

Statistical Methods for High Dimensional Biology

STAT/BIOF/GSAT 540

Lecture 7 – Linear models

Gabriela Cohen Freue

January 28 2019

Other contributors: Drs. Jenny Bryan and Sara Mostafavi

Book and online resources

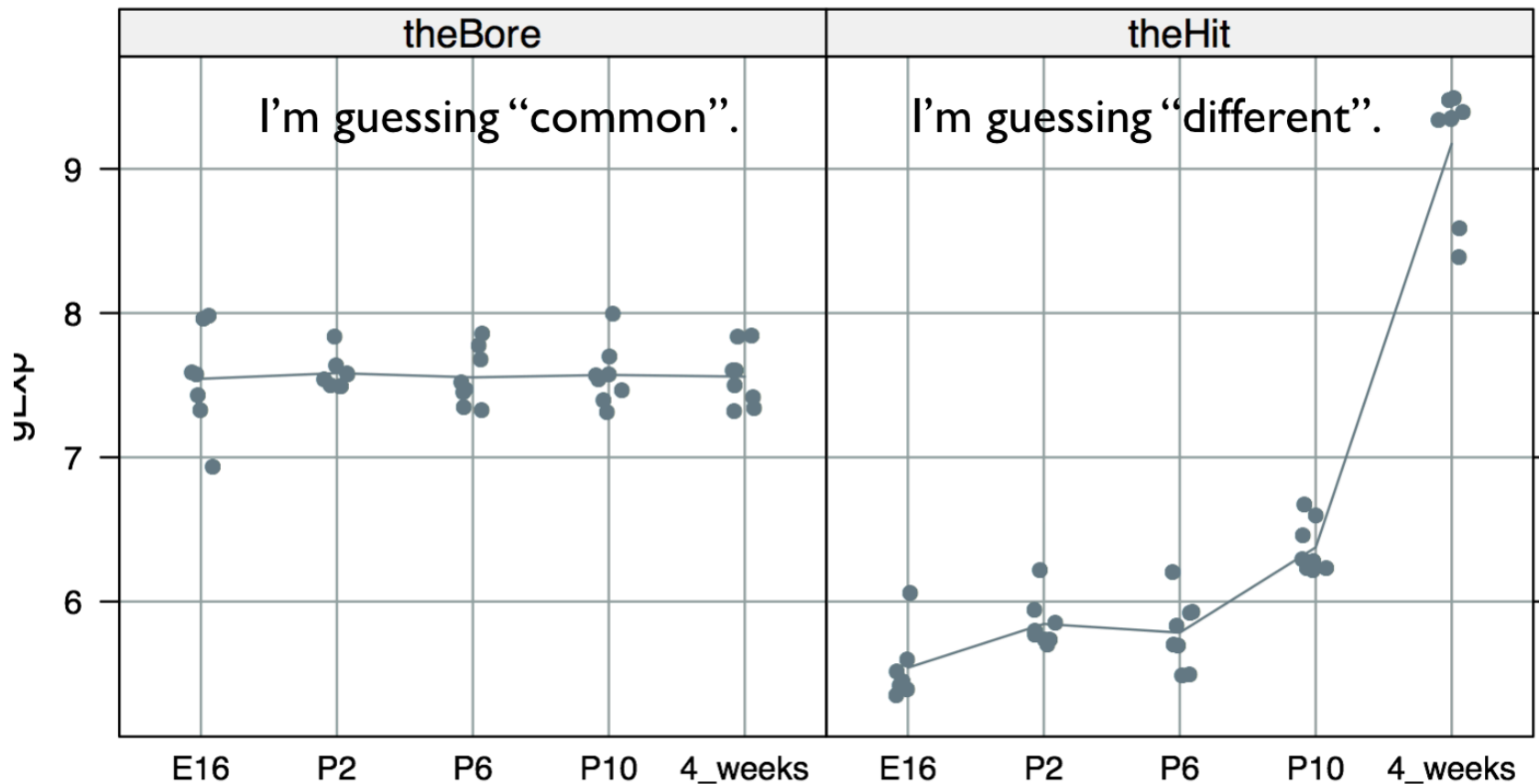
Linear Models with R by Julian J. J. Faraway, Chapman & Hall/CRC Texts in Statistical Science, 2004.

** www.biostat.jhsph.edu/~iruczins/teaching/jf/faraway.html **

** <https://sites.google.com/a/cs.washington.edu/genome560-spr18/CourseMaterials> **

** <http://genomicsclass.github.io/book/> **

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



From t -test to linear regression

2-sample t -test

$$\left. \begin{array}{l} Y \sim G; \ E[Y] = \mu_Y \\ Z \sim G; \ E[Z] = \mu_Z \end{array} \right\} H_0 : \mu_Y = \mu_Z$$



Linear regression

$$Y = X\alpha + \varepsilon \quad H_0 : \alpha_j = 0$$

HOW? and WHY??

WHY??

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

This gives us a VERY FLEXIBLE framework!!

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

**1 categorical
covariate**

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

**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

HOW??

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



$$Y_j = \mu_j + \varepsilon_j; \varepsilon_j \sim G; E[\varepsilon_j] = 0; j = \{1, 2\}$$

Response
variable



$$\begin{bmatrix} Y_{11} \\ \vdots \\ Y_{n_1 1} \\ Y_{12} \\ \vdots \\ Y_{n_2 2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{n_1 1} \\ \varepsilon_{12} \\ \vdots \\ \varepsilon_{n_2 2} \end{bmatrix}$$

I've used a matrix with dummy variables to re-write the problem in the form of a linear regression

DONE??

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

DONE??

$$\begin{bmatrix} Y_{11} \\ \vdots \\ Y_{n_1 1} \\ Y_{12} \\ \vdots \\ Y_{n_2 2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{n_1 1} \\ \varepsilon_{12} \\ \vdots \\ \varepsilon_{n_2 2} \end{bmatrix}$$

These are
population means

These are **NOT**
sample means

By default, R
uses a different
parametrization

```
> 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
```

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

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”

To parametrize difference in population means:

$$Y_{ij} = \theta + \tau_j + \varepsilon_{ij}, \quad (\tau_1 = 0) \quad E[Y_{ij}] = \theta + \tau_j; \text{ for all } i, j$$

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

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

These are **NOT**
population means

These are **differences**
in population means

ANOVA-style, “cell means”

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

```
lm(y ~ 0 + x, data = jDat)
lm(y ~ -1 + x, data = jDat)
```

```
> 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

ANOVA-style, “ref + tx effects”

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

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

```
> summary(hitFit)
```

Call:

```
lm(formula = gExp ~ devStage, <blah, blah>)
```

<snip, snip>

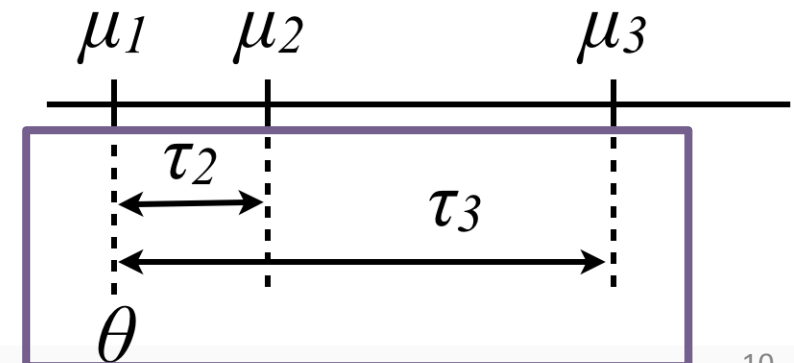
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.5409	0.1021	54.24	< 2e-16 ***
devStageP2	0.3040	0.1399	2.17	0.0368 *
devStageP6	0.2434	0.1399	1.74	0.0909 .
devStageP10	0.8343	0.1399	5.96	9.56e-07 ***
devStage4_weeks	3.6325	0.1399	25.97	< 2e-16 ***

<snip, snip>

F-statistic: 243.4 on 4 and 34 DF, p-value: < 2.2e-16

“ref + tx effects” is
what most people want,
most of the time, in this
setting



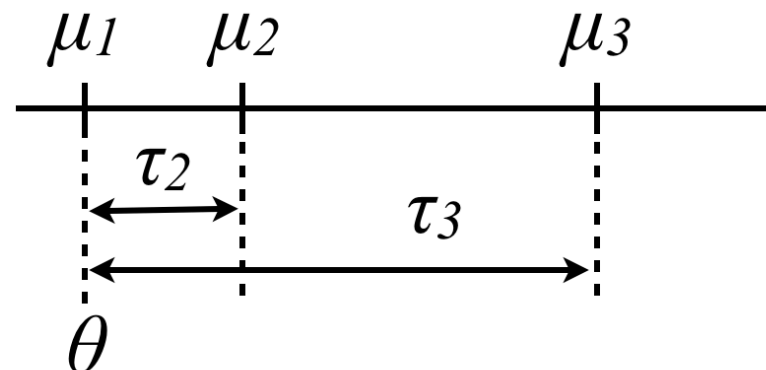
Recall we can obtain one set of parameters from the others!

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

These are
population means

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

These are **NOT** population means
These are **ref & TX** effects



```
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 ***

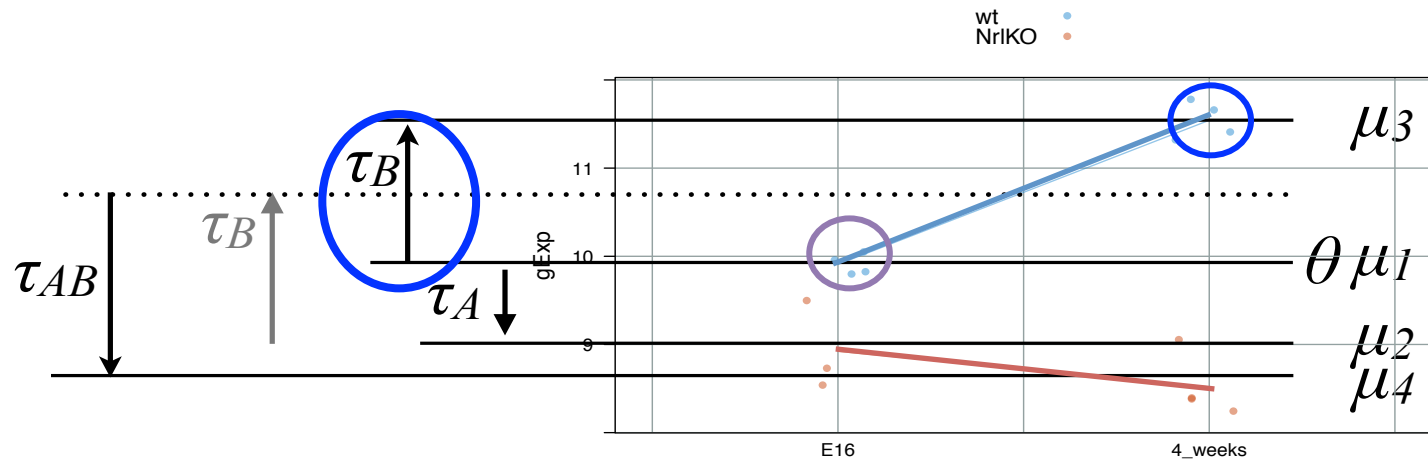
```
Call:
lm(formula = gExp ~ devStage, <blah, blah>)
<snip, snip>
Coefficients:

```

	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
```

Two factors with interaction: simple effect

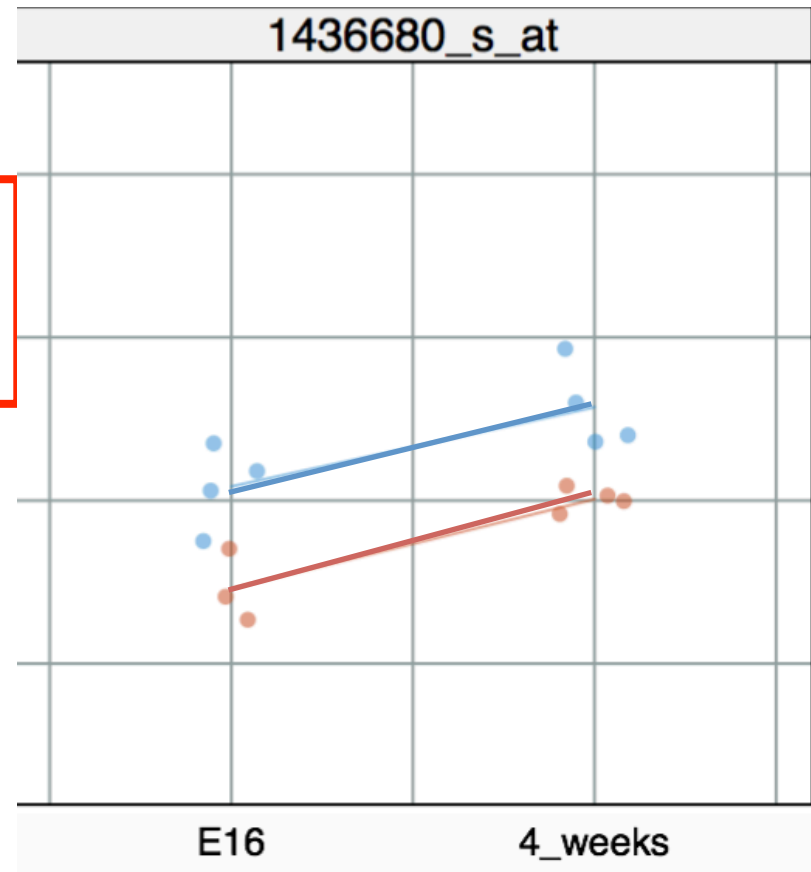


```
> summary(twoFactFit)
lm(formula = gExp ~ gType * devStage, data = miniDat)
<snip, snip>
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	9.9080	0.1575	62.911	2.03e-15	***
<u>gTypeNr1KO</u>	-0.9857	0.2406	-4.097	0.00177	**
<u>devStage4_weeks</u>	1.6345	0.2227	7.339	1.47e-05	***
<u>gTypeNr1KO:devStage4_weeks</u>	-2.0381	0.3278	-6.217	6.56e-05	***

NOTE: association between gExp and devStage **for wt**,
NOT the overall association!!

What if we have evidence of an
additive model??

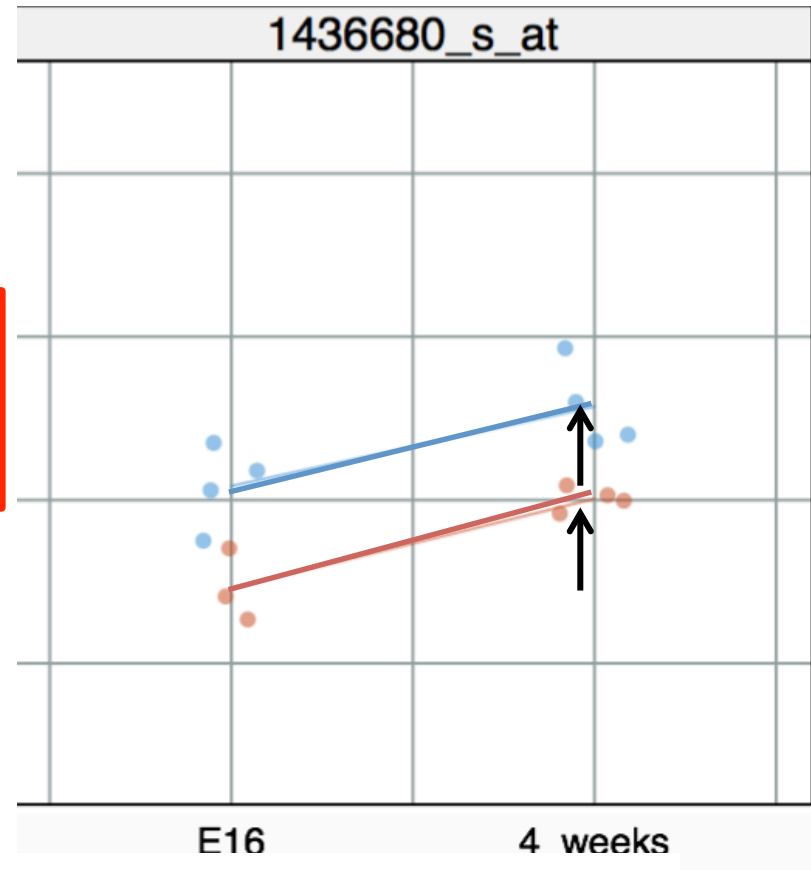


Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	10.0853	0.1076	93.763	< 2e-16	***
gTypeNr1K0	-0.6242	0.1643	-3.799	0.00295	**
devStage4_weeks	0.4873	0.1521	3.203	0.00841	**
gTypeNr1K0:devStage4_weeks	0.0600	0.2239	0.268	0.79368	

Two factors *without* interaction: simple effect

We now test the association between gExp and gType



Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	10.07140	0.09062	111.145	< 2e-16	***
gTypeNr1K0	-0.59194	0.10722	-5.521	0.000132	***
devStage4_weeks	0.51494	0.10722	4.803	0.000431	***

How do we test the **overall** association between the response and a factor??

- Is it easier to use an additive model?
 - Additive models are easier and smaller (less parameters).
 - But in some applications, we need to test the interaction term.
 - And it does not always test the overall association between a factor and a response.

Coefficients:

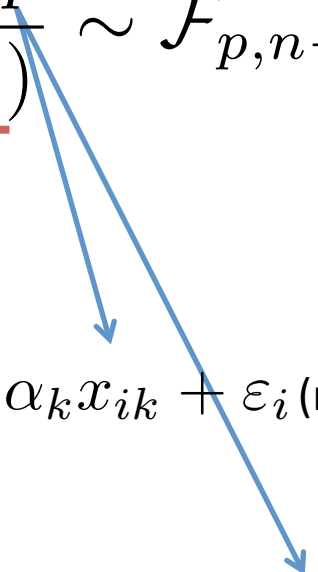
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	9.87496	0.16422	60.131	<2e-16	***
gTypeNr1K0	-0.13358	0.13101	-1.020	0.3153	
devStageP2	-0.08830	0.21153	-0.417	0.6791	
devStageP6	0.07883	0.21153	0.373	0.7118	
devStageP10	-0.35155	0.21153	-1.662	0.1060	
devStage4_weeks	0.48220	0.21153	2.280	0.0292	*

??

How do we test the **overall** association between the response and a factor??

F-test: selection of nested models

$$H_0 : \alpha_{k+1} = \dots = \alpha_{k+p} = 0$$

$$F = \frac{(SS_{reduced} - SS_{full}) / p}{SS_{full} / (n - \underline{p - k - 1})} \sim \mathcal{F}_{p, n-p-k-1}$$


Compares:

Model 1: $y_i = \alpha_0 + \alpha_1 x_{i1} + \dots + \alpha_k x_{ik} + \varepsilon_i$ (reduced: **1+k** parameters)

versus

Model 2: $y_i = \alpha_0 + \alpha_1 x_{i1} + \dots + \alpha_k x_{ik} + \dots + \alpha_{(k+p+1)} x_{i(k+p+1)} + \varepsilon_i$

(full: **1+k+p** parameters)

How do we test the **overall** association between the response and a factor??

F-test: selection of nested models

```
> addFullFit <- lm(gExp ~ gType + devStage, miniDat)
> addRedFit<- lm(gExp ~ gType, miniDat)
> anova(addRedFit,addFullFit)
```

Analysis of Variance Table

Model 1: gExp ~ gType

Model 2: gExp ~ gType + devStage

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	37	8.4386				
2	33	5.5018	4	2.9368	4.4038	0.005803 **

Tests the **overall** association between gExp and devStages, *controlling* for gType

Call:

```
lm(formula = gExp ~ gType + devStage, data = miniDat)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.96542	-0.22162	0.04283	0.28958	0.57283

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	9.87496	0.16422	60.131	<2e-16 ***
gTypeNr1K0	-0.13358	0.13101	-1.020	0.3153
devStageP2	-0.08830	0.21153	-0.417	0.6791
devStageP6	0.07883	0.21153	0.373	0.7118
devStageP10	-0.35155	0.21153	-1.662	0.1060
devStage4_weeks	0.48220	0.21153	2.280	0.0292 *

Signif. codes: 0 '***' 0.001 '**' 0.0

Another *F*-test

Residual standard error: 0.4083 on 33 degrees of freedom

Multiple R-squared: 0.3609, Adjusted R-squared: 0.2641

F-statistic: 3.728 on 5 and 33 DF, p-value: 0.008743

Goodnes of Fit: full vs intercept-only

$$H_0 : \alpha_1 = \dots = \alpha_p = 0 \quad \boxed{\text{Intercept-only}}$$

$$F = \frac{(SS_{reduced} - SS_{full}) / p}{SS_{full} / (n - \underline{p} - 1)} \sim \mathcal{F}_{p, n-p-1}$$

Compares:

$$\text{Model 1: } y_i = \alpha_0 + \varepsilon_i$$

(reduced: 1 parameter)

versus

$$\text{Model 2: } y_i = \alpha_0 + \alpha_1 x_{i1} + \alpha_2 x_{i2} + \dots + \alpha_p x_{ip} + \varepsilon_i$$

(full: 1+p parameters)

Goodnes of Fit: full vs intercept-only

```
devStage4_weeks  0.48220    0.21153    2.280    0.0292 *
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.4083 on 33 degrees of freedom
```

```
Multiple R-squared:  0.3609,    Adjusted R-squared:  0.2641
```

```
F-statistic: 3.728 on 5 and 33 DF,  p-value: 0.008743
```

```
> addInterceptFit<- lm(gExp ~ 1, miniDat)
```

```
> addFullFit <- lm(gExp ~ gType + devStage, miniDat)
```

```
> anova(addInterceptFit,addFullFit)
```

```
Analysis of Variance Table
```

```
Model 1: gExp ~ 1
```

```
Model 2: gExp ~ gType + devStage
```

```
    Res.Df    RSS Df Sum of Sq    F    Pr(>F)
```

```
1      38 8.6091
```

```
2      33 5.5018  5     3.1074 3.7276 0.008743 **
```

```
---
```

Summary so far

- t -tests can be used to test the equality of **2** population means.
- ANOVA can be used to test the equality of **more than 2** population means.
- **Linear regression** provides a general framework for modeling the relationship between response variable and different type of explanatory variables.
- **t -tests** can be used to test the significance of *individual* coefficients.
- **F -tests** can be used to test the *simultaneous significance of multiple* coefficients. We need it to test the association between a response and categorical variables.

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

This gives us a VERY FLEXIBLE framework!!

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

**1 categorical
covariate**

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

**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**

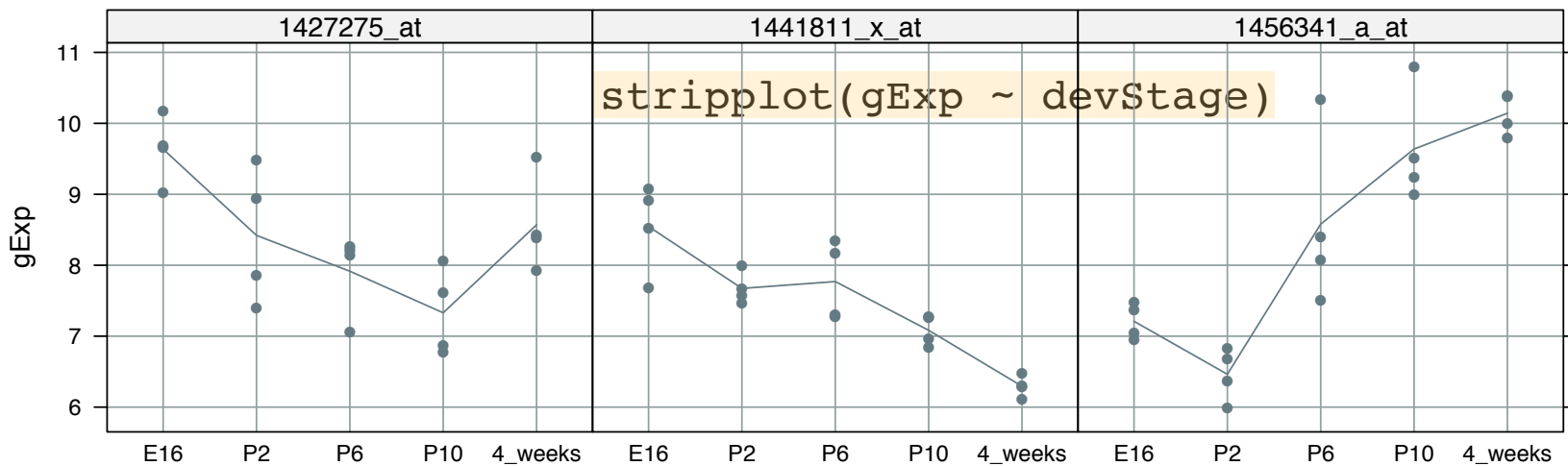
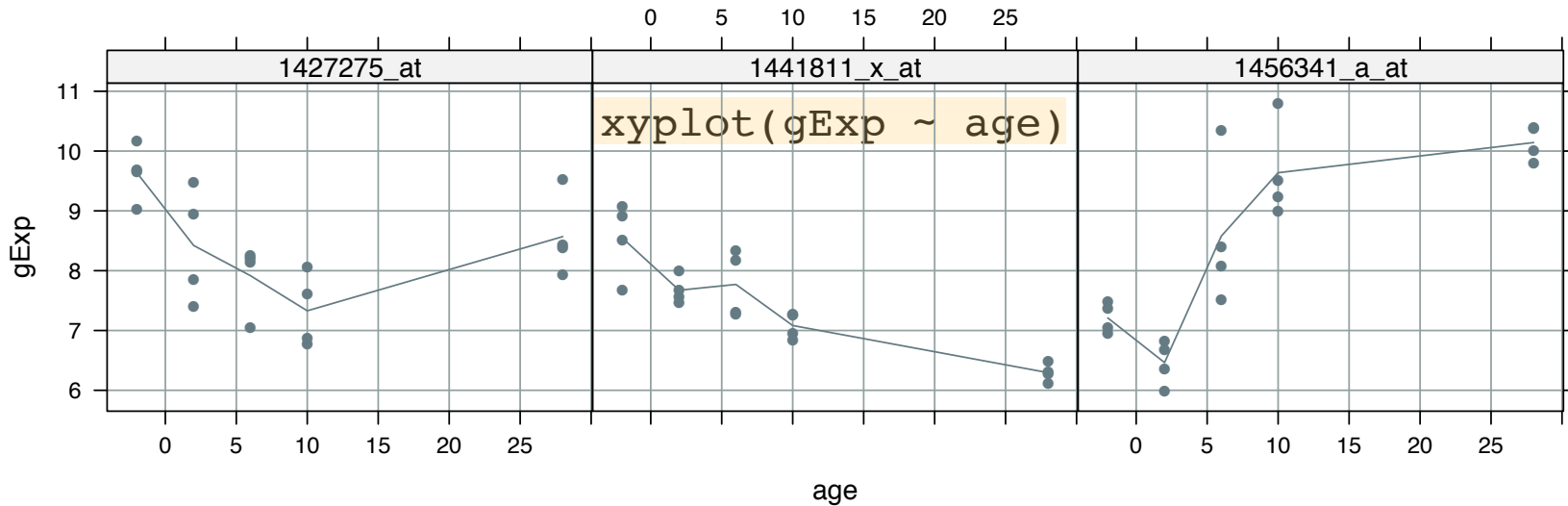
$$\begin{bmatrix} 1 & 0 & 1.22 & 0 \\ 1 & 0 & 2.02 & 0 \\ 1 & 0 & 1.42 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1.89 & 0 \\ 1 & 1 & 2.01 & 2.01 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1.56 & 1.56 \\ 1 & 1 & 2.17 & 2.17 \\ 1 & 1 & 1.51 & 1.51 \end{bmatrix}$$

**1 continuous
1 categorical**

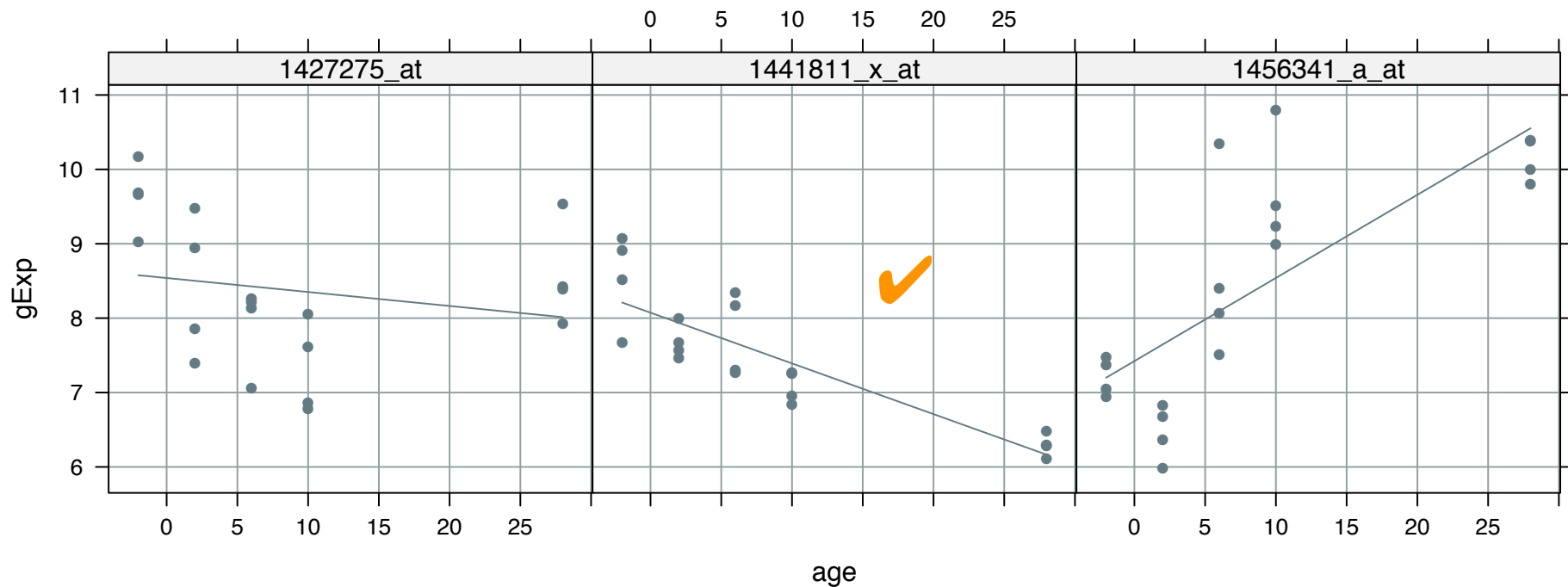
AND MANY MORE

Tip: ?model.matrix

Age as a continuous variable ...



Kind of a different look to the data, no?

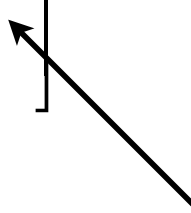


linear looks reasonable for 1, but
not the other two

For now, we'll just assume a linear fit is good enough.
We'll come back to relax this later.

Plain vanilla linear model, matrix formulation

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

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


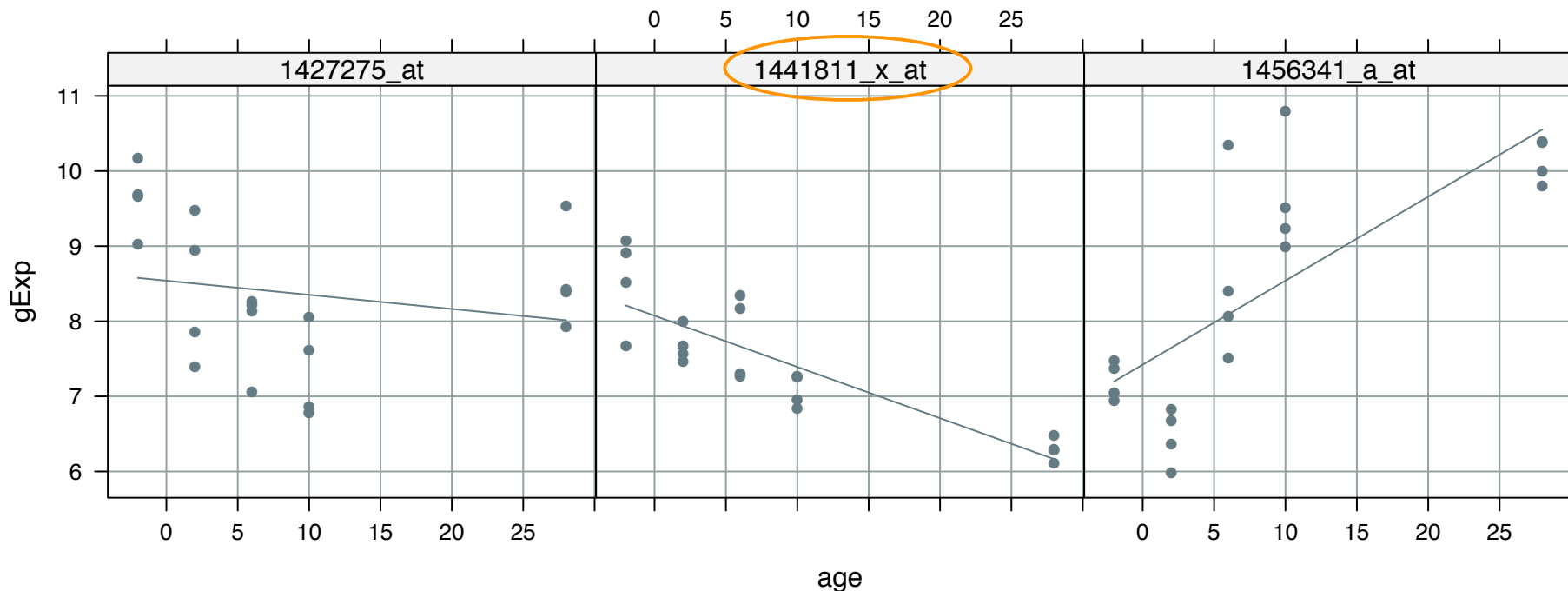
Here's what a design matrix would look like with 1 quantitative covariate.

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

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

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

Remember / convince yourself that the matrix algebra does indeed reproduce simple linear regression.



```
> summary(linFits[["1441811_x_at"]])
```

Call:

```
lm(formula = gExp ~ age, data = z)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.55059	-0.37459	-0.08398	0.31011	0.86827

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	8.073374	0.133118	60.648	< 2e-16 ***
age	-0.068179	0.009771	-6.978	1.62e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

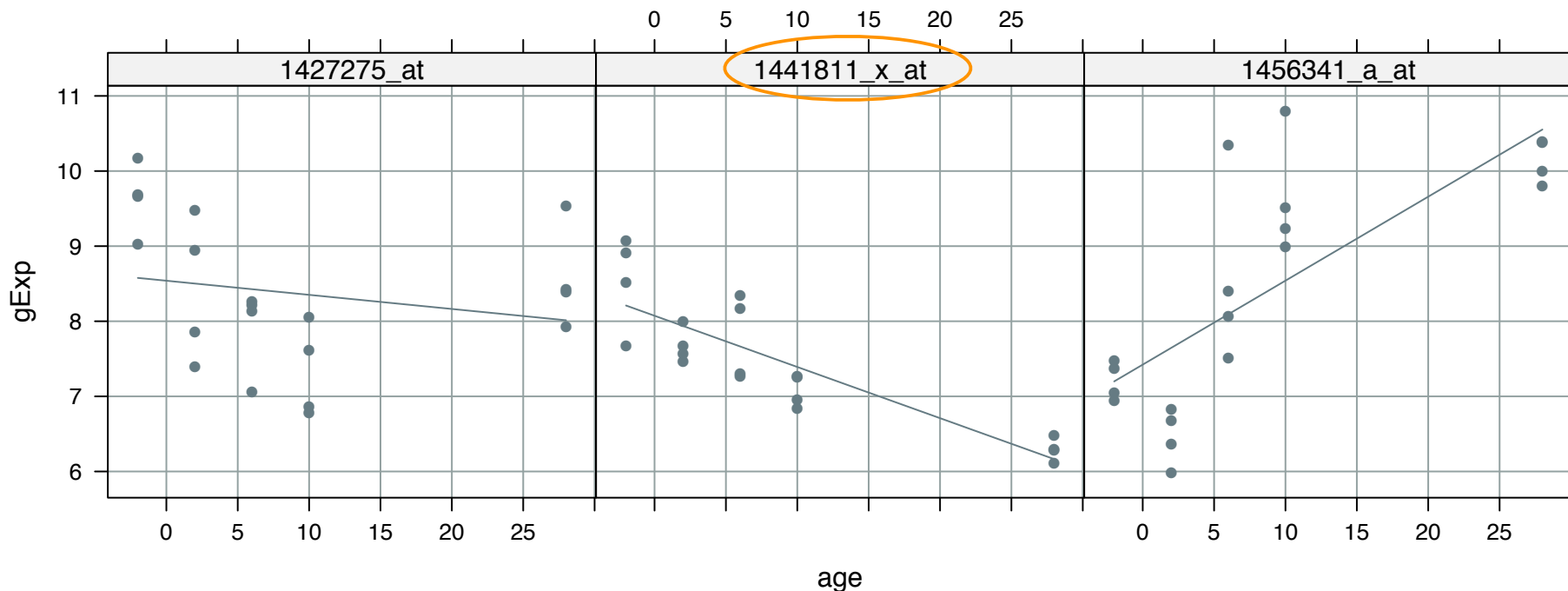
Residual standard error: 0.4545 on 18 degrees of freedom

Multiple R-squared: 0.7301, Adjusted R-squared: 0.7151

F-statistic: 48.69 on 1 and 18 DF, p-value: 1.622e-06

(usually, not of interest)

$$H_0 : \alpha_0 = 0$$



```
> summary(linFits[["1441811_x_at"]])
```

Call:

```
lm(formula = gExp ~ age, data = z)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.55059	-0.37459	-0.08398	0.31011	0.86827

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	8.073374	0.133118	60.648	< 2e-16 ***
age	-0.068179	0.009771	-6.978	1.62e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4545 on 18 degrees of freedom
 Multiple R-squared: 0.7301, Adjusted R-squared: 0.7151
 F-statistic: 48.69 on 1 and 18 DF, p-value: 1.622e-06

Tests the association
between gExp and age

$$H_0 : \alpha_1 = 0$$

How do we estimate the intercept and the slope?

Is there an optimal line?

Call:

```
lm(formula = gExp ~ age, data = z)
```

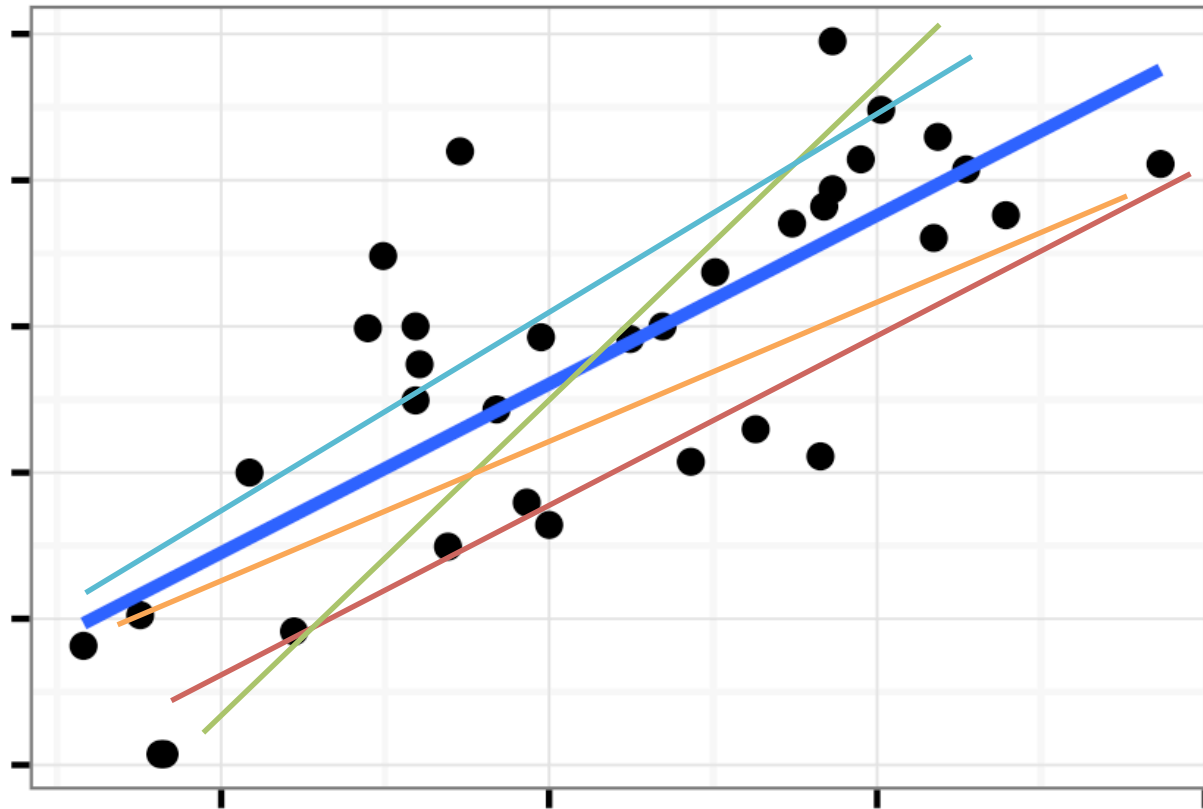
Residuals:

Min	1Q	Median	3Q	Max
-0.55059	-0.37459	-0.08398	0.31011	0.86827

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	8.073374	0.133118	60.648	< 2e-16	***
age	-0.068179	0.009771	-6.978	1.62e-06	***

Which one is the best line?





The error is the vertical distance between the line and the real observation

Ordinary least squares (OLS) estimates of the parameters minimize the sum of squares of the errors

Ordinary Least Square Estimator

Visual representation of the squared errors

<http://setosa.io/ev/ordinary-least-squares-regression/>

- The squares of the errors are represented by squared areas in the second plot:
 - select different lines by changing the intercept and the slope
 - see how the squares of the errors change
 - Which line minimizes the sum of these areas? OLS answers this question
- Move a point of the first plot along the line and away from the line. See how sensitive is the estimation.

Ordinary Least Square (OLS) Estimator for Simple Regression (1 covariate)


Mathematically:

$$y_i = \alpha_0 + \alpha_1 x_i + \boxed{\varepsilon_i}, \quad i = 1, \dots, n$$

error

We want to find a line (i.e., an intercept and a slope) such that the sum of the squared errors is minimized

$$S(\alpha_0, \alpha_1) = \sum_{i=1}^n \underbrace{(y_i - \alpha_0 - \alpha_1 x_i)}_{\text{error}}^2$$

 S is a function of your parameters, usually called the objective function.

error
(solve for epsilon in the equation above)

OLS for Multiple Linear Regression (many covariates)

In matrix notation:

$$\begin{aligned} S(\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_p) &= \sum_{i=1}^n (y_i - \alpha_0 - \alpha_1 x_{i1} - \alpha_2 x_{i2} - \dots - \alpha_p x_{ip})^2 \\ &= (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}) \end{aligned}$$

We need to find values of betas that minimize the sum of squares:

$$\frac{\partial S}{\partial \boldsymbol{\alpha}} = \begin{bmatrix} \frac{\partial S}{\partial \alpha_0} \\ \frac{\partial S}{\partial \alpha_1} \\ \vdots \\ \frac{\partial S}{\partial \alpha_i} \\ \vdots \\ \frac{\partial S}{\partial \alpha_p} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Greatest Hits of Regression Results

(normal iid errors)

Not needed for OLS.
If true OLS is also the MLE

$$Y = X\alpha + \varepsilon \quad \text{regression model}$$

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

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

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

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

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

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

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

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

Inference in Regression (normal iid errors)

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

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

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

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

How test $H_0 : \alpha_j = 0$?

With a t-statistic. Under H_0 ,

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

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

Inference in Regression

(normal iid errors)

What if this assumption
does not hold??

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

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

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

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

How test $H_0 : \alpha_j = 0$?

With a t-statistic. Under H_0 , asymptotically (CLT)

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

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

- The nature of the regression function $f(x; \alpha)$ is one of the defining characteristics of a regression model
 - f linear in $\alpha \Rightarrow$ linear model
 - f not linear in $\alpha \Rightarrow$ nonlinear model

nonlinear parametric regression

$$Y = \frac{1}{1 + e^{(\phi - x)/\xi}} + \varepsilon$$

simple linear regression (a linear model)

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

What we just did.



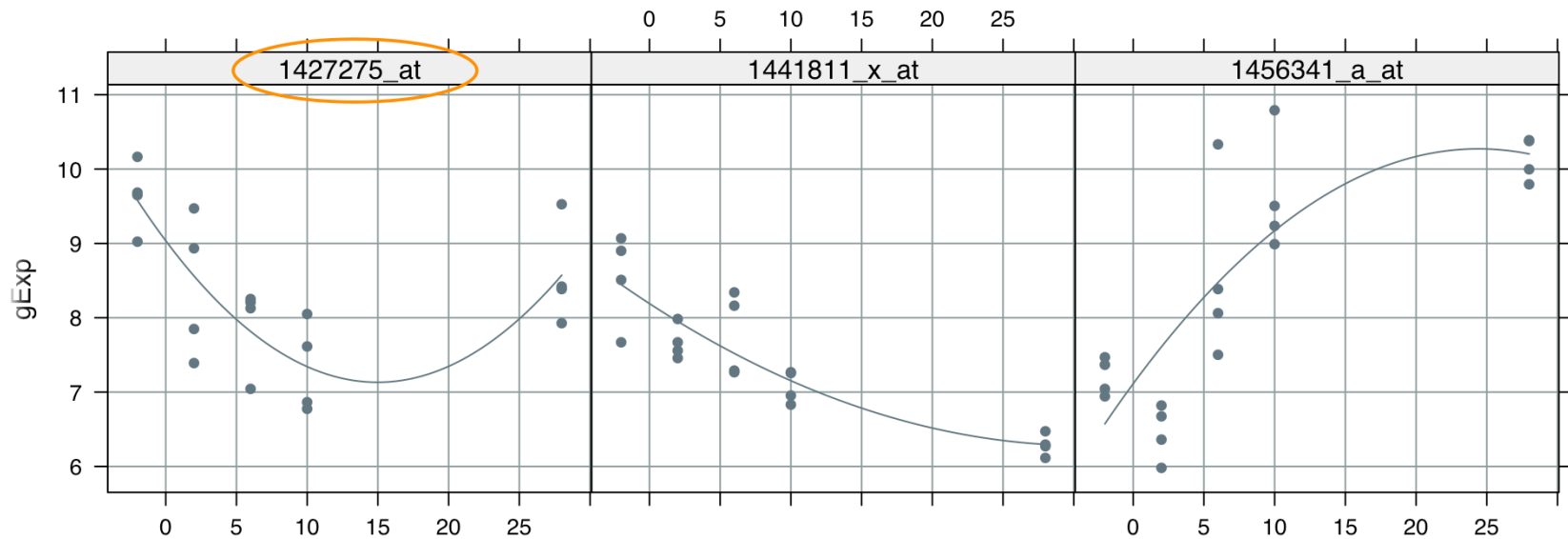
polynomial regression (also a linear model)

$$Y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \varepsilon$$

What we're
about to do.



$$Y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \varepsilon$$



```
> summary(quadFits[["1427275_at"]])
```

age

Call:

```
lm(formula = gExp ~ age + I(age^2), data = z)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.16275	-0.55506	0.09503	0.40804	0.95803

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	9.036401	0.212313	42.562	< 2e-16 ***
age	-0.254305	0.048125	-5.284	6.07e-05 ***
I(age^2)	0.008490	0.001661	5.110	8.71e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6444 on 17 degrees of freedom

Multiple R-squared: 0.6218, Adjusted R-squared: 0.5773

F-statistic: 13.98 on 2 and 17 DF, p-value: 0.0002572

- The nature of the regression function $f(x; \alpha)$ is one of the defining characteristics of a regression model
 - f linear in $\alpha \Rightarrow$ linear model
 - f not linear in $\alpha \Rightarrow$ nonlinear model

polynomial regression (also a linear model)

$$Y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \varepsilon$$

NOTE: This is a linear model, because it is linear in the alphas. It is easy but wrong to focus on the x 's and mistake this for a nonlinear model.

Conclusions

linear model framework is extremely general!

one extreme (simple): two-sample common variance t-test

another extreme (flexible): a polynomial, potentially different for each level of some factor

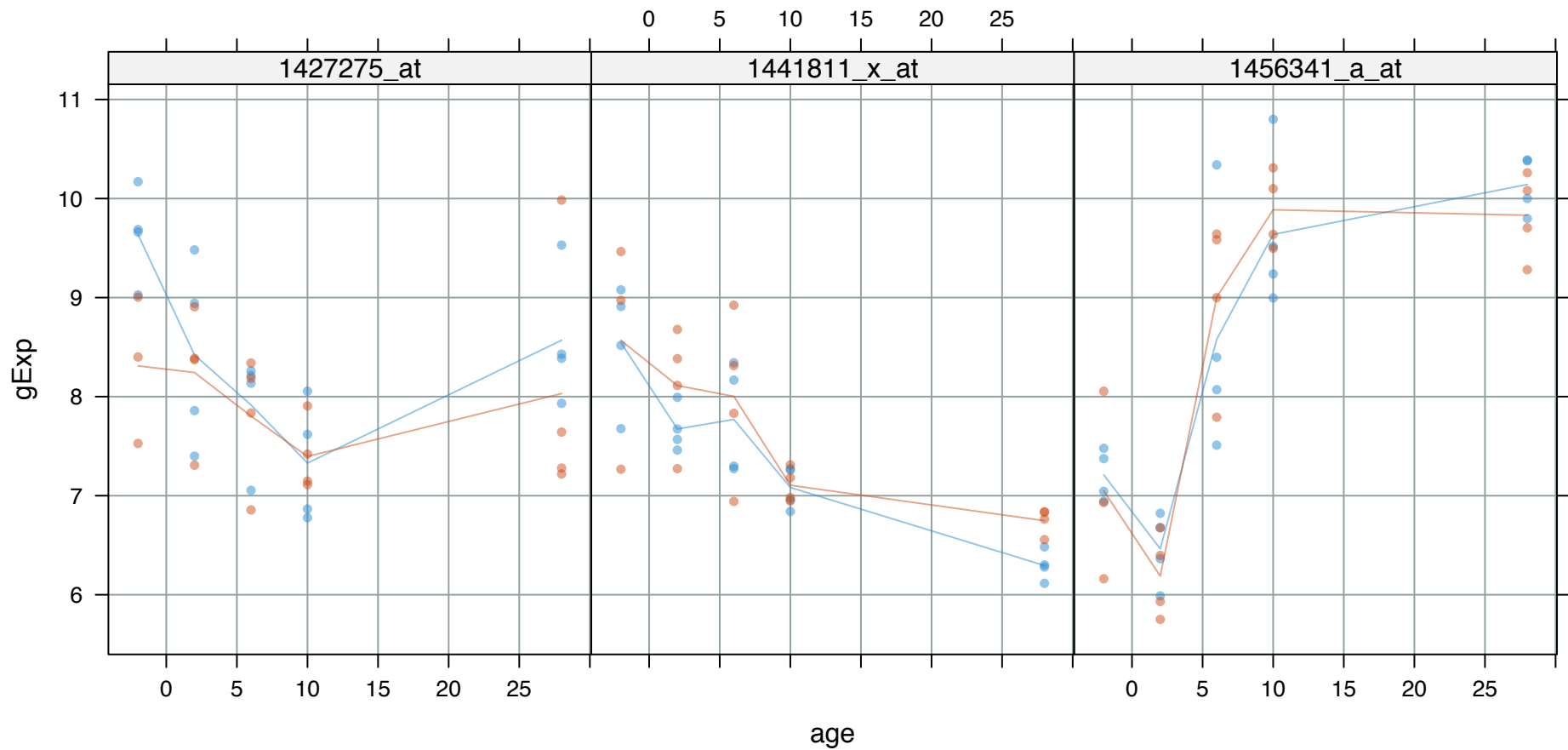
dichotomous variable? OK!

categorical variable? OK!

quantitative variable? OK!

various combinations of the above? OK!

don't be afraid to build models with more than 1 covariate



What about the other 29,946 probesets?

To be continued