Lecture 6 – ANOVA and Linear Models

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

Keegan Korthauer

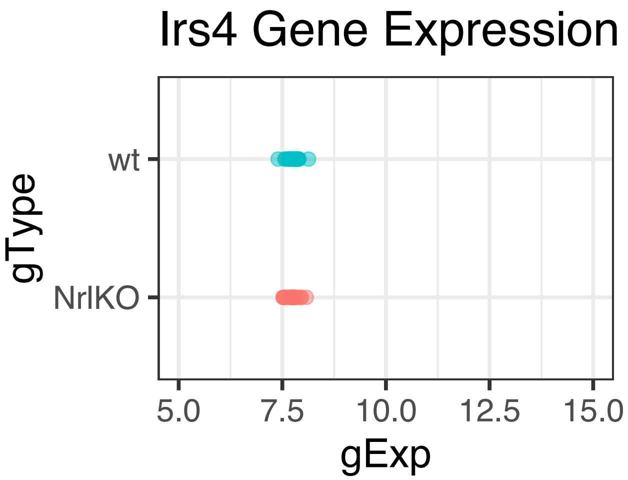
2020/01/22

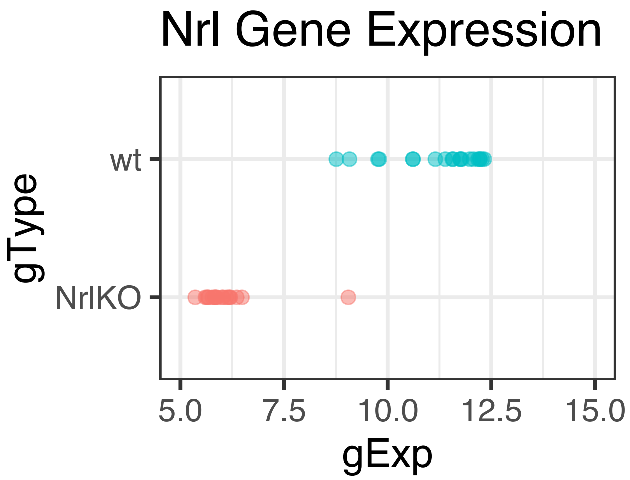
Slides by: Gabriela Cohen Freue with contributions from Jenny Bryan and Keegan Korthauer

Recap: Are these genes truly different in NrlKO compared to WT?

H_0: the expression level of gene g 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

Population (Unknown)

$$Y \sim F$$

$$Z \sim G$$

$$E[Y] = \mu_Y$$

$$E[Z] = \mu_Z$$

$$H_0 : \mu_Y = \mu_Z$$

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

Sample (Observed, with randomness)

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

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

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

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

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 *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 value (α, 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 with equal variance.

```r
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 NrlKO    mean in group wt
##            7.739684            7.765750
```

Today...

1. 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
2. Write a linear model using matrix notation
   - understand which matrix is built by R
3. distinguish between conditional and marginal effects
   - t-tests vs F-tests

$$H_0 : \mu_1 = \mu_2$$

2-sample t-test (with equal variance)

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

(one-way) Analysis of Variance (ANOVA)

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

Linear regression model

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

All three methods give the same result!

2-sample t-test (with equal variance)

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

(one-way) Analysis of Variance (ANOVA)

```
## 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
```

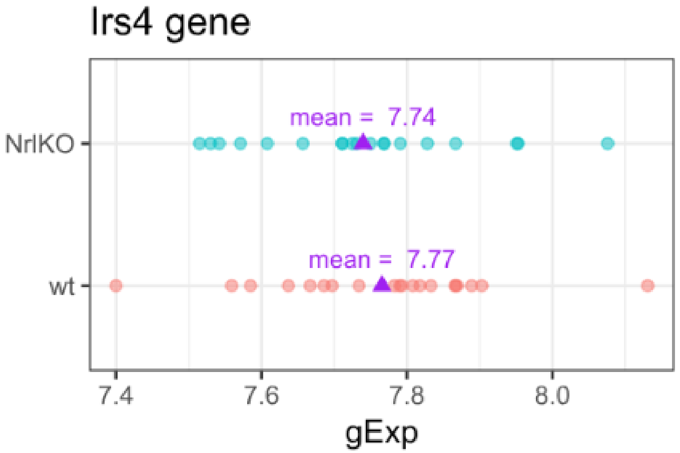
Linear regression model

```
## Coefficients:
## Estimate Std. Error t value
Pr(>|t|)
## (Intercept) 7.73968 0.03531
219.198 <2e-16 ***
## gTypewt 0.02607 0.04931 0.529 0.6
```

```
> 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!

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

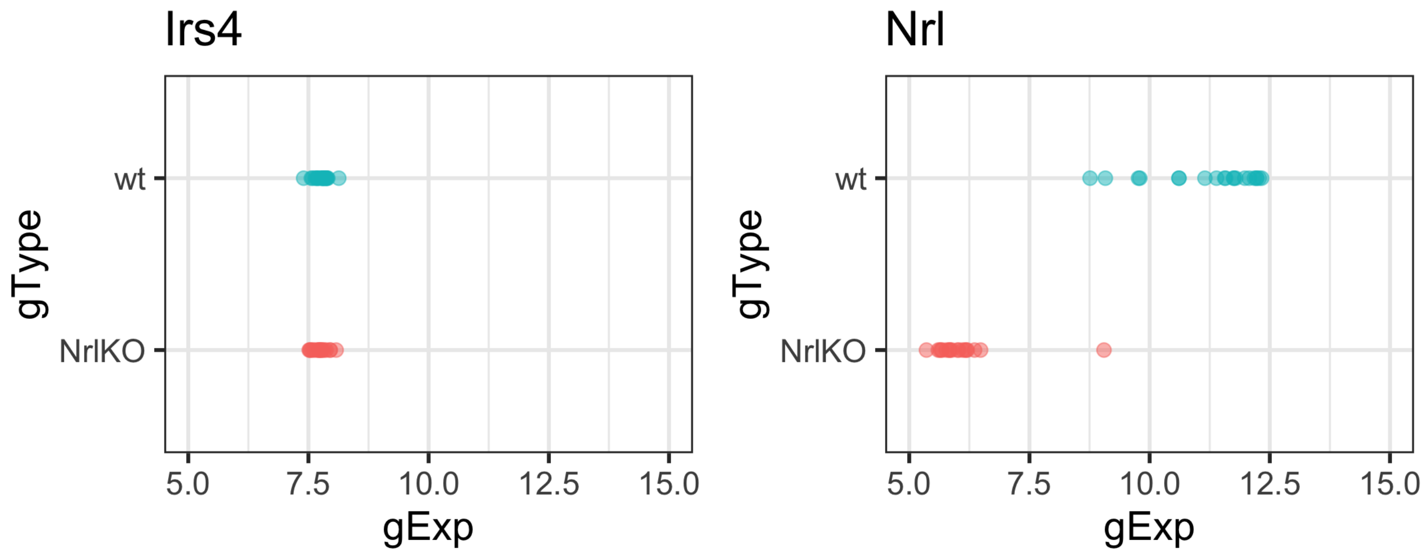
```
irs4Dat <- subset(miniDat,gene=="Irs4")
ttest.irs4<-t.test(gExp ~ gType, irs4Dat, var.equal = TRUE)
list("t value"=ttest.irs4$stat,"p-value"=ttest.irs4$p.value)
```

```
## $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

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

$$\downarrow$$

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

We can use a subindeces to distinguish observations from each group, i.e.,

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

where $j = \{\text{wt, NrlKO}\}$ 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

Cell-means model

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 $i = 1, \ldots, n_j$.

> 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 (μ_j) parametrization

Recall: sample mean estimator of population mean

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

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

```r
(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
```

```r
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 should 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

Let's reformulate from **cell-means** (μ_j):

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

$$\downarrow$$

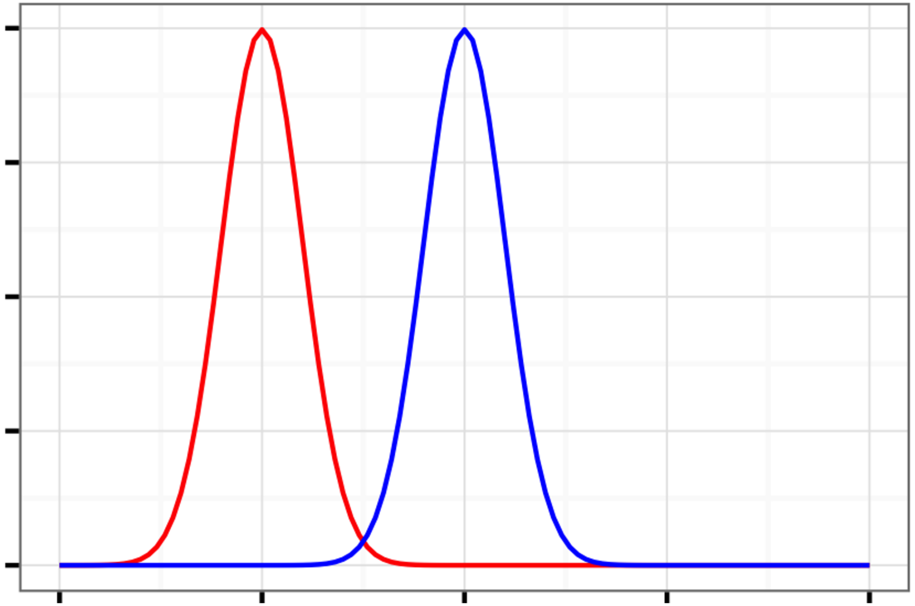
to **reference-treatment effect** (θ, τ_j):

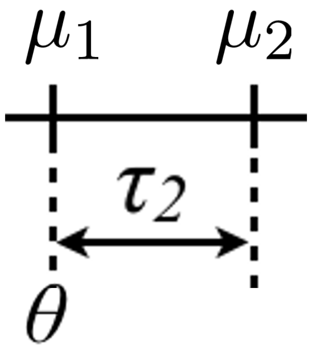
$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij}; \quad \tau_1 = 0, \quad \varepsilon_{ij} \sim G; \quad E[\varepsilon_{ij}] = 0;$$

- Note that for each group, the population mean is given by $E[Y_{ij}] = \theta + \tau_j = \mu_j$, 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

Relation between parametrizations





$$H_0 : \mu_1 = \mu_2$$

$$H_0 : \tau_2 = 0$$

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

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

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

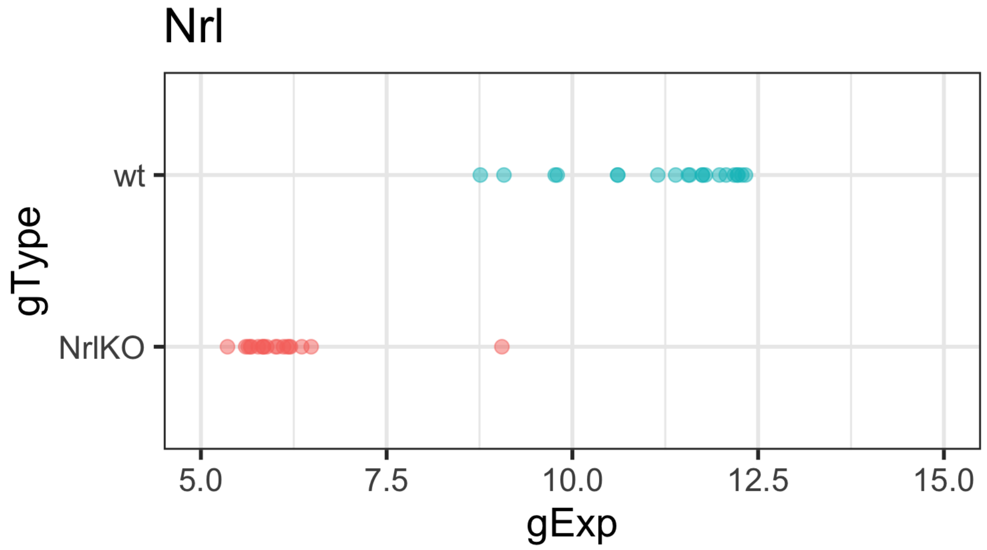
```
## (Intercept)     gTypewt
##  7.73968421  0.02606579
```

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

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

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

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



Dummy variables

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

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

$$\downarrow$$

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

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

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

Using dummy variables to model our categorical variables `gtype` 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
```

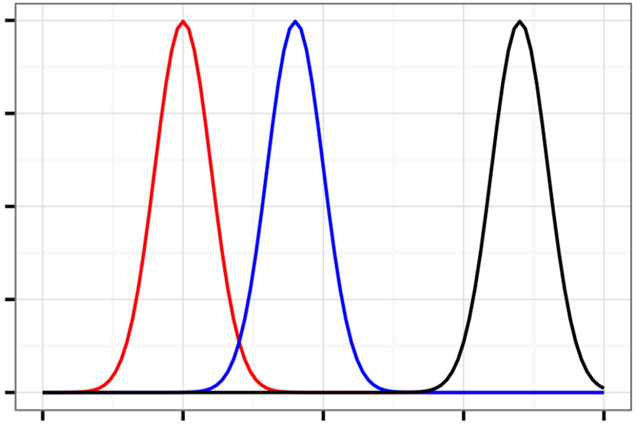
Beyond 2-groups comparisons: difference of means

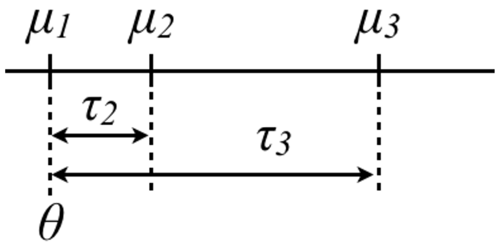
“cell-means”

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

“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 $^{(*)}$ using a linear model

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

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

$^{(*)}$ in general, *yet* another parametrization can be 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

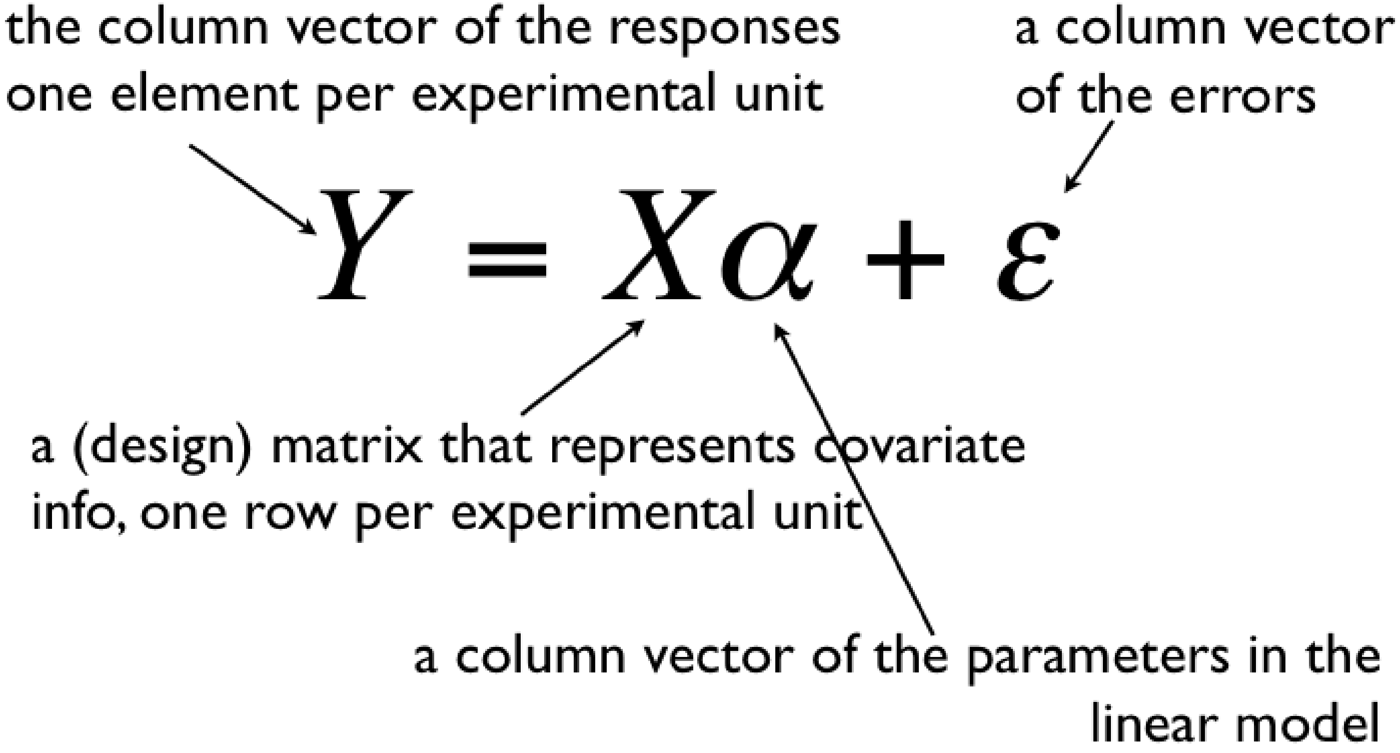
> Special case of linear regression, but with linear regression you can include **quantitative variables** in the model.

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:

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

Note that x_{ij2} and x_{ij3} become the 2nd and 3rd columns of X:

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

$$\underbrace{\begin{bmatrix} Y_{11} \\ \vdots \\ Y_{n_1 1} \\ Y_{12} \\ \vdots \\ Y_{n_2 2} \\ Y_{13} \\ \vdots \\ Y_{n_3 3} \end{bmatrix}}_{\text{response } \boldsymbol{Y}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 1 \end{bmatrix}}_{\text{design matrix } \boldsymbol{X}} \underbrace{\begin{bmatrix} \theta \\ \tau_2 \\ \tau_3 \end{bmatrix}}_{\text{regression parameters}} + \underbrace{\begin{bmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{n_1 1} \\ \varepsilon_{12} \\ \vdots \\ \varepsilon_{n_2 2} \\ \varepsilon_{13} \\ \vdots \\ \varepsilon_{n_3 3} \end{bmatrix}}_{\text{error term}}$$

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

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

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

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

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

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

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

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

Reference group: μ_1

$\mu_2 - \mu_1$

$\mu_3 - \mu_1$

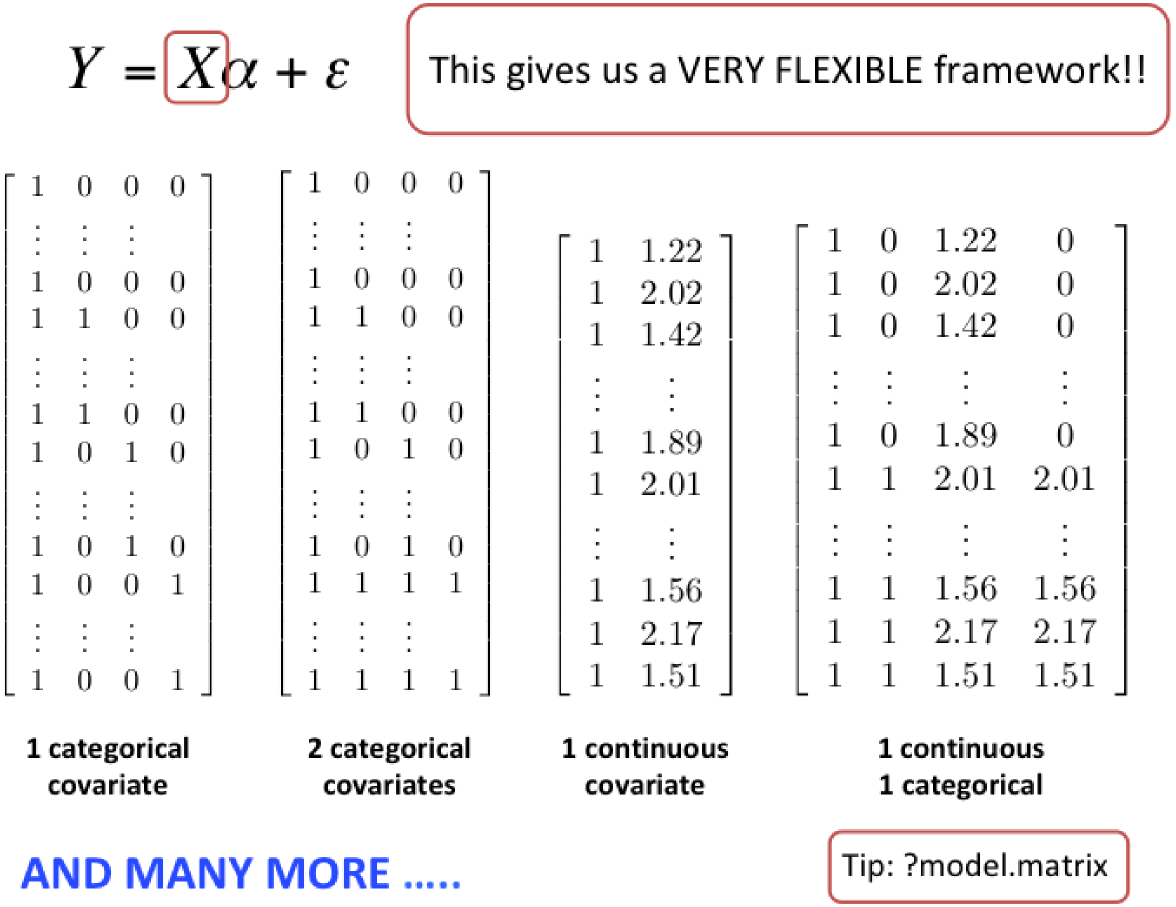
Note that the model is still written with a reference-treatment parametrization (difference of means)

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

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

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

Linear regression can include **quantitative & qualitative covariates**.



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

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

1 continuous
1 categorical

AND MANY MORE .....

Tip: ?model.matrix

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

How it works in practice using lm() in R

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

$$\downarrow$$

```
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!

Special `factor` class in R

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

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

> especially important if your categorical variables are encoded numerically (`lm` will automaticlaly treat character variables as factors)!

- R creates appropriate dummy variables for factors!

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

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

Under the hood, R creates a numeric X:

```
model.matrix(gExp ~ gType, irs4Dat) %>% head(10)
```

```
##    (Intercept) gTypewt
## 1            1       1
## 2            1       1
## 3            1       1
## 4            1       1
## 5            1       0
## 6            1       0
## 7            1       0
## 8            1       1
## 9            1       1
## 10           1       1
```

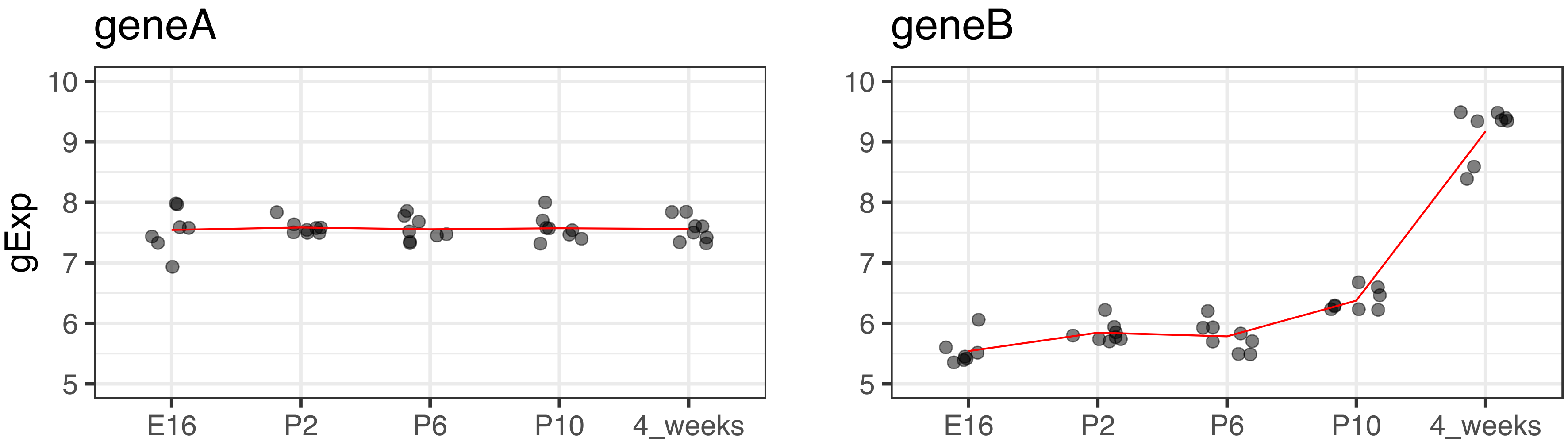
```
irs4Dat$gType %>% head(10)
```

```
##  [1] wt    wt    wt    wt    NrlKO NrlKO NrlKO wt    wt    wt   
## Levels: NrlKO wt
```

Beyond 2-group comparisons in our case study:

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

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

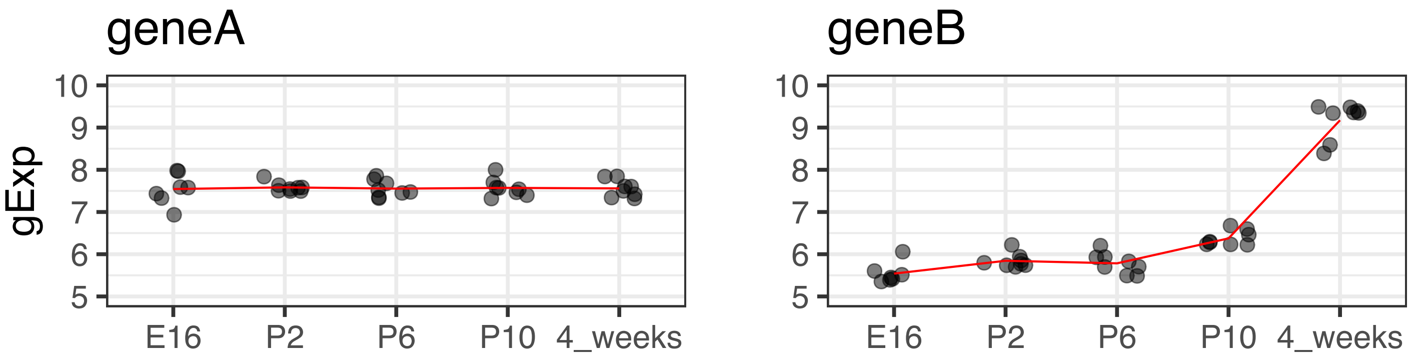


Note: 4W = 4_weeks

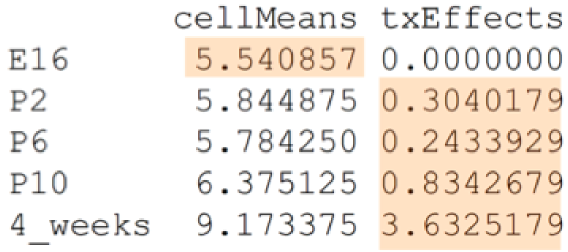
The sample means: $\hat{\mu}_{E16}, \ \hat{\mu}_{P2}, \ \hat{\mu}_{P6}, \ \hat{\mu}_{P10}, \ \hat{\mu}_{4W}$

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

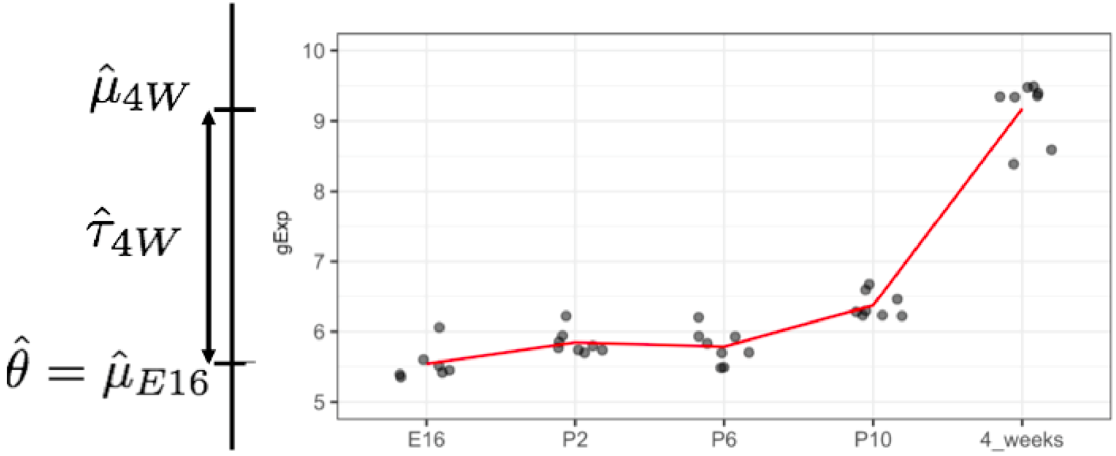
```
##              geneA    geneB
## E16      7.544143 5.540857
## P2       7.583500 5.844875
## P6       7.554000 5.784250
## P10      7.571000 6.375125
## 4_weeks 7.559000 9.173375
```



"geneB" 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
```



Can you guess the size of the X matrix??

> How many dummy variables do we need?

"geneB" 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 (between 5 time points):

> 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:

```
model.matrix(gExp ~ devStage, irs4Dat) %>% head(16)
```

```
##    (Intercept) devStageP2 devStageP6 devStageP10 devStage4_weeks
## 1            1          0          0           0               0
## 2            1          0          0           0               0
## 3            1          0          0           0               0
## 4            1          0          0           0               0
## 5            1          0          0           0               0
## 6            1          0          0           0               0
## 7            1          0          0           0               0
## 8            1          1          0           0               0
## 9            1          1          0           0               0
## 10           1          1          0           0               0
## 11           1          1          0           0               0
## 12           1          1          0           0               0
## 13           1          1          0           0               0
## 14           1          1          0           0               0
## 15           1          1          0           0               0
## 16           1          0          1           0               0
```

```r
summary(lm(gExp~devStage,subset(devDat,gene=="geneB")))$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
```

```r
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
```

$H_0 : \theta = 0$ 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

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

$H_0 : \tau_{P2} = 0$ 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

```
summary(lm(gExp~devStage,subset(devDat,gene=="geneB")))$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])
```

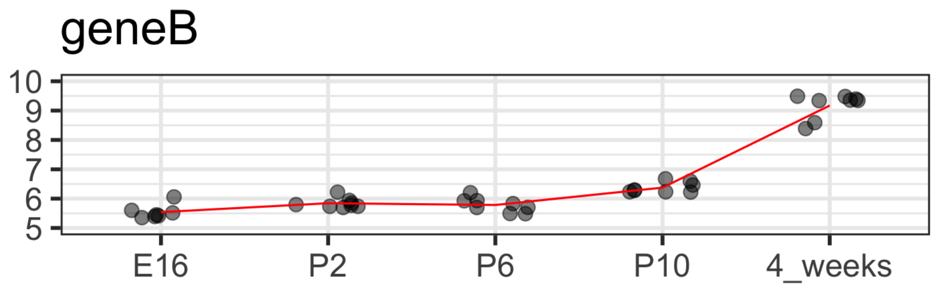
```
##   devStage cellMeans txEffects
## 1      E16  5.540857 0.0000000
## 2       P2  5.844875 0.3040179
## 3       P6  5.784250 0.2433929
## 4      P10  6.375125 0.8342679
## 5  4_weeks  9.173375 3.6325179
```

$H_0 : \tau_{4W} = 0$ or $H_0 : \mu_{4W} = \mu_{E16}$

Estimate:

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

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



```
> 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
```

All data points are used to estimate the variance of the error term!!

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

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

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 differencially 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

$H_0 : \tau_j \neq 0$

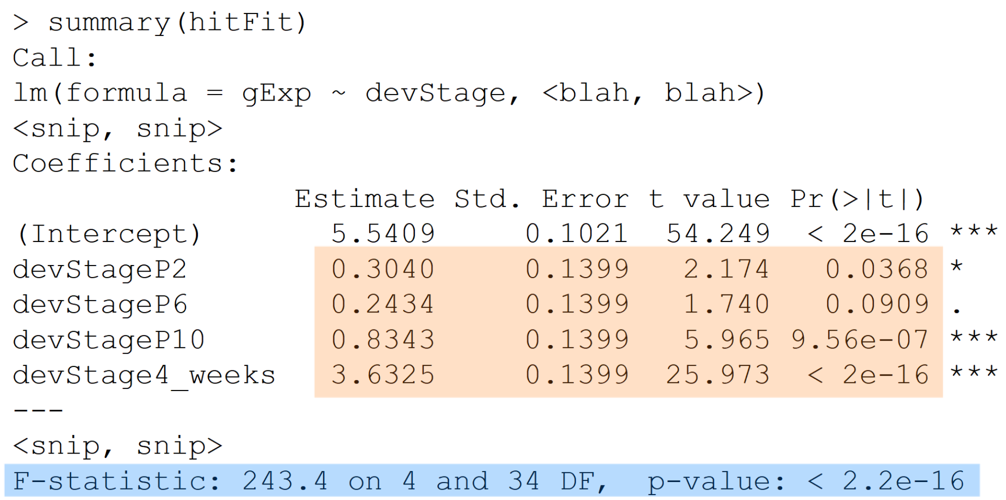
for each j individually

$H_0 : \tau_j = 0$ 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
```

F-test and overall significance of one or more covariates

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

$$H_0 : \tau_j = 0$$

$$H_A : \tau_j \neq 0$$

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

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

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

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

To conclude

1. We can use different parametrizations to write statistical models

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

to **reference-treatment effect** (θ, τ_j): (used by default by `lm`)

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

2. 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

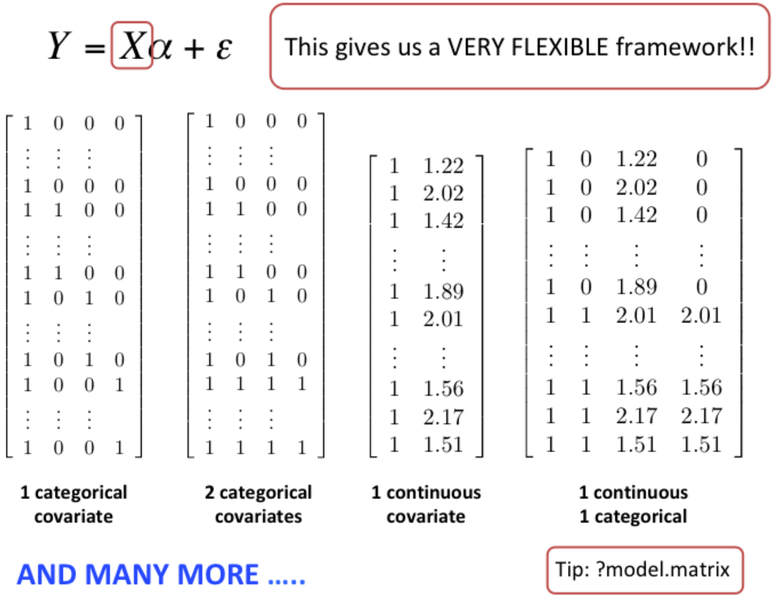
$$Y_{ij} = \theta + \tau_{P2} \times x_{ijP2} + \tau_{P6} \times x_{ijP6} + \tau_{P10} \times x_{ijP10} + \tau_{4W} \times x_{ij4W} + \varepsilon_{ij}$$

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

3. We can write a linear model using matrix notation:

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

4. Linear models 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

1 continuous
1 categorical

AND MANY MORE .....

Tip: ?model.matrix

5. We use different tests to distinguish between single and joint hypotheses:

- t-tests vs F-tests