

# Lecture 6 – ANOVA and Linear Models

STAT/BIOF/GSAT 540: Statistical Methods for High Dimensional Biology

Keegan Korthauer

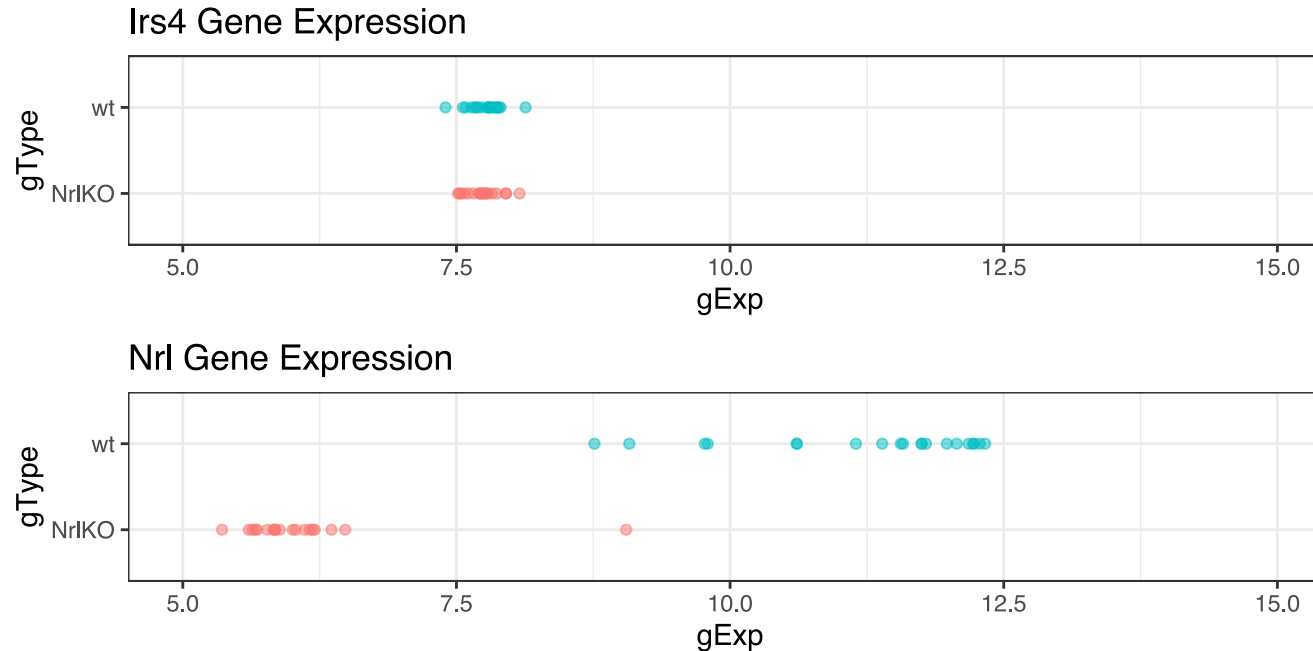
2020/01/22

Slides by: Gabriela Cohen Freue with contributions from Jenny Bryan and Keegan Korthauer

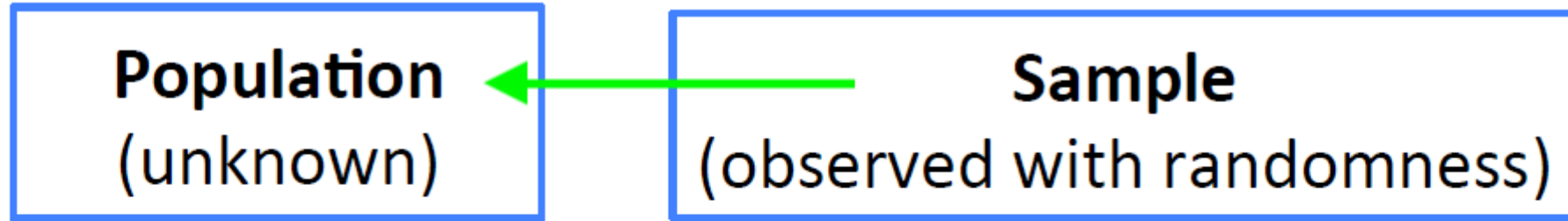
Are these genes truly different in NrlKO compared to WT?

$H_0$ : the expression level of gene A is the same in both conditions.

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



**Statistics:** use a random sample to learn about the population



$$Y \sim F$$

$$Y_1, Y_2, \dots, Y_{n_Y}$$

$$Z \sim G$$

$$Z_1, Z_2, \dots, Z_{n_Z}$$

$$E[Y] = \mu_Y$$

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

$$E[Z] = \mu_Z$$

$$\hat{\mu}_Z = \bar{Z} = \frac{\sum_{i=1}^{n_Z} Z_i}{n_Z}$$

$$H_0: \mu_Y = \mu_Z$$

$$\bar{Y} = \bar{Z}$$

# Last class: hypothesis testing

1. Define a **test statistic** to test  $H_0$

- 2-sample  $t$ -test
- Welch  $t$ -test
- Wilcoxon rank-sum test
- Kolmogorov-Smirnov test

2. Compute the **observed value** for the test statistic

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 value (alpha, significance level)

# We can run these tests in R

Example: use the `t.test` function to test  $H_0$  using a classical 2-sample  $t$ -test.

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

```
##  
##      Two Sample t-test  
##  
## data:  gExp by gType  
## t = -0.52865, df = 37, p-value = 0.6002  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
##  -0.12597002  0.07383844  
## sample estimates:  
## mean in group NrlK0      mean in group wt  
##           7.739684           7.765750
```

# Today...

- show how to compare means of different groups (2 or more) using a linear regression model
  - dummy variables to model the levels of a qualitative explanatory variable
- write a linear model using matrix notation
  - understand which matrix is built by R
- distinguish between conditional and marginal effects
  - $t$ -tests vs  $F$ -tests

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

two sample t test

$$H_0 : \mu_1 = \mu_2$$

```
> 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

It seems that we can use any of these methods to test  $H_0$

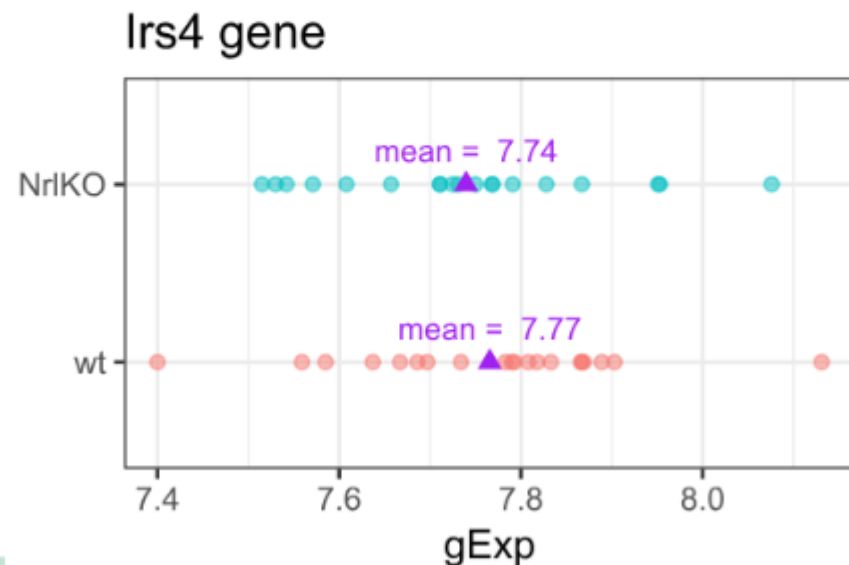
```
> t.test(gExp ~ gType, miniDat,  
+        subset = gene == "Irs4", var.equal = TRUE)
```

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
```

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



$$7.739684 - 7.765750 = -0.026066$$

$$-0.5286494^2 = 0.2794702$$



# *t*-test vs linear regression: why the same results?

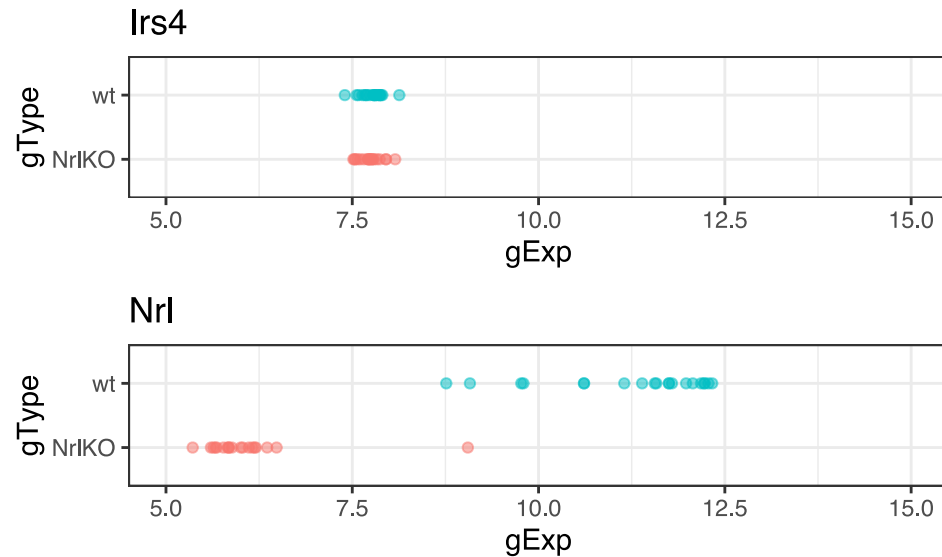
```
irs4Dat <- subset(miniDat, gene=="Irs4")  
ttest.irs4 <- t.test(gExp ~ gType, irs4Dat, var.equal = TRUE)  
list("t value"=ttest.irs4$stat, "p-value"=ttest.irs4$p.value)
```

```
## $t value  
##          t  
## -0.5286494  
##  
## $p-value  
## [1] 0.6002058
```

```
lm.irs4 <- summary(lm(gExp ~ gType, irs4Dat))  
list("t value"=lm.irs4$coeff[2,3], "p-value"=lm.irs4$coeff[2,4])
```

```
## $t value  
## [1] 0.5286494  
##  
## $p-value  
## [1] 0.6002058
```

## $t$ -test *vs* linear regression: where's the line?



Note that the  $y$ -axis in these plots is not numerical, thus a line in this space does not have any mathematical meaning.

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

# 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 a subindex to distinguish observations from each group, i.e.,

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

where  $j = \{\text{wt}, \text{NrlKO}\}$  or  $j = \{1, 2\}$  identifies the groups; and  $i = 1, \dots, n_j$  identifies the observations within each group

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

The goal is to test

$$H_0 : \mu_1 = \mu_2$$

using data from the model

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

where  $j = \{\text{wt, NrlKO}\}$  or  $j = \{1, 2\}$ ; and  $i = 1, \dots, n_j$ .

■ For simplicity, we assume a common distribution  $G$  for all groups

Note that the population means are given by  $E[Y_{ij}] = \mu_j$ , i.e., the model is written with a **cell-means** -  $\mu_j$  - parametrization

Note that for each group, the **population** mean is given by

$$E[Y_{ij}] = \mu_j,$$

A natural **estimator** of the population mean is the **sample mean**

Classical hypothesis testing methods use the group sample means as estimators

See, for example, the `t.test` function in R:

```
ttest.irs4$estimate
```

```
## mean in group Nr1K0      mean in group wt  
##           7.739684           7.765750
```

However, the `lm` function reports other estimates, **why?**

```
(means.irs4 <- as.data.frame(irs4Dat %>% group_by(gType) %>%  
  summarize(meanGroups = mean(gExp, digits = 6))))
```

```
##   gType meanGroups  
## 1 NrlKO    7.739684  
## 2   wt     7.765750
```

```
lm.irs4$coefficients[,1]
```

```
## (Intercept)      gTypewt  
## 7.73968421  0.02606579
```



(Intercept) is the **sample mean** of NrlKO group

but gTypewt is **not** the sample mean of the WT group

Parametrizations: which parameters we use to write the model?

By default, the `lm` does not use the cell-means parametrization. The goal is to *compare* the means, not to study each in isolation.

From **cell-means** -  $\mu_j$ :

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

↓

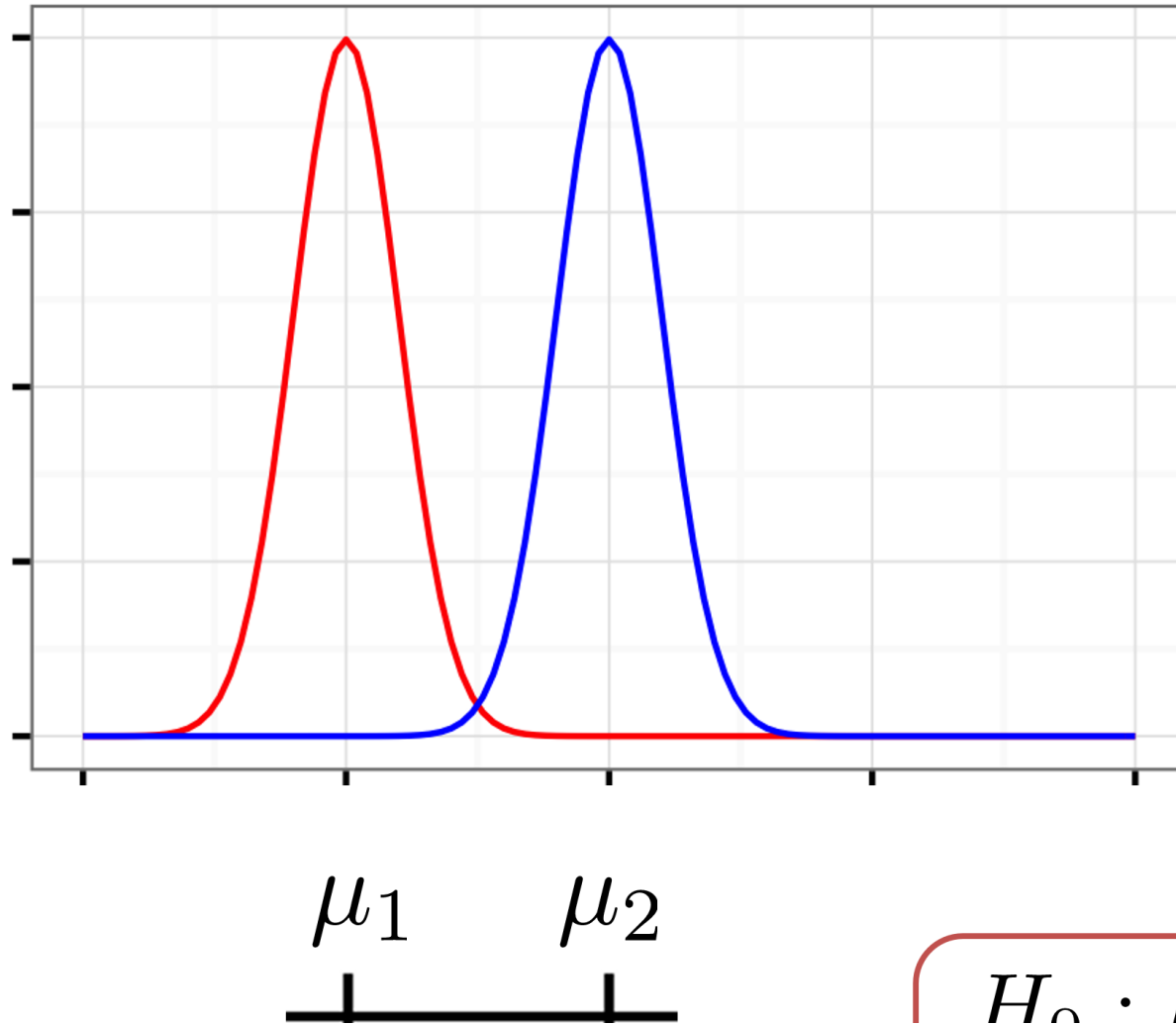
to **reference-treatment effect** -  $(\theta, \tau_j)$ :

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

Note that for each group, the population mean is given by

$$E[Y_{ij}] = \theta + \tau_j = \mu_j,$$

# Relation between parametrizations





lm reports the sample mean of the **reference** group (Nr1KO):  $\hat{\theta}$

and the **treatment effect**, i.e., difference between the sample means of both groups:  $\hat{\tau}_2$

```
lm.irs4$coefficients[, 1]
```

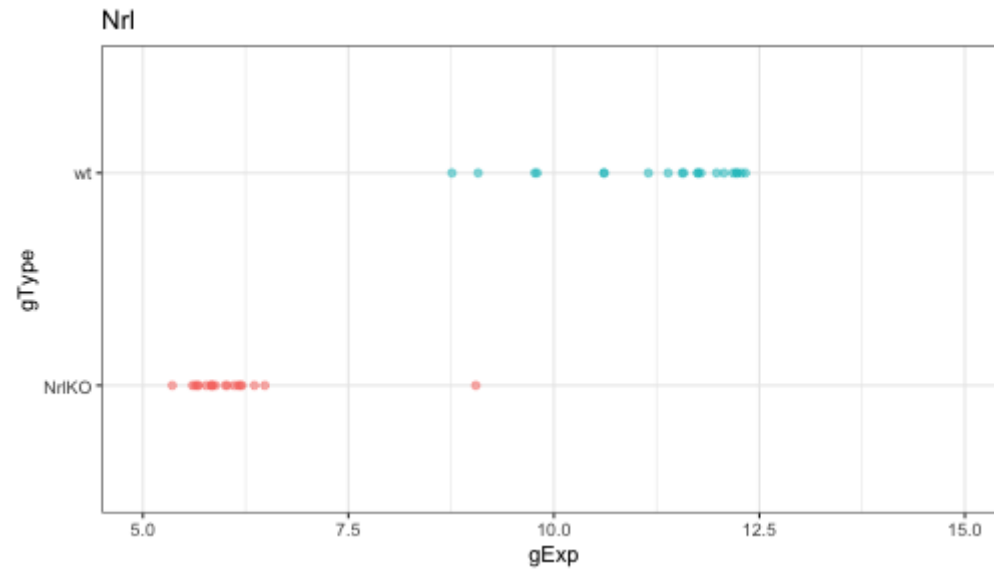
```
## (Intercept)      gTypewt  
## 7.73968421 0.02606579
```

```
data.frame(meanWT = means.irs4[1, 2],  
           meanDiff = diff(means.irs4$meanGroups))
```

```
##      meanWT  meanDiff  
## 1 7.739684 0.02606579
```

We still haven't answered ... where's the line??

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



# Dummy variables

Let's re-write our model using **dummy** (or indicator) variables:

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

↓

$$Y_{ij} = \theta + \tau_2 \times x_{ij} + \varepsilon_{ij}; \quad x_{ij} = \begin{cases} 1 & \text{if } j = 2 \\ 0 & \text{otherwise} \end{cases}$$

Note that  $Y_{i1} = \theta + \varepsilon_{i1}$ , because  $\tau_1 = 0$  and  $x_{i1} = 0$  and  $Y_{i2} = \theta + \tau_2 + \varepsilon_{i2}$ , because  $x_{i2} = 1$  (for all  $i$ )

Using a dummy variables to model our categorical variables `gt type` we can perform a **2-sample *t*-test** with a linear model

$$Y_{ij} = \theta + \tau_2 \times x_{ij} + \varepsilon_{ij}; \quad x_{ij} = \begin{cases} 1 & \text{if } j = 2 \\ 0 & \text{if } j = 1 \end{cases}$$

```
list("t value"=ttest.irs4$stat,"p-value"=ttest.irs4$p.value)
```

```
## $t value
##      t
## -0.5286494
##
## $p-value
## [1] 0.6002058
```

```
list("t value"=lm.irs4$coeff[2,3],"p-value"=lm.irs4$coeff[2,4])
```

```
## $t value
## [1] 0.5286494
##
## $p-value
## [1] 0.6002058
```

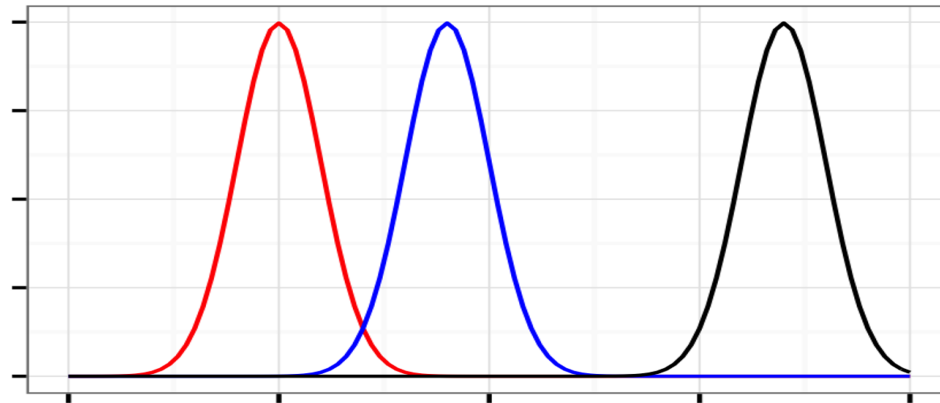
# Beyond 2-groups comparisons: difference of means

“cell-means”

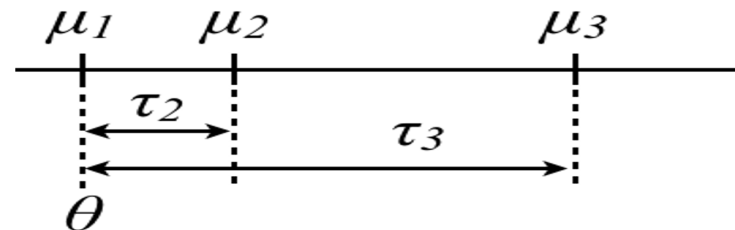
$$Y_{ij} = \mu_j + \varepsilon_{ij}$$

“reference-treatments”

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



More than 2 groups!



Dummy variables can be used to model one *or more* categorical variables with 2 *or more* levels!

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

$$Y_{ij} = \theta + \tau_2 \times x_{ij} + \varepsilon_{ij}; \quad x_{ij} = \begin{cases} 1 & \text{if } j = 2 \\ 0 & \text{if } j = 1 \end{cases}$$

1-way ANOVA with many levels <sup>(\*)</sup> using a linear model

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

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

(\*) in general, yet another parametrization is used to present ANOVA

## t-test

Special case of **ANOVA**, but with ANOVA you can compare **more than two groups** and **more than one factor**.

## ANOVA

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 **many quantitative and qualitative variables**.

**In R:** all 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



Let's form an  $X$  matrix for a 3-groups comparison:

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

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

response  $Y$ 
design matrix  $X$ 
regression parameters
error term

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

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

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

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

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

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

Which is the same as  $\rightarrow Y_{ij} = \theta + \tau_j + \varepsilon_{ij}, \tau_1 = 0$

Which is the same as  $\rightarrow Y_{ij} = \theta + \tau_2 \times x_{ij2} + \tau_3 \times x_{ij3} + \varepsilon_{ij}$

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

$$\begin{bmatrix} Y_{11} \\ Y_{21} \\ \vdots \\ Y_{n_33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \vdots \\ \varepsilon_{n_33} \end{bmatrix}$$

Diagram illustrating the relationship between the parameters  $\theta$ ,  $\tau_2$ ,  $\tau_3$  and the group means  $\mu_1$ ,  $\mu_2$ ,  $\mu_3$ :

- $\theta$  is the mean for the reference group ( $\mu_1$ ).
- $\tau_2$  is the difference between the mean for group 2 and the reference group ( $\mu_2 - \mu_1$ ).
- $\tau_3$  is the difference between the mean for group 3 and the reference group ( $\mu_3 - \mu_1$ ).

Note that the model is still written with a reference-treatment parametrization (difference of means)

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

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

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

Linear regression can include quantitative & qualitative covariates.

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

This gives us a VERY FLEXIBLE framework!!

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

1 categorical  
covariate

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

2 categorical  
covariates

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

1 continuous  
covariate

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

1 continuous  
1 categorical

AND MANY MORE .....

Tip: ?model.matrix

Linear in the parameters  $\alpha$ :  $X$  can contain  $x^2$ ,  $\log(x)$ , etc.

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

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



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

**y ~ x:** formula,  
    **y** numeric,  
**x** numeric and/or factor

**yourData:** data.frame in which x  
and y are to be found (optional but  
recommended)

By default, R uses a ref-tx parametrization but you can control that!

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

- Mathematically,  $X$  is a numeric matrix
- If your data contains categorical variables (e.g., `gType`), you need to set them as **factors**
- R creates appropriate dummy variables for factors!

```
str(irs4Dat$gType)
```

```
## Factor w/ 2 levels "NrlK0","wt": 2 2 2 2 1 1 1 2 2 2 ...
```

## Under the hood, R creates a numeric $X$ :

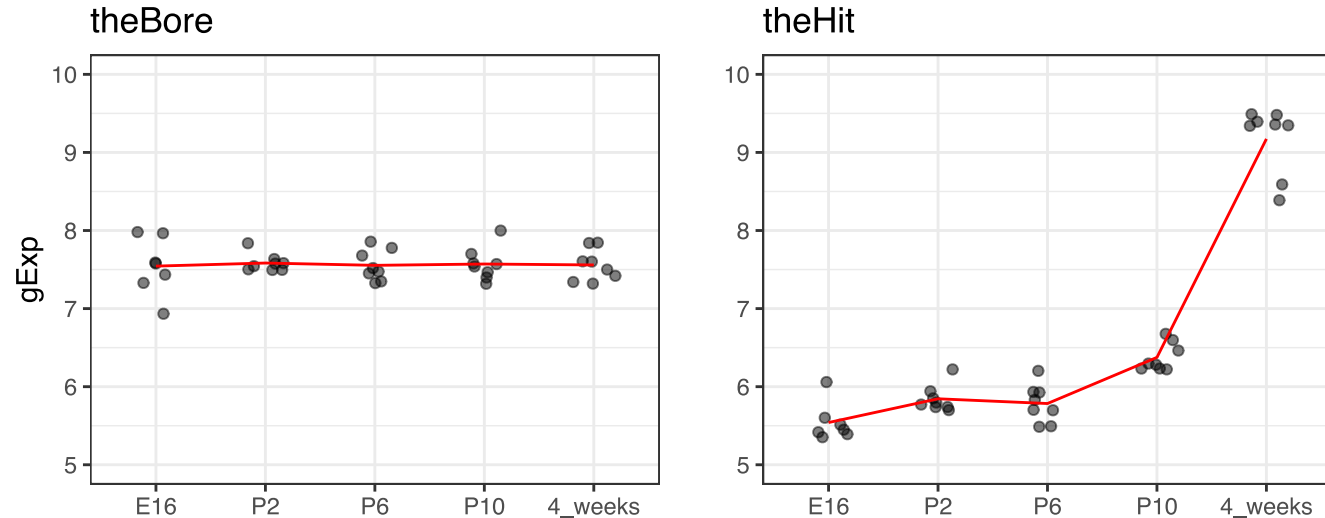
```
data.frame(X = model.matrix(gExp ~ gType, irs4Dat),  
           gType = irs4Dat$gType) %>% head(10)
```

```
##      X..Intercept. X.gTypewt gType  
## 1             1         1      wt  
## 2             1         1      wt  
## 3             1         1      wt  
## 4             1         1      wt  
## 5             1         0 Nr1KO  
## 6             1         0 Nr1KO  
## 7             1         0 Nr1KO  
## 8             1         1      wt  
## 9             1         1      wt  
## 10            1         1      wt
```

# Beyond 2-group comparisons in our case study:

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

$$H_0 : \mu_{E16} = \mu_{P2} = \mu_{P6} = \mu_{P10} = \mu_{4W}$$



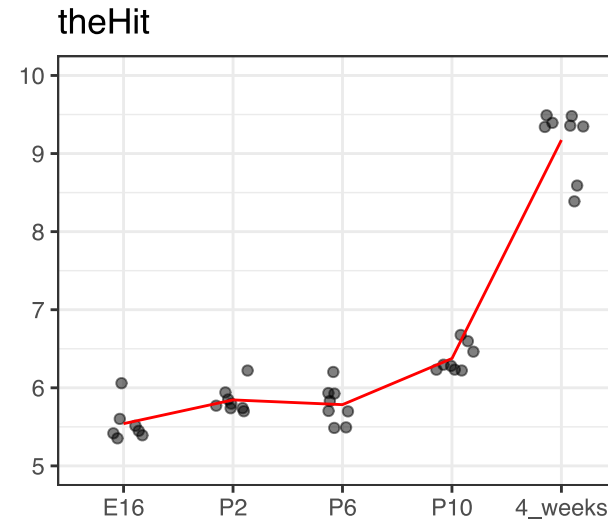
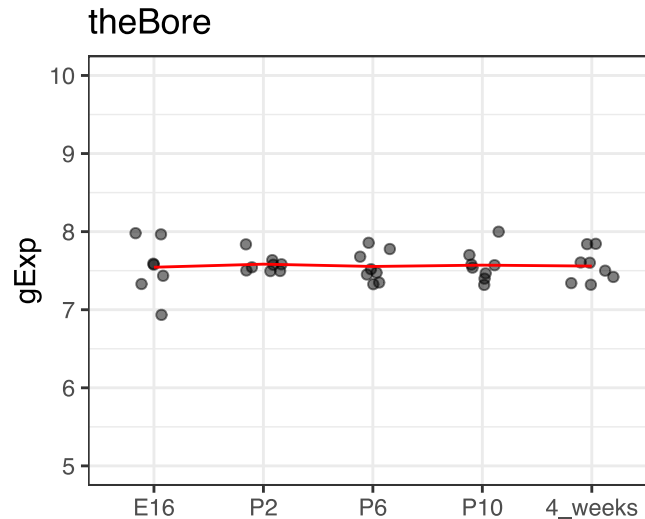
Note: 4W = 4\_weeks



The **sample** means:  $\hat{\mu}_{E16}$ ,  $\hat{\mu}_{P2}$ ,  $\hat{\mu}_{P6}$ ,  $\hat{\mu}_{P10}$ ,  $\hat{\mu}_{4W}$

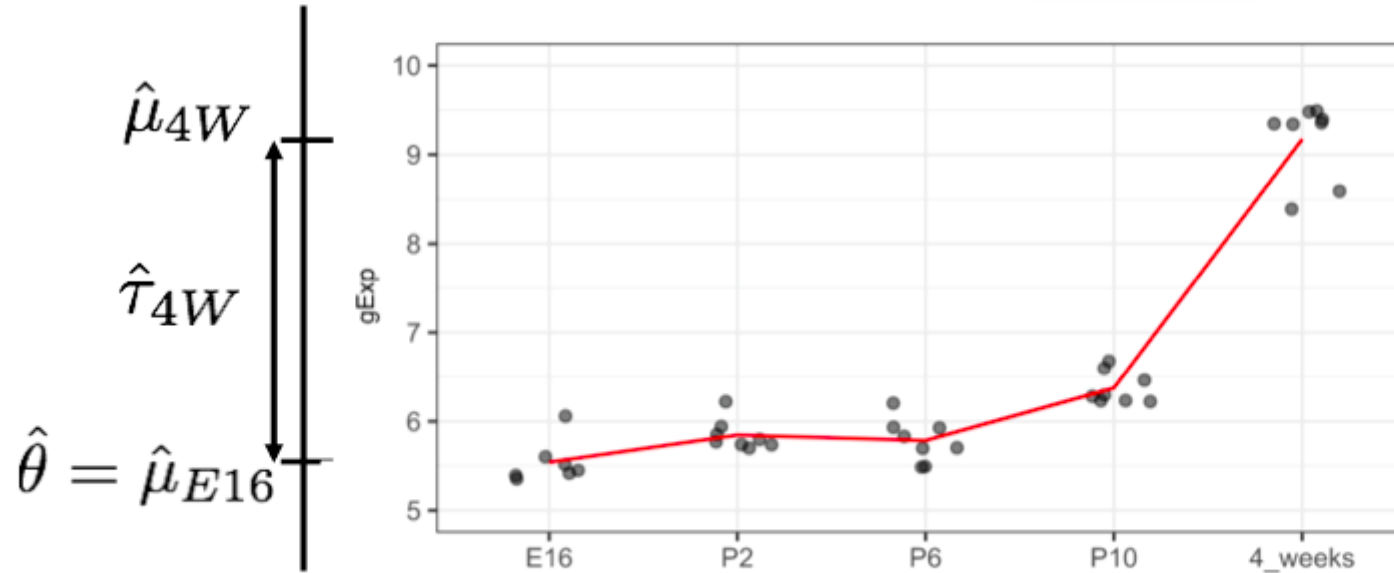
```
with(devDat, tapply(gExp, list(devStage, gene), mean))
```

```
##           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
```



# "theHit" with significant time ("treatment") effect

	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



"theHit" with significant time ("treatment") effect

Can you guess the size of the  $X$  matrix??

- | How many dummy variables do we need?

## "theHit" with significant time ("treatment") effect

```
##      devStage cellMeans txEffects
## 1      E16    5.540857 0.0000000
## 2      P2     5.844875 0.3040179
## 3      P6     5.784250 0.2433929
## 4     P10     6.375125 0.8342679
## 5  4_weeks    9.173375 3.6325179
```

We need 4 dummy variables to estimate and test 4 time differences:

|  $x_{P2}$ : P2 vs E16,  $x_{P6}$ : P6 vs E16,  $x_{P10}$ : P10 vs E16,  $x_{4W}$ : 4W vs E16)

Mathematically:

$$Y_{ij} = \theta + \tau_{P2} \times x_{ijP2} + \tau_{P6} \times x_{ijP6} + \tau_{P10} \times x_{ijP10} + \tau_{4W} \times x_{ij4W} + \varepsilon_{ij}$$

Notation:  $x_{ijk}$ , where  $i$  is an index for the observation,  $j$  for the level of devStage, and  $k$  for the name of the dummy variable

## Under the hood, R creates a numeric $X$ :

```
X.matrix <- data.frame(X = model.matrix(gExp ~ devStage, irs4Dat),  
  devStage = irs4Dat$devStage)
```

##	X..Intercept.	X.dStP2	X.dStP6	X.dStP10	X.dS4W	dS
## 1	1	0	0	0	0	E16
## 2	1	0	0	0	0	E16
## 3	1	0	0	0	0	E16
## 4	1	0	0	0	0	E16
## 5	1	0	0	0	0	E16
## 6	1	0	0	0	0	E16
## 7	1	0	0	0	0	E16
## 8	1	1	0	0	0	P2
## 9	1	1	0	0	0	P2
## 10	1	1	0	0	0	P2
## 11	1	1	0	0	0	P2
## 12	1	1	0	0	0	P2
## 13	1	1	0	0	0	P2
## 14	1	1	0	0	0	P2
## 15	1	1	0	0	0	P2
## 16	1	0	1	0	0	P6

Note: column names changed and first 16 rows displayed to fit output in the page (code hidden)

```
summary(lm(gExp~devStage,subset(devDat, gene=="theHit")))$coeff
```

```
##              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
```

```
means.dev %>% mutate(txEffects=cellMeans-cellMeans[1])
```

```
##   devStage cellMeans txEffects
## 1      E16    5.540857 0.0000000
## 2       P2    5.844875 0.3040179
## 3       P6    5.784250 0.2433929
## 4      P10    6.375125 0.8342679
## 5   4_weeks    9.173375 3.6325179
```

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

$H_0 : \theta = 0$  or

$H_0 : \mu_{E16} = 0$

we are not usually interested in testing this hypothesis

```
summary(lm(gExp~devStage,subset(devDat, gene=="theHit")))$coeff
```

```
##              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
```

```
means.dev %>% mutate(txEffects=cellMeans-cellMeans[1])
```

```
##   devStage cellMeans txEffects
## 1      E16  5.540857 0.0000000
## 2      P2  5.844875 0.3040179
## 3      P6  5.784250 0.2433929
## 4     P10  6.375125 0.8342679
## 5  4_weeks  9.173375 3.6325179
```

**Estimate:**

$$\hat{\tau}_{P2} = \hat{\mu}_{P2} - \hat{\mu}_{E16} = \bar{Y}_{.P2} - \bar{Y}_{.E16}$$

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

we *are* usually interested in testing this hypothesis: first 2 days after birth

```
summary(lm(gExp~devStage,subset(devDat, gene=="theHit")))$coeff
```

```
##              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
```

```
means.dev %>% mutate(txEffects=cellMeans-cellMeans[1])
```

```
##   devStage cellMeans txEffects
## 1      E16   5.540857 0.0000000
## 2       P2   5.844875 0.3040179
## 3       P6   5.784250 0.2433929
## 4      P10   6.375125 0.8342679
## 5  4_weeks   9.173375 3.6325179
```

### Estimate:

$$\hat{\tau}_{4W} = \hat{\mu}_{4W} - \hat{\mu}_{E16} = \bar{Y}_{.4W} - \bar{Y}_{.E16}$$

$$H_0 : \tau_{4W} = 0 \text{ or}$$

we *are* usually interested in testing this hypothesis: 4 weeks after birth





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

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

We generally test two types of null hypotheses:

$$H_0 : \tau_j = 0$$

vs

$$H_0 : \tau_j \neq 0$$

for each  $j$  **individually**

e.g., Is gene A differentially expressed 2 days after birth?

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

$$H_0 : \tau_j = 0$$

vs

$$H_0 : \tau_j \neq 0$$

for all  $j$  **at the same time**

e.g., Is gene A significantly affected by time (devStage)?

$$H_0 : \tau_{P2} = \tau_{P6} = \tau_{P10} = \tau_{4W} = 0$$

## Two types of null hypotheses in R:

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

$$H_0 : \tau_j = 0$$

vs

$$H_0 : \tau_j \neq 0$$

for each  $j$  individually

$$H_0 : \tau_j = 0 \quad \text{AND statement}$$

vs

$$H_0 : \tau_j \neq 0 \quad \text{OR statement}$$

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
```

## *F*-test and overall significance of one or more covariates

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

$$H_0 : \tau_i = 0$$

$$H_A : \tau_j \neq 0$$

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

$$H_0 : \tau_{P2} = \tau_{P6} = \tau_{P10} = \tau_{4W} = 0 \text{ [AND statement]}$$

$$H_A : \tau_i \neq 0 \text{ for some } i \text{ [OR statement]}$$

the *F*-test allows us to test such compound tests

# To conclude

- we can use different parametrizations to write statistical models

From **cell-means** -  $\mu_j$ :  $Y_{ij} = \mu_j + \varepsilon_{ij}$ ;  $\varepsilon_{ij} \sim G$ ;  $E[\varepsilon_{ij}] = 0$ ;

to **reference-treatment effect** -  $(\theta, \tau_j)$ : (used by default by `lm`)

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

- we can compare group means (2 or more) using a linear model
  - **dummy variables** (e.g.,  $x_{ijP2}$ ) to model the levels of a qualitative explanatory variables

$$Y_{ij} = \theta + \tau_{P2} \times x_{ijP2} + \tau_{P6} \times x_{ijP6} + \tau_{P10} \times x_{ijP10} + \tau_{4W} \times x_{ij4W} + \varepsilon_{ij}$$

- qualitative variables need to be set as "factors" in the data --> R creates the dummy variables

- we can write a linear model using matrix notation:

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

- **Linear model** can include **quantitative & qualitative covariates**.

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

This gives us a VERY FLEXIBLE framework!!

$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 1 \end{bmatrix}$ <p>1 categorical covariate</p>	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 \end{bmatrix}$ <p>2 categorical covariates</p>	$\begin{bmatrix} 1 & 1.22 \\ 1 & 2.02 \\ 1 & 1.42 \\ \vdots & \vdots \\ 1 & 1.89 \\ 1 & 2.01 \\ \vdots & \vdots \\ 1 & 1.56 \\ 1 & 2.17 \\ 1 & 1.51 \end{bmatrix}$ <p>1 continuous covariate</p>	$\begin{bmatrix} 1 & 0 & 1.22 & 0 \\ 1 & 0 & 2.02 & 0 \\ 1 & 0 & 1.42 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1.89 & 0 \\ 1 & 1 & 2.01 & 2.01 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1.56 & 1.56 \\ 1 & 1 & 2.17 & 2.17 \\ 1 & 1 & 1.51 & 1.51 \end{bmatrix}$ <p>1 continuous 1 categorical</p>
---	--	--	--

AND MANY MORE .....

Tip: ?model.matrix

- distinguish between single and joint hypotheses:

- $t$ -tests vs  $F$ -tests