
```

$H_0 : \tau_{P2} = 0$ or $H_0 : \mu_{P2} = \mu_{E16}$

Estimate: $\hat{\tau}_{P2} = \hat{\mu}_{P2} - \hat{\mu}_{E16} = \bar{Y}_{\cdot P2} - \bar{Y}_{\cdot E16}$

we *are* usually interested in testing this hypothesis: change from E16 to 2 days old = 0

Hypothesis tests in lm output

```
# A tibble: 5 × 3
  dev_stage cellMeans timeEffect
  <fct>         <dbl>      <dbl>
1 E16            5.54      0    
2 P2             5.84      0.304
3 P6             5.78      0.243
4 P10            6.38      0.834
5 P28            9.17      3.63 
```

```r
twoGenes %>% filter(gene == "BB114814") %>%
  lm(expression ~ dev_stage, data = .) %>%
  tidy()
```

```
# A tibble: 5 × 5
  term         estimate std.error statistic  p.value
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)     5.54      0.102     54.2  1.31e-34
2 dev_stageP2     0.304     0.140      2.17 3.69e- 2
3 dev_stageP6     0.243     0.140      1.74 9.11e- 2
4 dev_stageP10    0.834     0.140      5.96 9.62e- 7
5 dev_stageP28    3.63      0.140     26.0  5.30e-24
```

$H_0 : \tau_{P28} = 0$ or $H_0 : \mu_{P28} = \mu_{E16}$

Estimate:

$$\hat{\tau}_{P28} = \hat{\mu}_{P28} - \hat{\mu}_{E16} = \bar{Y}_{\cdot P28} - \bar{Y}_{\cdot E16}$$

we *are* usually interested in testing this hypothesis: change from E16 to 4 weeks old = 0

Notice the standard error estimates

▶ Code

```
              Estimate Std. Error   t value     Pr(>|t|)
(Intercept)  5.5409162  0.1021560 54.239748 1.314828e-34
dev_stageP2  0.3037855  0.1398829  2.171713 3.694652e-02
dev_stageP6  0.2432795  0.1398829  1.739166 9.105366e-02
dev_stageP10 0.8341163  0.1398829  5.962962 9.620151e-07
dev_stageP28 3.6323772  0.1398829 25.967276 5.303201e-24
```

All data points are used to estimate the variance of the error term for the indicator variables

Two types of null hypotheses: single vs joint

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

$$\alpha = (\theta, \tau_{P2}, \tau_{P6}, \tau_{P10}, \tau_{P28})$$

$H_0 : \tau_j = 0$ vs $H_0 : \tau_j \neq 0$

for each *j* individually

For example: Is gene *A* differentially expressed 2 days after birth (compared to embryonic day 16)?

$$H_0 : \tau_{P2} = 0$$

Note

This single hypothesis can be tested with a **t-test**

$H_0 : \tau_j = 0$ vs $H_0 : \tau_j \neq 0$

for all *j* at the same time

For example: Is gene *A* significantly affected by time? In other words, is gene *A* differentially expressed at *any* time point?

$$H_0 : \tau_{P2} = \tau_{P6} = \tau_{P10} = \tau_{P28} = 0$$

Key Question

How do we test this joint hypothesis?

F-test and overall significance of one or more coefficients

- the *t*-test in linear regression allows us to test single hypotheses:

$$H_0 : \tau_j = 0$$

$$H_A : \tau_j \neq 0$$

- but we often like to test multiple hypotheses *simultaneously*:

$$H_0 : \tau_{P2} = \tau_{P6} = \tau_{P10} = \tau_{P28} = 0 \text{ [AND statement]}$$

$$H_A : \tau_j \neq 0 \text{ for some j [OR statement]}$$

- the ***F*-test** allows us to test such compound tests
  - more on this type of test next week

Single and joint tests in `lm` output

Can you locate the results of each type of test in the `lm` output?

$H_0 : \tau_j = 0$ vs $H_0 : \tau_j \neq 0$ for each j **individually**

$H_0 : \tau_j = 0$ vs $H_0 : \tau_j \neq 0$ for all j **together**

```
twoGenes %>% filter(gene == "BB114814") %>%
  lm(expression ~ dev_stage, data = .) %>%
  summary()
```

```
Call:
lm(formula = expression ~ dev_stage, data = .)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.78553 -0.13324 -0.04796  0.17038  0.51846 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)    5.5409     0.1022  54.240  < 2e-16 ***
dev_stageP2    0.3038     0.1399   2.172   0.0369 *  
dev_stageP6    0.2433     0.1399   1.739   0.0911 .  
dev_stageP10   0.8341     0.1399   5.963 9.62e-07 ***
dev_stageP28   3.6324     0.1399  25.967  < 2e-16 ***
```

To conclude

1. We can compare group means (2 or more) using a linear model
2. We can use different parameterizations (**cell means** and **reference-treatment effect**) to write statistical models
3. We can write a **linear model** using matrix notation: $Y = X\alpha + \varepsilon$
4. Linear models can include **quantitative & qualitative covariates**
5. We use different tests to distinguish between **single** and **joint** hypotheses (e.g. t-tests vs F-tests)