Class 8:
General linear model I

Alison Presmanes Hill

Reading for today: Chapter 8

Reading for next class: Chapter 10 (note: we are skipping Chapter 9)

Change of notation

X	Y
Independent variable (IV)	Dependent variable (DV)
Explanatory variable	Response variable
Predictor variable	Outcome variable

Data are paired observations of *X* and *Y*

$$(x_1, y_1), \ldots, (x_n, y_n)$$

Anscombe's quartet

- 4 datasets
- 11 paired observations per set
- Observed data of rvs *X* and *Y*

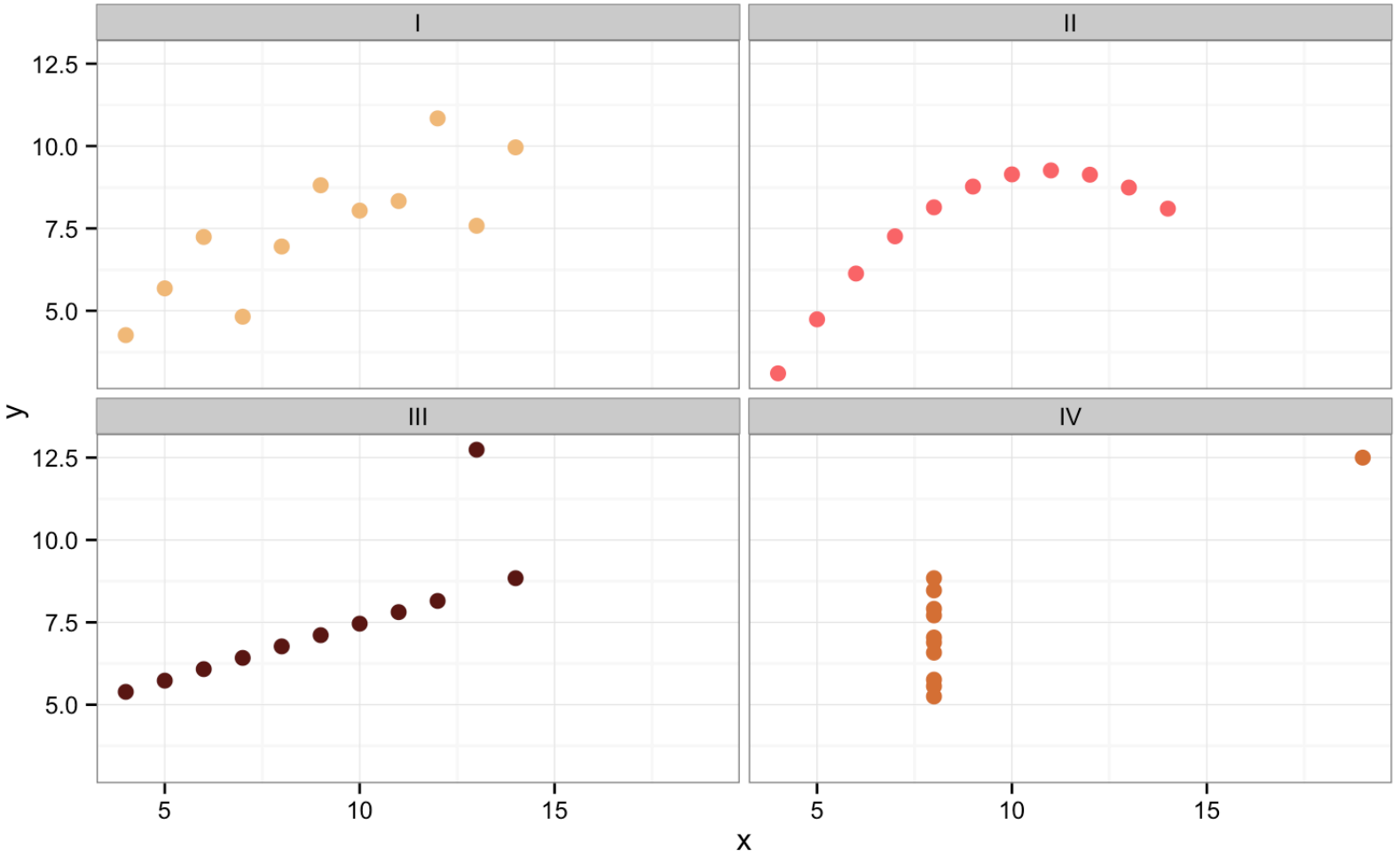
	x1	x2	x3	x4	y1	y2	y3	y4
1	10	10	10	8	8.04	9.14	7.46	6.58
2	8	8	8	8	6.95	8.14	6.77	5.76
3	13	13	13	8	7.58	8.74	12.74	7.71
4	9	9	9	8	8.81	8.77	7.11	8.84
5	11	11	11	8	8.33	9.26	7.81	8.47
6	14	14	14	8	9.96	8.10	8.84	7.04
7	6	6	6	8	7.24	6.13	6.08	5.25
8	4	4	4	19	4.26	3.10	5.39	12.50
9	12	12	12	8	10.84	9.13	8.15	5.56
10	7	7	7	8	4.82	7.26	6.42	7.91
11	5	5	5	8	5.68	4.74	5.73	6.89

Anscombe's quartet

- 4 datasets
- 11 paired observations per set
- Observed data of rvs *X* and *Y*
- Exact same descriptive statistics
- Exact same correlation between X and Y

	set	n	x_mean	x_sd	y_mean	y_sd	cor
1	I	11	9	3.316625	7.5	2.031568	0.816
2	II	11	9	3.316625	7.5	2.031657	0.816
3	III	11	9	3.316625	7.5	2.030424	0.816
4	IV	11	9	3.316625	7.5	2.030579	0.817

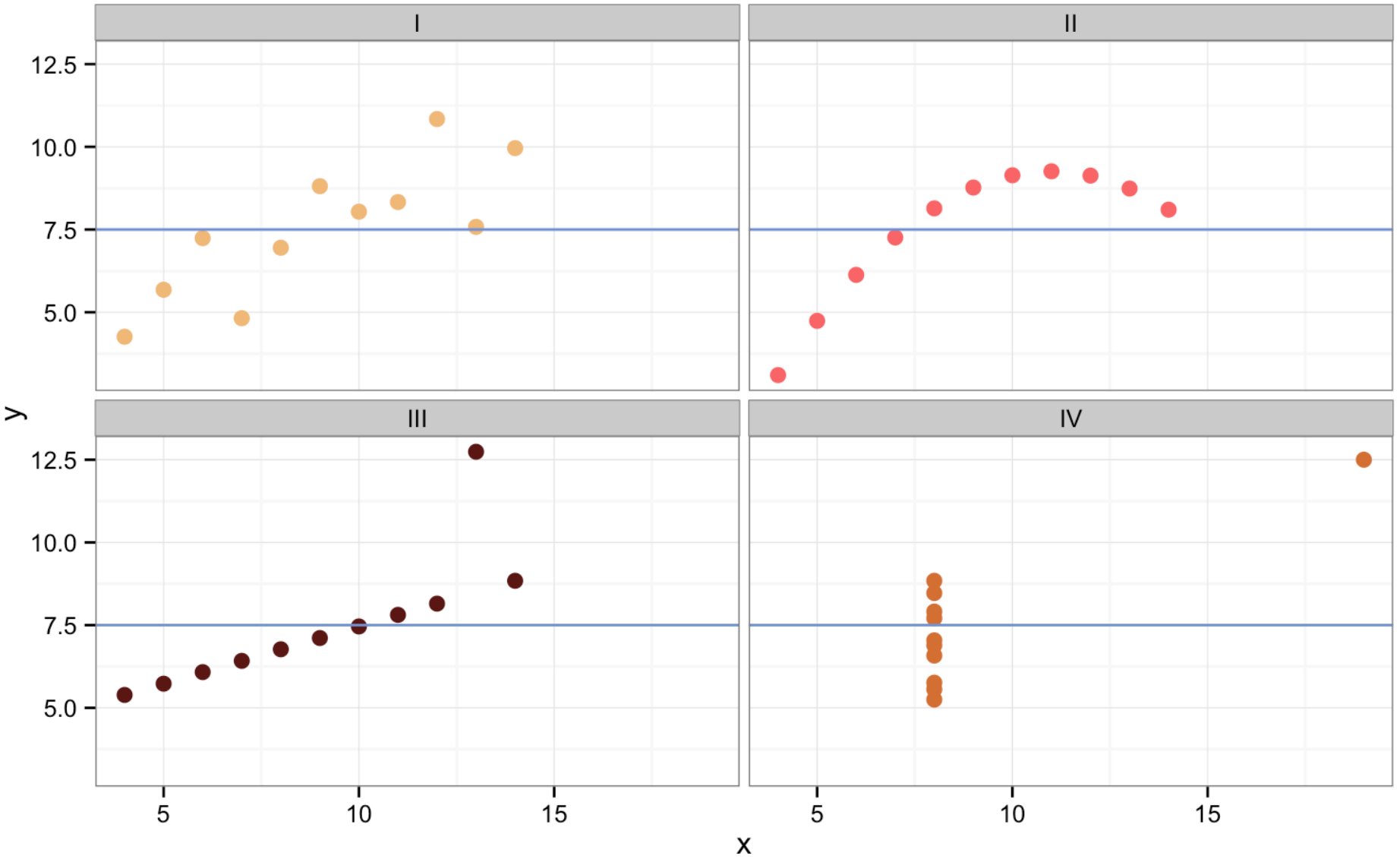
Anscombe's quartet



THE SIMPLEST MODEL EVER

Hint #1: it is linear

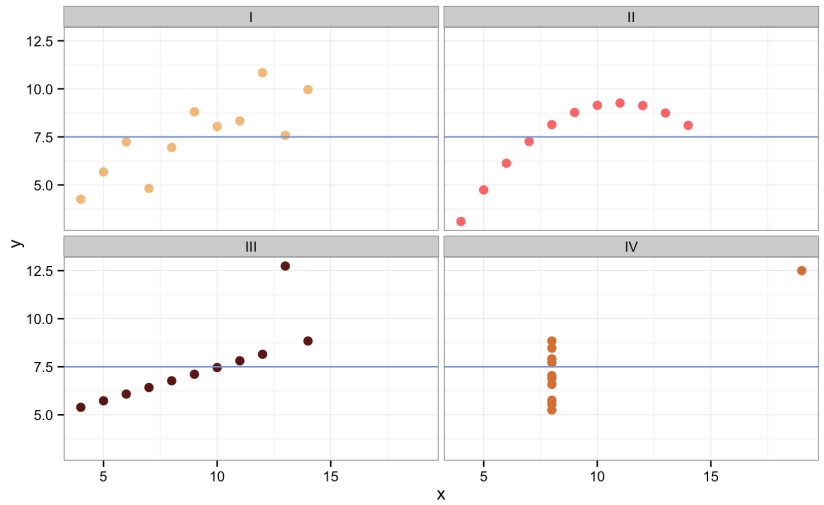
Hint #2: its slope = 0


I
II
III
IV
12.5
10.0
7.5
5.0
y
5
10
15
x

The simplest linear model ever…

- Is a horizontal line!
- Intercept = mean; slope = 0
- Decomposing an rv into its mean and what is left over, that is, the mean + prediction error of the mean

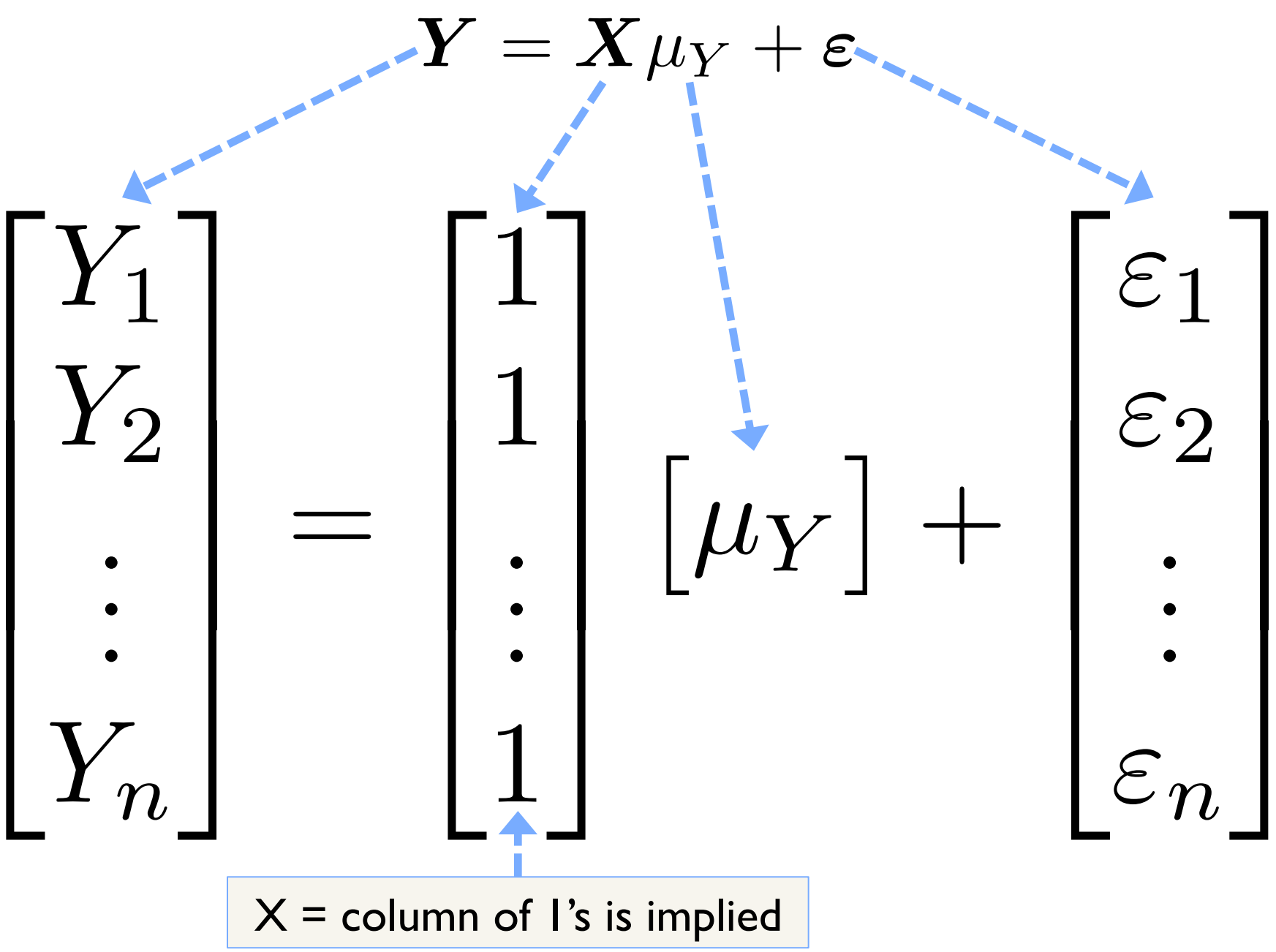
$$Y = \mu + \varepsilon, \text{where } \varepsilon \sim F, E(\varepsilon) = 0$$

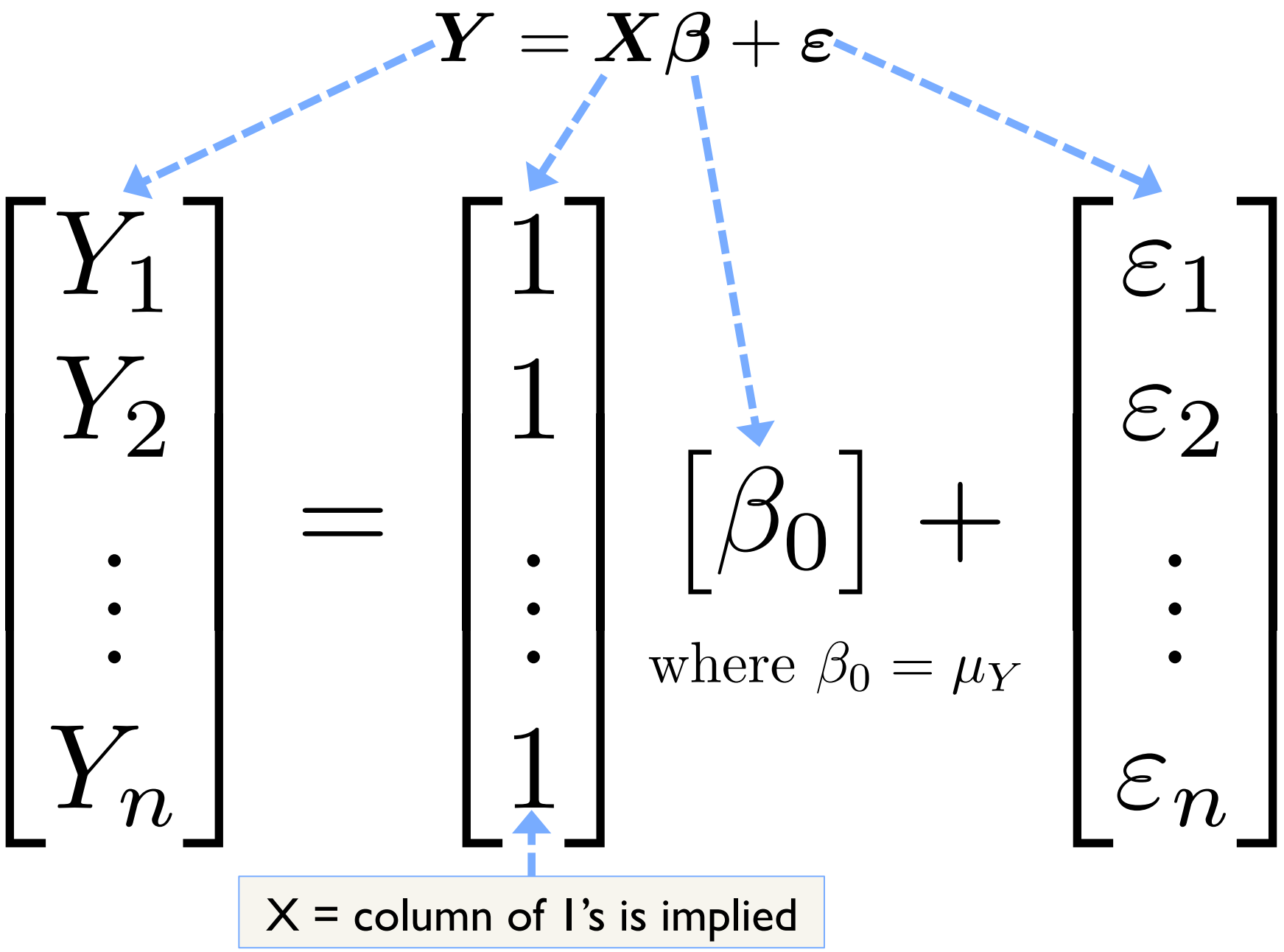


The mean as a matrix

- Where ***Y*** is the rv vector of length n
- ***X*** is a matrix with 1 column and n rows, with only values of 1 (this is column is implied)
  - Note that number of columns in *X* must match number of rows in β
- β is a 1x1 matrix, where the one value in this matrix equals μ
- ε is an rv- the error vector- of length n

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

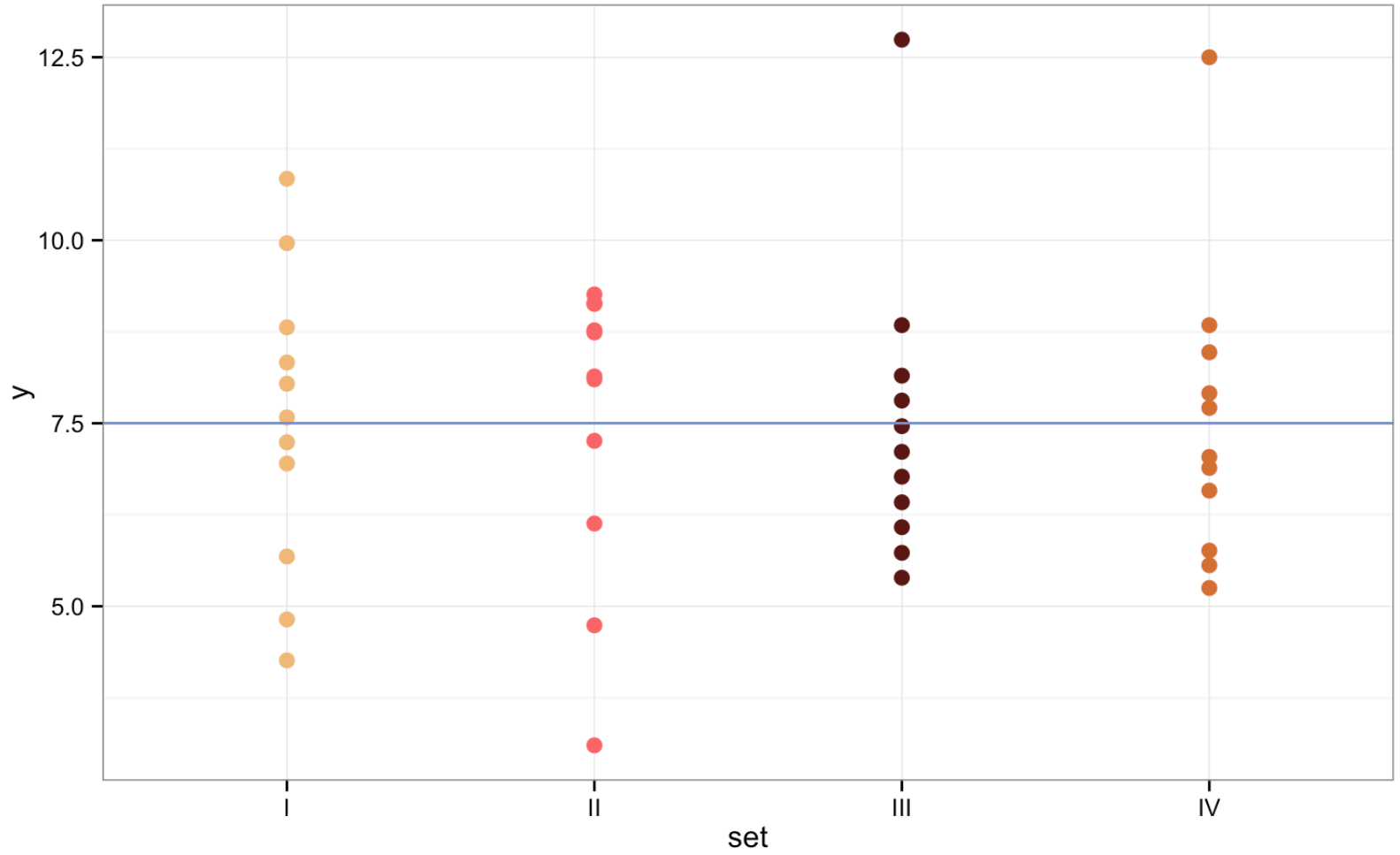

$\boldsymbol{Y} = \boldsymbol{X}\mu_Y + \boldsymbol{\varepsilon}$
$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} [\mu_Y] + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$
X = column of 1's is implied

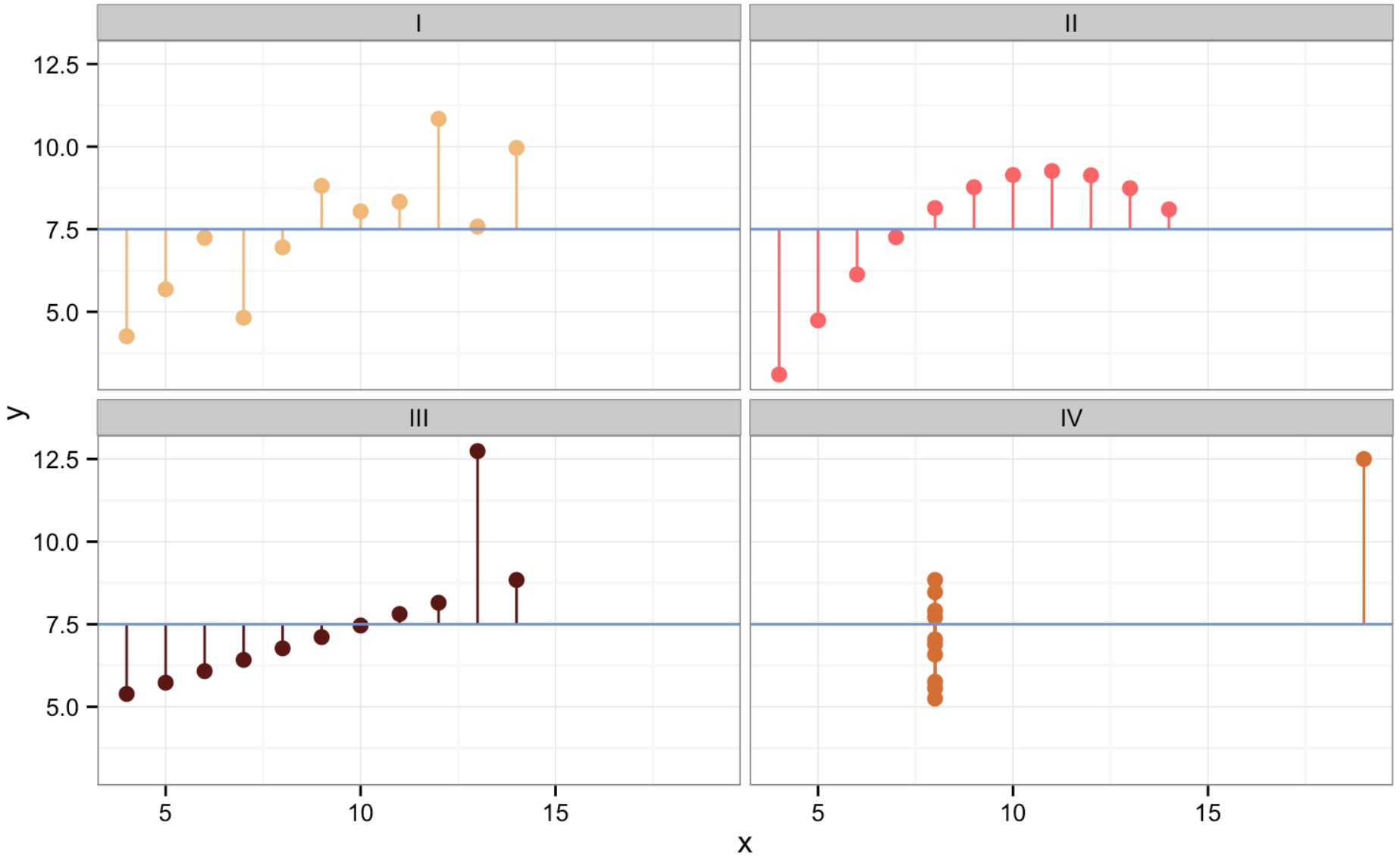

$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$
$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} [\beta_0] + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$
where $\beta_0 = \mu_Y$
X = column of 1's is implied

THE SIMPLEST MODEL EVER

How do we measure how
"good" the fit is of this model?

Some "distance" of each y from the mean?





Observed?
Model?

Total sums of squares (TSS) for *Y*

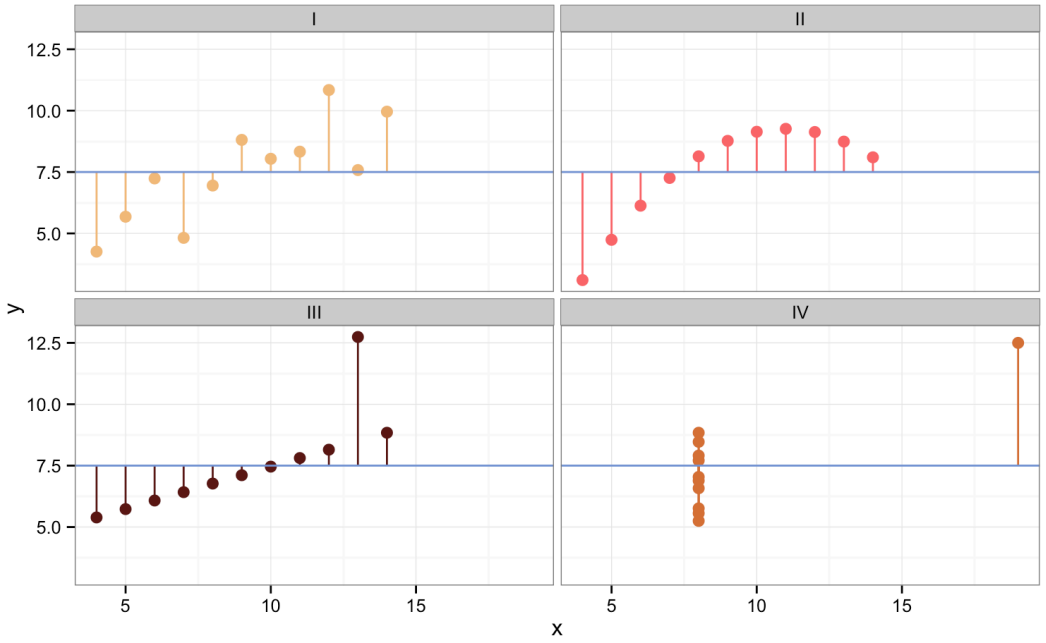
observed – model

$$\sum \varepsilon_i'^2 = \sum (y_i - \overline{y})^2$$

Evaluating fit of the mean line

- How do we measure how "good" the fit of this line is?
- Our "error of prediction" when our model is the mean of *Y*

```
> obs_sum %>%
+     group_by(set) %>%
+     summarise(tot_ss = sum((y - mean(y))^2))
Source: local data frame [4 x 2]

  set   tot_ss
1   I 41.27269
2  II 41.27629
3 III 41.22620
4  IV 41.23249
```



THE 2ND SIMPLEST MODEL EVER

"Before you do support vector regression, why don't you do something really stupid-- and by really stupid, I mean like linear regression."
- Jan van Santen

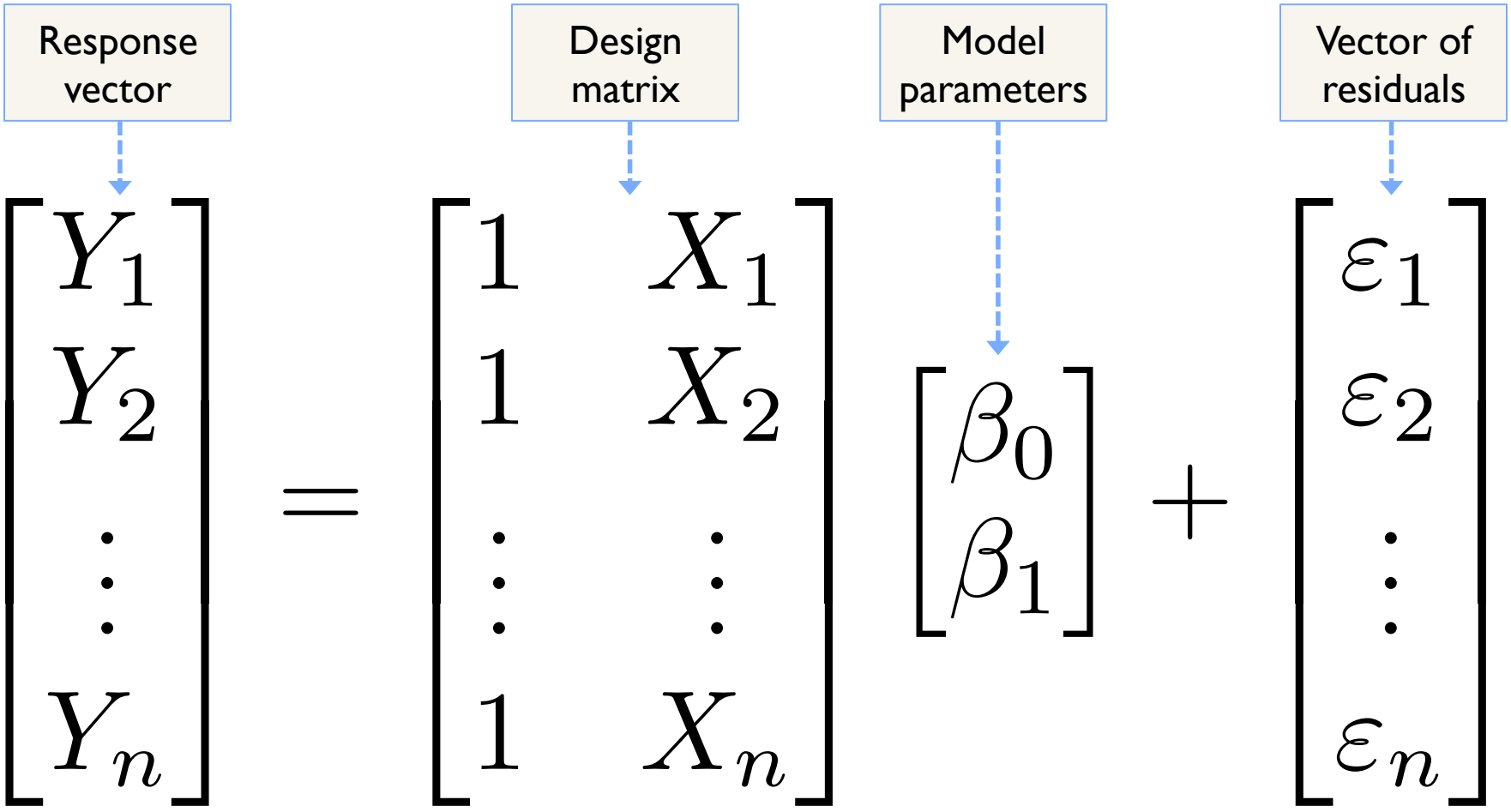
Simple linear regression (SLR): a straight line

$$Y = \beta_0 + \beta_1 x + \varepsilon$$

SLR: a straight line

$$Y = \overbrace{\beta_0}^{intercept} + \overbrace{\beta_1}^{slope} x + \overbrace{\varepsilon}^{residuals}$$

SLR: matrix notation



For > 1 predictors, note that number of columns in design matrix must equal number of rows of model parameters.

Least squares criterion

$$g(\beta_0, \beta_1) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum (Y_i - \beta_0 - \beta_1 x_i)^2$$

Betas must minimize this sum

Linear models in R

$$Y = \beta_0 + \beta_1 x + \varepsilon$$

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

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

(R formulas are expressed in 'Wilkinson-Rogers' notation)

```
> model_one <- lm(y ~ x, data = anscombe_one) # one lm model
> summary(model_one)

Call:
lm(formula = y ~ x, data = anscombe_one)

Residuals:
     Min       1Q   Median       3Q      Max
-1.92127 -0.45577 -0.04136  0.70941  1.83882

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0001     1.1247   2.667  0.02573 *
x             0.5001     0.1179   4.241  0.00217 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.237 on 9 degrees of freedom
Multiple R-squared:  0.6665,    Adjusted R-squared:  0.6295
F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

$$\hat{y} = 3.0 + 0.5x$$

```
> 4.241^2
[1] 17.98608
```

```
> # str(model_one)
> tidy(model_one) # library(broom)
         term  estimate std.error statistic     p.value
1 (Intercept) 3.0000909 1.1247468  2.667348 0.025734051
2           x 0.5000909 0.1179055  4.241455 0.002169629
```

Quick aside: `library(broom)`

This package provides three S3 methods that do three distinct kinds of tidying.

1. tidy: constructs a data frame that summarizes the model's statistical findings. This includes coefficients and p-values for each term in a regression, per-cluster information in clustering applications, or per-test information for multtest functions.
2. augment: add columns to the original data that was modeled. This includes predictions, residuals, and cluster assignments.
3. glance: construct a concise *one-row* summary of the model. This typically contains values such as R^2, adjusted R^2, and residual standard error that are computed once for the entire model.

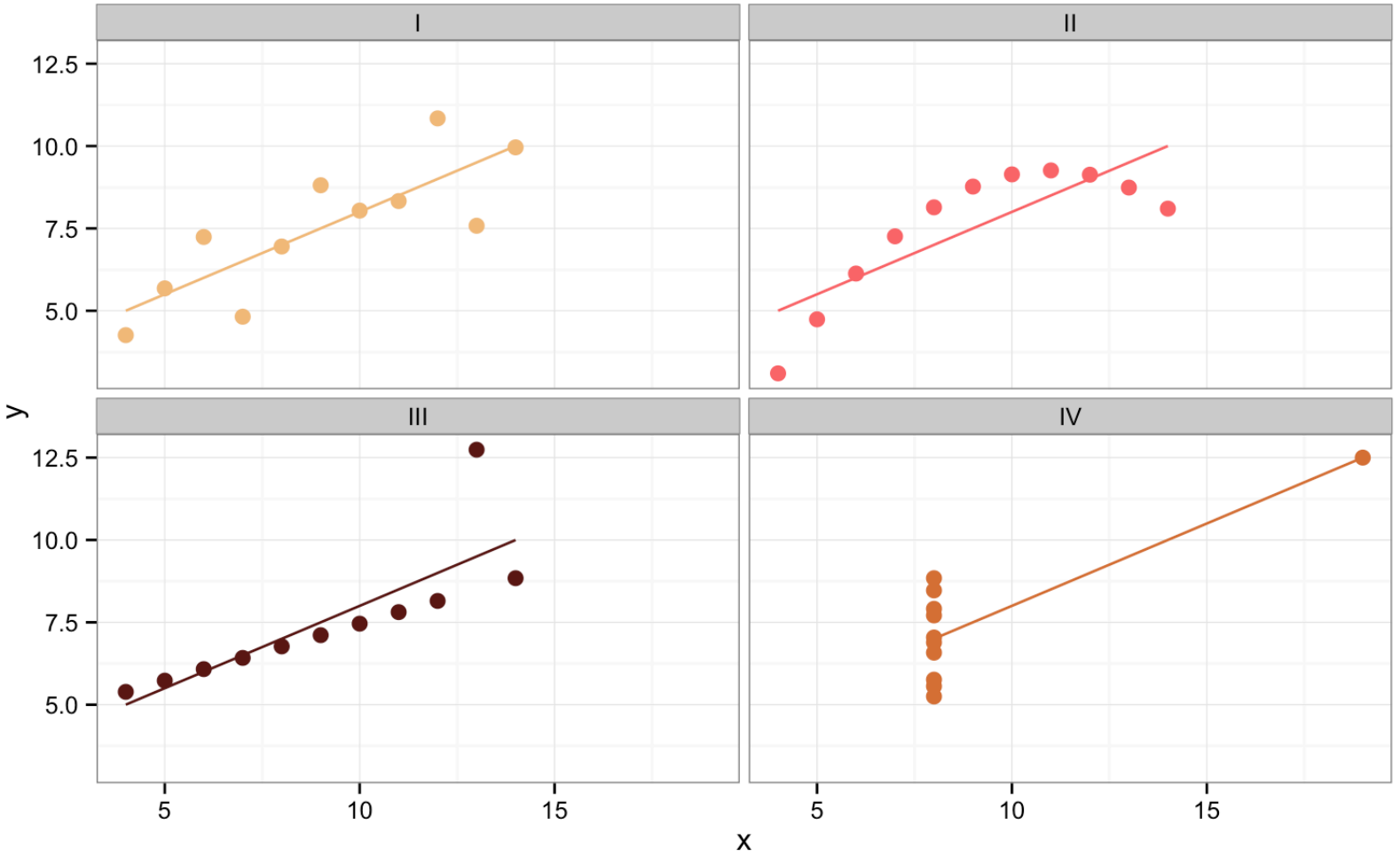
```
> models <- anscombe_tidy %>%
+   group_by(set) %>%
+   do(mod = lm(y ~ x, data = .))
> models
Source: local data frame [4 x 2]
Groups: <by row>

  set     mod
1   I <S3:lm>
2  II <S3:lm>
3 III <S3:lm>
4  IV <S3:lm>
> coefs <- models %>% tidy(mod) # library(broom)
> coefs # many lm models
Source: local data frame [8 x 6]
Groups: set
```

$$\hat{y} = 3.0 + 0.5x$$

set	term	estimate	std.error	statistic	p.value
I	(Intercept)	3.0000909	1.1247468	2.667348	0.025734051
I	x	0.5000909	0.1179055	4.241455	0.002169629
II	(Intercept)	3.0009091	1.1253024	2.666758	0.025758941
II	x	0.5000000	0.1179637	4.238590	0.002178816
III	(Intercept)	3.0024545	1.1244812	2.670080	0.025619109
III	x	0.4997273	0.1178777	4.239372	0.002176305
IV	(Intercept)	3.0017273	1.1239211	2.670763	0.025590425
IV	x	0.4999091	0.1178189	4.243028	0.002164602

The linear regression model

$$\hat{y} = 3.0 + 0.5x$$



THE 2ND SIMPLEST MODEL EVER

How do we measure how
"good" the fit is of this model?

Neat side-effects of SLR

- In addition to our observed/actual data $(x_1, y_1), \ldots, (x_n, y_n)$, running a regression model produces the following for every (x_i, y_i):
  - Predicted or "fitted" values of y_i: $\hat{Y}_i$
  - The residuals of y_i: $\mathcal{E}_i$

- Predictably, since both of these are statistics, they each have:
  - Expected values
  - Standard errors (recall this is the standard deviation of the sampling distribution of a statistic)

SLR: actual observed values of Y = predicted + residuals

Actual observed Y_i's

$$Y_i = \underbrace{\beta_0 + \beta_1 x_i}_{} + \underbrace{\varepsilon_i}_{}$$

$$= \underbrace{\hat{Y}_i}_{} + \underbrace{\varepsilon_i}_{}$$

Predicted or fitted Y_i's

residuals

SLR: predicted values of *Y*

$$\hat{Y}_i = \beta_0 + \beta_1 x_i$$

Predicted or fitted Y_i's

Expectation and variance of predicted values

Expectation:
The conditional mean of *Y* is linear in *X*, with an intercept of β_0 and a slope of β_1

$$E(Y_i|X = x_i) = \beta_0 + \beta_1 x_i$$

Variance
The conditional variance of *Y* is constant with respect to *X*

$$Var(Y|X = x) = \sigma^2$$

SLR: residual values of *Y* (reversing previous formula)

Actual observed Y_i's

$$\varepsilon_i = Y_i - \hat{Y}_i$$

$$= Y_i - (\beta_0 + \beta_1 x_i)$$

Predicted or fitted Y_i's

Expectation and variance of residual values

Expectation:

The values of the residuals are unrelated to X, such that if we plot the residuals vs. the x's, we see a null scatterplot with no patterns

$$E(\varepsilon_i) = E(\varepsilon|x_i) = 0$$

Variance

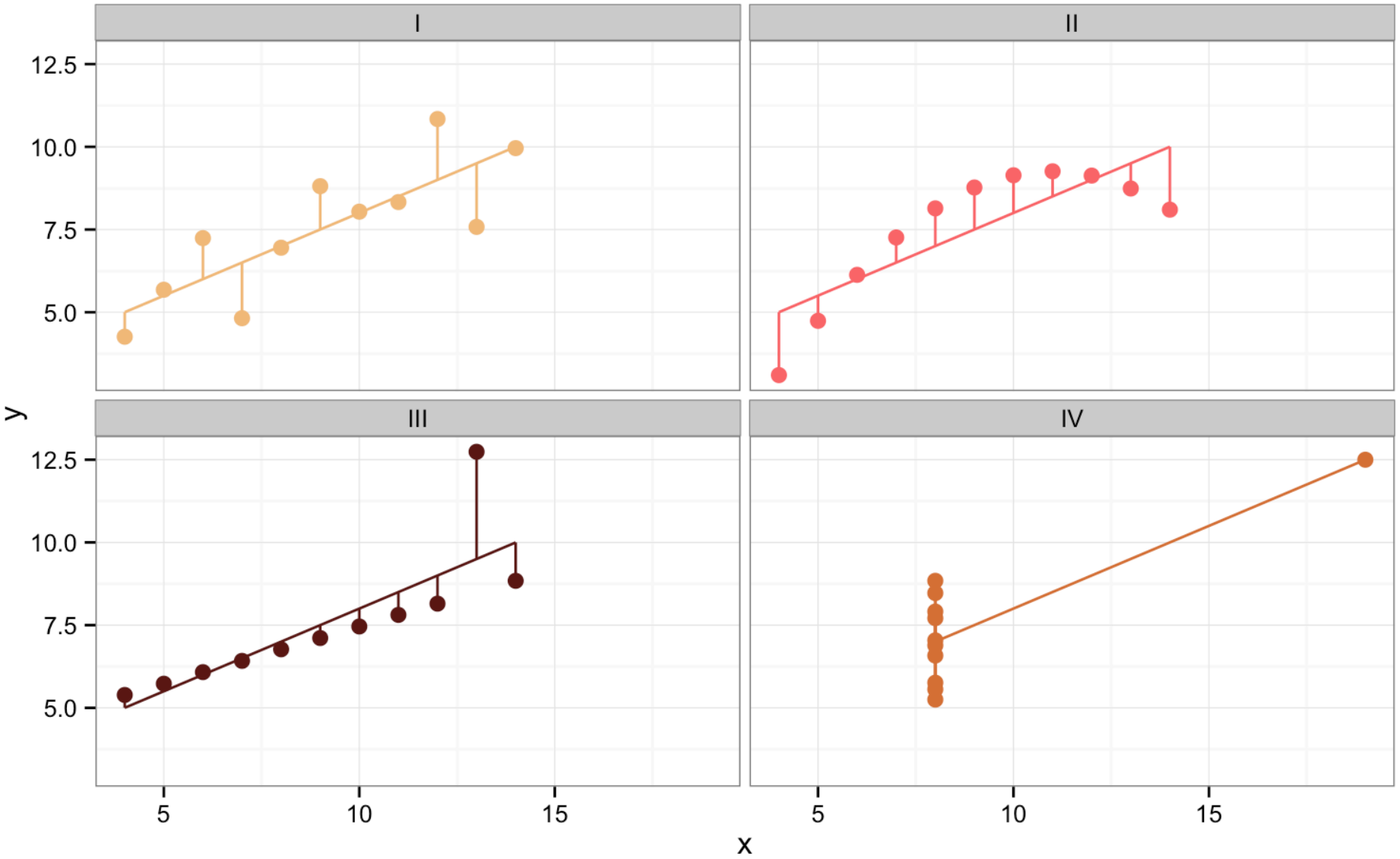
The conditional variance of the residuals is constant with respect to X

$$Var(\varepsilon_i) = Var(\varepsilon|x_i) = Var(Y|x_i) = \sigma_\varepsilon^2$$

Residual standard error

$$SE_{resid} = \sqrt{\frac{\sum E_i^2}{(n-2)}}$$

$$= \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{(n-2)}}$$



Observed?
Model?

Residual sums of squares* (RSS) for *Y*

observed – model

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

*Your text calls this the sums of squares for errors (SSE)- try to mentally switch to the residual sums of squares terms, as their notation will confuse you eventually

```
> model_one <- lm(y ~ x, data = anscombe_one) # one lm model
> augment(model_one) # library(broom)
```

	y	x	.fitted	.se.fit	.resid	.hat	.sigma	.cooksd	.std.resid
1	8.04	10	8.001000	0.3910483	0.03900000	0.10000000	1.311535	0.00006139788	0.03324397
2	6.95	8	7.000818	0.3910483	-0.05081818	0.10000000	1.311479	0.00010424672	-0.04331791
3	7.58	13	9.501273	0.6012024	-1.92127273	0.23636364	1.056460	0.48920927577	-1.77793266
4	8.81	9	7.500909	0.3728499	1.30909091	0.09090909	1.218483	0.06163699895	1.11028824
5	8.33	11	8.501091	0.4411620	-0.17109091	0.12727273	1.310017	0.00159934188	-0.14810075
6	9.96	14	10.001364	0.6975383	-0.04136364	0.31818182	1.311496	0.00038289951	-0.04050923
7	7.24	6	6.000636	0.5139382	1.23936364	0.17272727	1.219936	0.12675648475	1.10190458
8	4.26	4	5.000455	0.6975383	-0.74045455	0.31818182	1.272721	0.12269989634	-0.72515977
9	10.84	12	9.001182	0.5139382	1.83881818	0.17272727	1.099742	0.27902959338	1.63487302
10	4.82	7	6.500727	0.4411620	-1.68072727	0.12727273	1.147055	0.15434122237	-1.45488131
11	5.68	5	5.500545	0.6012024	0.17945455	0.23636364	1.309605	0.00426801143	0.16606601

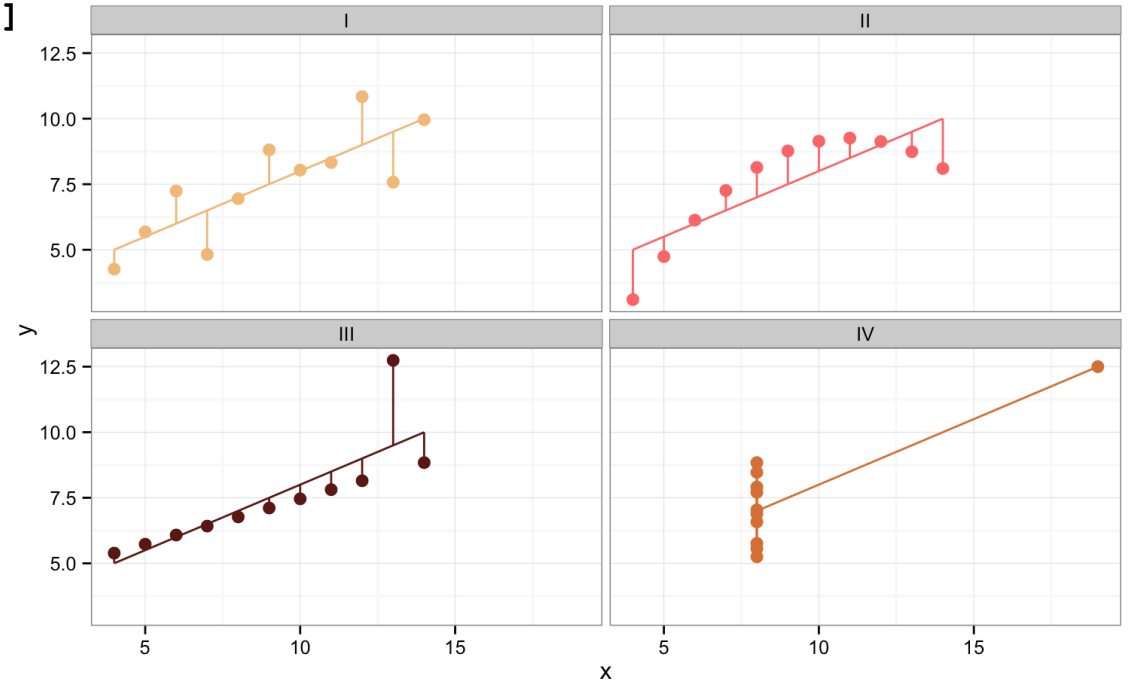
```
> obs_sum <- models %>% augment(mod) # library(broom)
> obs_sum # many lm models
Source: local data frame [44 x 10]
Groups: set
```

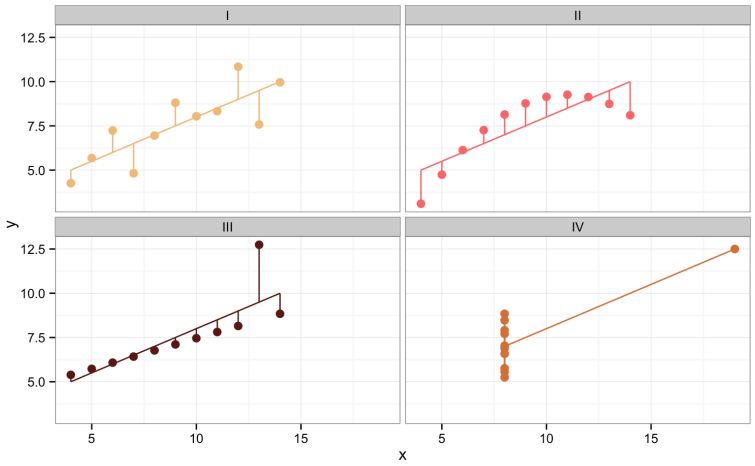
	set	y	x	.fitted	.se.fit	.resid	.hat	.sigma	.cooksd	.std.resid
1	I	8.04	10	8.001000	0.3910483	0.03900000	0.10000000	1.311535	6.139788e-05	0.03324397
2	I	6.95	8	7.000818	0.3910483	-0.05081818	0.10000000	1.311479	1.042467e-04	-0.04331791
3	I	7.58	13	9.501273	0.6012024	-1.92127273	0.23636364	1.056460	4.892093e-01	-1.77793266
4	I	8.81	9	7.500909	0.3728499	1.30909091	0.09090909	1.218483	6.163700e-02	1.11028824
5	I	8.33	11	8.501091	0.4411620	-0.17109091	0.12727273	1.310017	1.599342e-03	-0.14810075
6	I	9.96	14	10.001364	0.6975383	-0.04136364	0.31818182	1.311496	3.828995e-04	-0.04050923
7	I	7.24	6	6.000636	0.5139382	1.23936364	0.17272727	1.219936	1.267565e-01	1.10190458
8	I	4.26	4	5.000455	0.6975383	-0.74045455	0.31818182	1.272721	1.226999e-01	-0.72515977
9	I	10.84	12	9.001182	0.5139382	1.83881818	0.17272727	1.099742	2.790296e-01	1.63487302
10	I	4.82	7	6.500727	0.4411620	-1.68072727	0.12727273	1.147055	1.543412e-01	-1.45488131

Evaluating fit of the regression line

- How do we measure how "good" the fit of this line is?

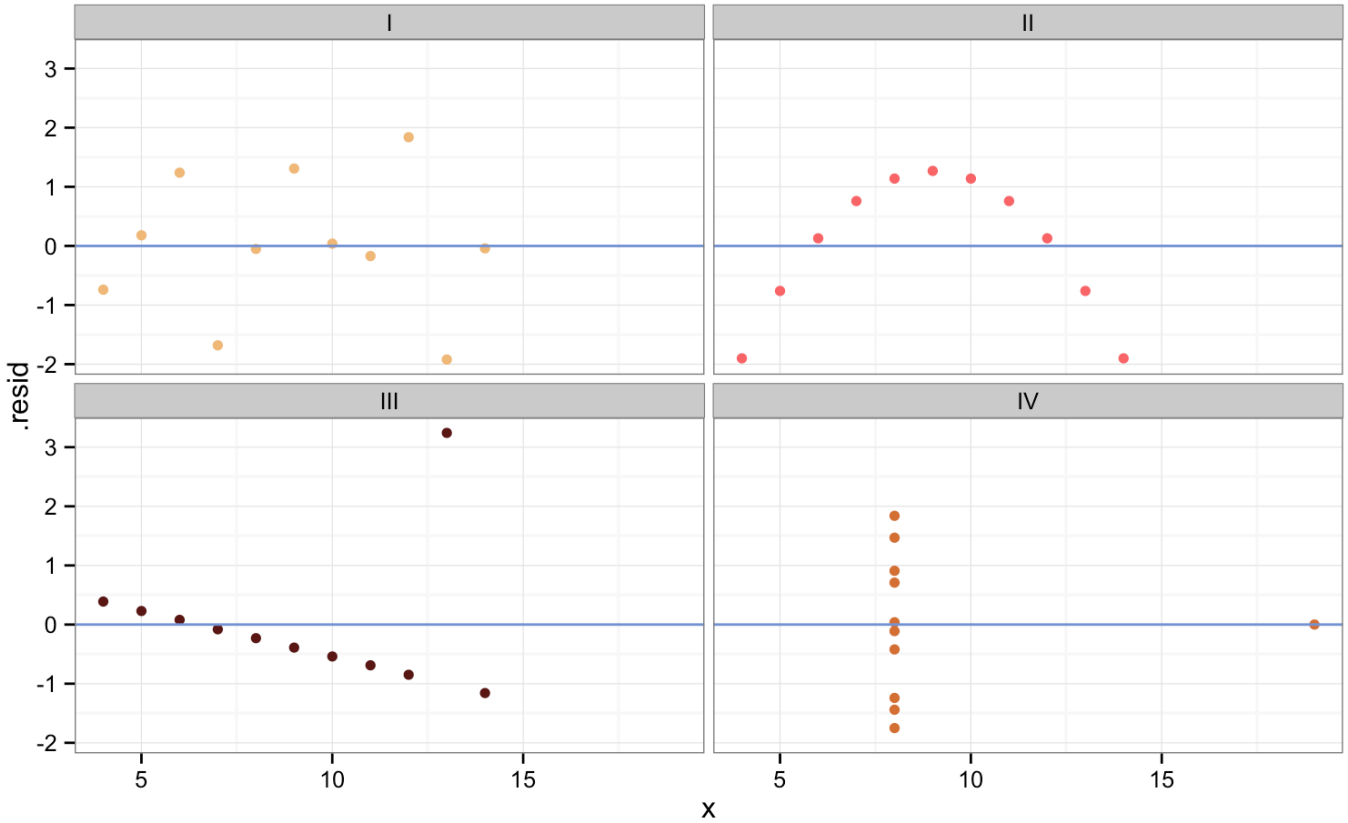
```
> obs_sum %>%
+     group_by(set) %>%
+     summarise(tot_ss = sum((y - mean(y))^2),
+               res_ss = sum((y - .fitted)^2))
Source: local data frame [4 x 3]

  set   tot_ss   res_ss
1   I 41.27269 13.76269
2  II 41.27629 13.77629
3 III 41.22620 13.75619
4  IV 41.23249 13.74249
```





Always plot residuals against *x*!

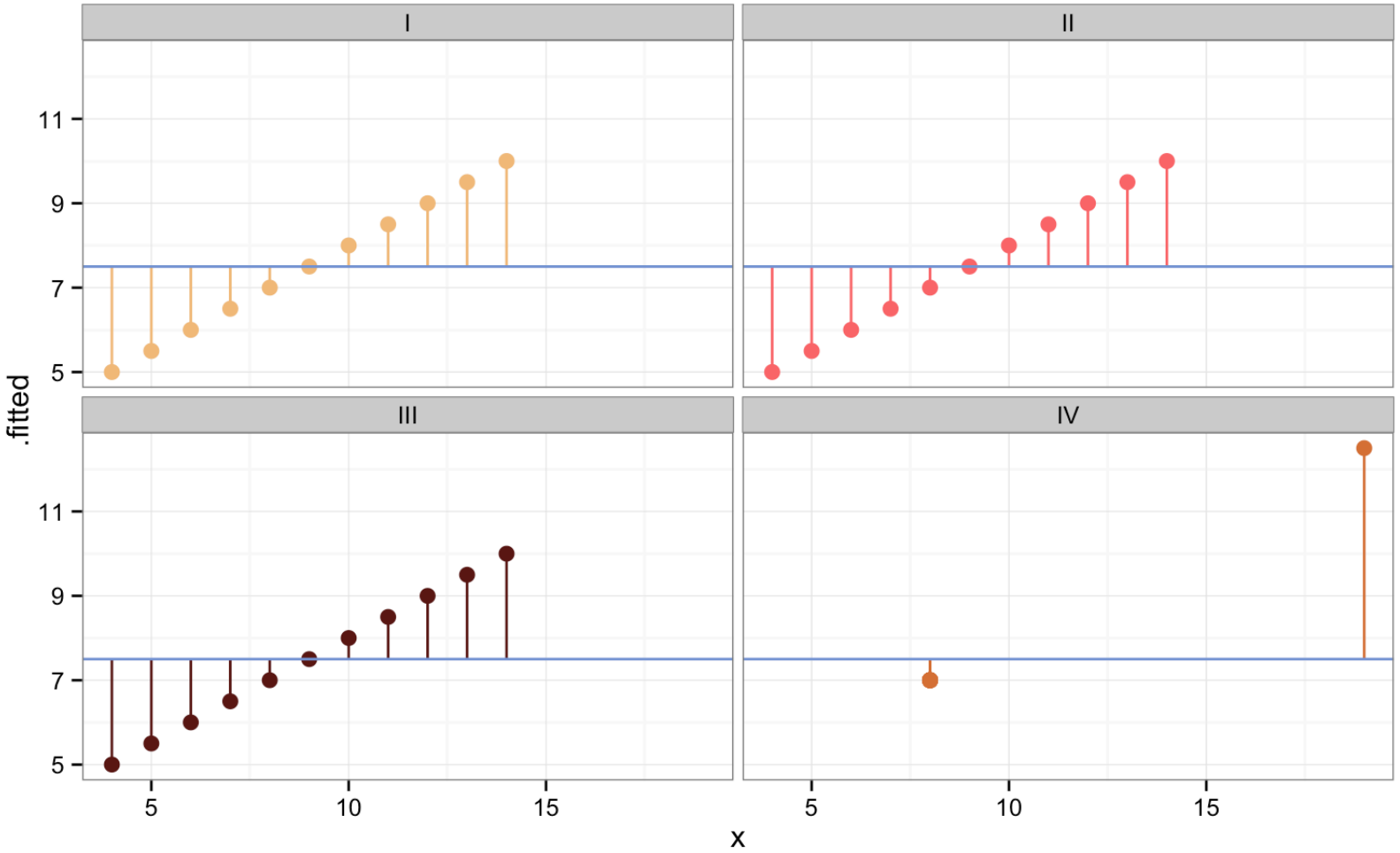
What you want to see: no pattern at all




ONE MORE COOKIE

Model sums of squares

Observed?
Model?



Model sums of squares

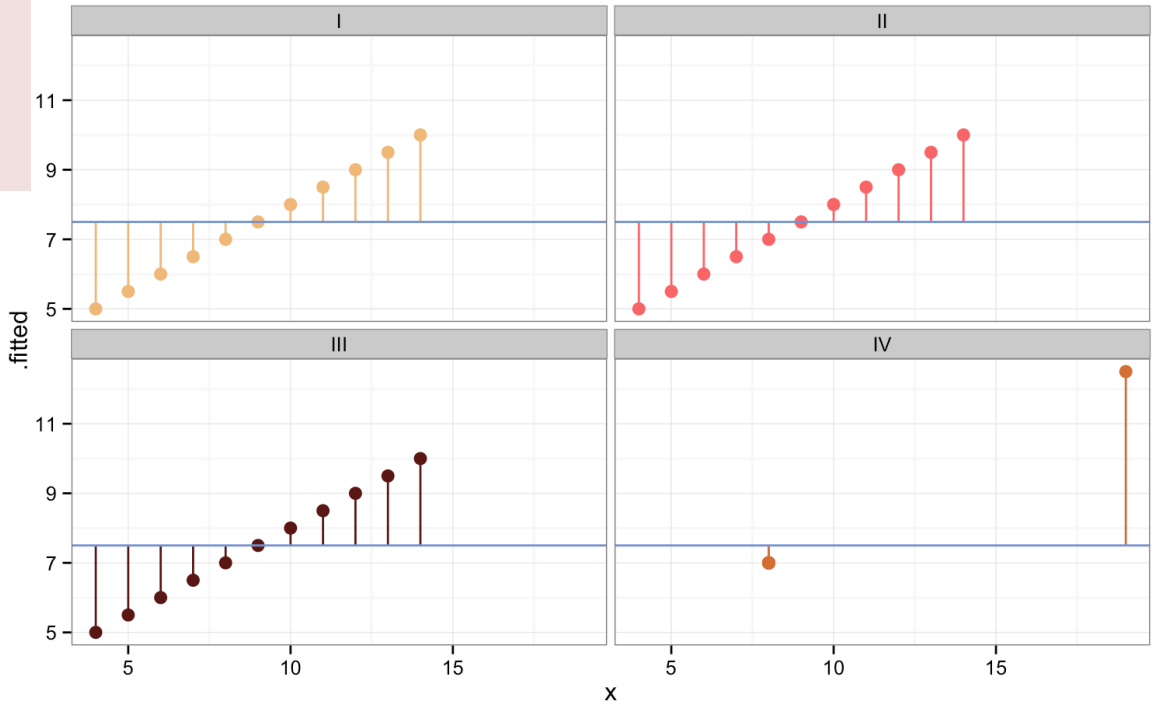
observed – model

$$ModelSS = \sum (\hat{y}_i - \overline{y})^2$$

More like:
Linear regression model – mean model

Evaluating fit of regression model vs mean model

```
> obs_sum %>%
+   group_by(set) %>%
+   summarise(tot_ss = sum((y - mean(y))^2),
+             res_ss = sum((y - .fitted)^2),
+             mod_ss = sum((.fitted - mean(y))^2))
Source: local data frame [4 x 4]

  set   tot_ss   res_ss   mod_ss
1   I 41.27269 13.76269 27.51000
2  II 41.27629 13.77629 27.50000
3 III 41.22620 13.75619 27.47001
4  IV 41.23249 13.74249 27.49000
```



Sums of squares

Total sums of squares	**Model sums of squares**	**Residual sums of squares**
$\sum (y_i - \overline{y})^2$	$\sum (\hat{y}_i - \overline{y})^2$	$\sum (y_i - \hat{y}_i)^2$

total variation = "explained" variation + residual variation

Sums of squares

Total sums of squares	Model sums of squares	Residual sums of squares
$\sum (y_i - \overline{y})^2$	$\sum (\hat{y}_i - \overline{y})^2$	$\sum (y_i - \hat{y}_i)^2$

total variation	=	"explained" variation	+	residual variation
41.2	=	27.5	+	13.7

```
  set   tot_ss   res_ss   mod_ss
1   I 41.27269 13.76269 27.51000
2  II 41.27629 13.77629 27.50000
3 III 41.22620 13.75619 27.47001
4  IV 41.23249 13.74249 27.49000
```

Squared multiple correlation, R^2

Total sums of squares	**Model sums of squares**	**Residual sums of squares**
$\sum(y_i - \overline{y})^2$	$\sum(\hat{y}_i - \overline{y})^2$	$\sum(y_i - \hat{y}_i)^2$

total variation = "explained" variation + residual variation

$$R^2 = \frac{\text{explained variation}}{\text{total variation}}$$

$$= \frac{\sum(\hat{y}_i - \overline{y})^2}{\sum(y_i - \overline{y})^2}$$

$$= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y})^2}$$

$$\frac{27.5}{41.2} = .667$$

```
> model_one <- lm(y ~ x, data = anscombe_one) # one lm model
> summary(model_one)

Call:
lm(formula = y ~ x, data = anscombe_one)

Residuals:
     Min       1Q   Median       3Q      Max
-1.92127 -0.45577 -0.04136  0.70941  1.83882

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0001     1.1247   2.667  0.02573 *
x             0.5001     0.1179   4.241  0.00217 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.237 on 9 degrees of freedom
Multiple R-squared:  0.6665,        Adjusted R-squared:  0.6295
F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217

> glance(model_one) # library(broom)
  r.squared adj.r.squared    sigma statistic     p.value df    logLik      AIC      BIC deviance df.residual
1 0.6665425     0.6294916 1.236603  17.98994 0.002169629  2 -16.84069 39.68137 40.87506 13.76269           9
```

```
> model_sum <- models %>% glance(mod) # library(broom)
> model_sum # many lm models
Source: local data frame [4 x 12]
Groups: set

  set r.squared adj.r.squared    sigma statistic     p.value df    logLik      AIC      BIC deviance df.residual
1   I 0.6665425     0.6294916 1.236603  17.98994 0.002169629  2 -16.84069 39.68137 40.87506 13.76269           9
2  II 0.6662420     0.6291578 1.237214  17.96565 0.002178816  2 -16.84612 39.69224 40.88593 13.77629           9
3 III 0.6663240     0.6292489 1.236311  17.97228 0.002176305  2 -16.83809 39.67618 40.86986 13.75619           9
4  IV 0.6667073     0.6296747 1.235695  18.00329 0.002164602  2 -16.83261 39.66522 40.85890 13.74249           9
```

Anscombe's quartet: sum up

- The linear least-squares regression is a good summary of the relationship between x and y only for the first dataset.
- In the second dataset, the relationship is nonlinear.
- In the third dataset, there is an outlier.
- In the fourth dataset, the least-squares line "chases" the influential observation.
- None of these problems is clear from the fitted regression equation and correlation
- None (but the last) is clear from looking at the numerical data
- All are clear from looking at the scatterplots + residual plots

Errors

- There is another parameter in the regression model, but it does not show up directly in the formula. That is:
  - We know the errors we can measure- these are the residuals. The are "known knowns" – we can measure these.
  - But there is another type of error – the "known unknowns". We know they are there, but we cannot measure them.

"…there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know."

Errors != Residuals

Errors

- **Errors** are the vertical distances between observations and the unknown Conditional Expectation Function.
- Therefore, they are **unknown**.

Residuals

- **Residuals** are the vertical distances between observations and the estimated regression function.
- Therefore, they are **known**.

Notation

Errors

- **Errors** represent the difference between the outcome and the true mean.

$$y = X\beta + \varepsilon$$

$$\varepsilon = y - X\beta$$

Residuals

- **Residuals** represent the difference between the outcome and the estimated mean.

$$y = X\hat{\beta} + \hat{\varepsilon}$$

$$\hat{\varepsilon} = y - X\hat{\beta}$$

Canadian occupational prestige dataset

```
library(car)
data(Prestige)
```

Description

- The Prestige data frame has 102 rows and 6 columns. The observations are occupations.

Source

- Canada (1971) *Census of Canada*. Vol. 3, Part 6. Statistics Canada [pp. 19-1–19-21].
- Personal communication from B. Blishen, W. Carroll, and C. Moore, Departments of Sociology, York University and University of Victoria.

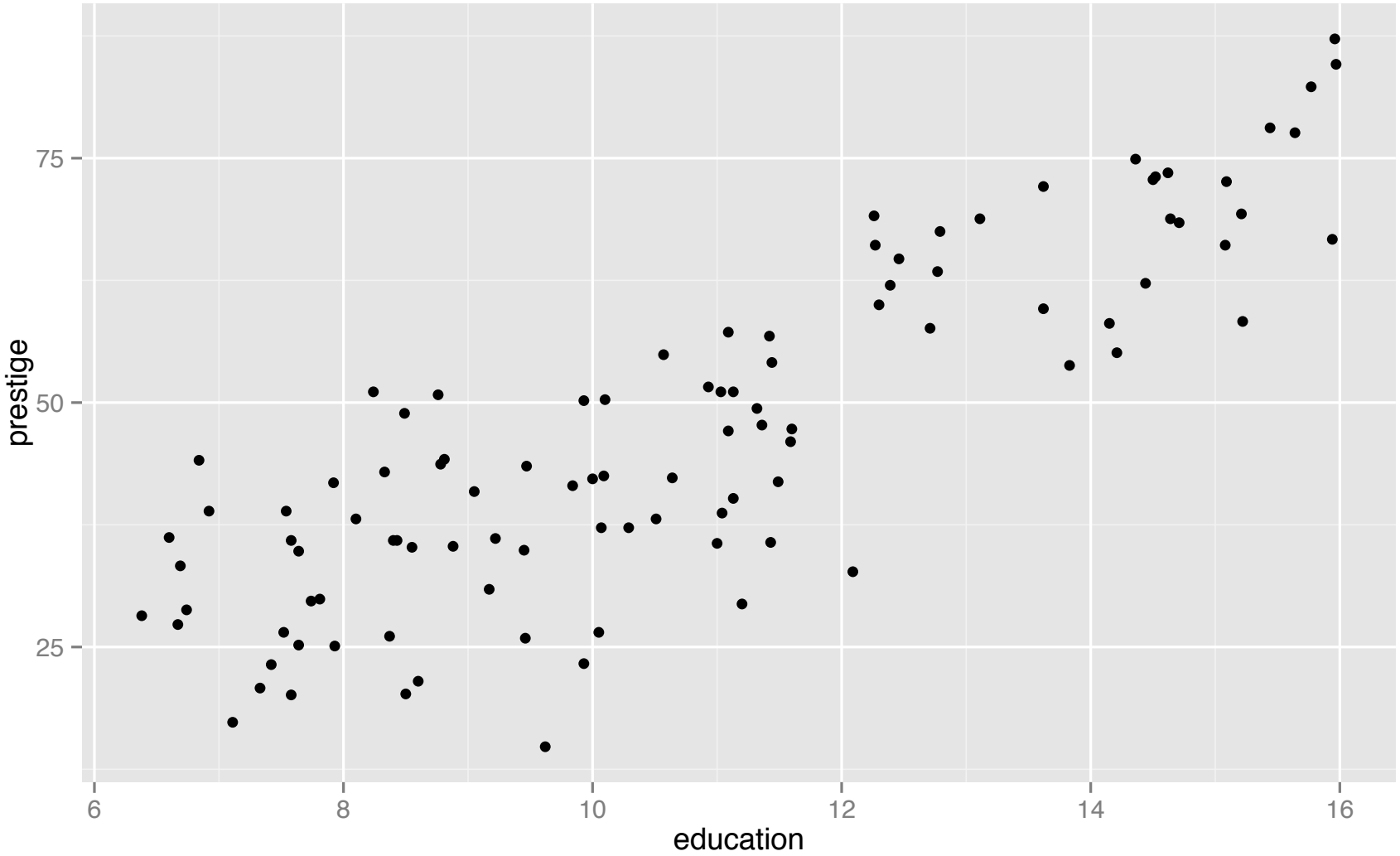
Canadian occupational prestige dataset

`education`

- Average education of occupational incumbents, years, in 1971.

`prestige`

- Pineo-Porter prestige score for occupation, from a social survey conducted in the mid-1960s.


prestige
75
50
25
6
8
10
12
14
16
education

Regression model parameters

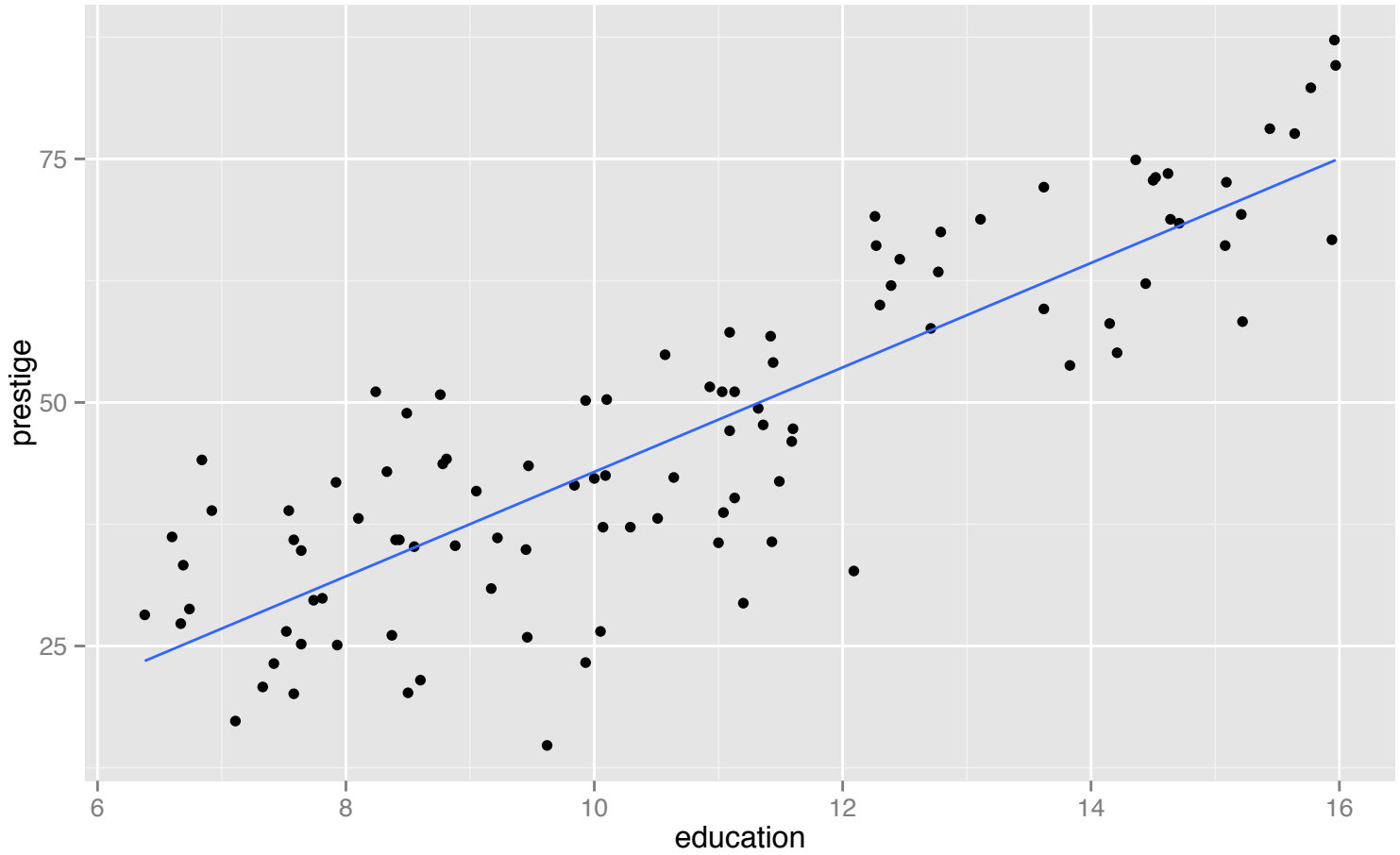
```
> Prestige %>%
+   summarise(r_xy = cor(education, prestige),
+             sd_x = sd(education),
+             sd_y = sd(prestige),
+             mean_x = mean(education),
+             mean_y = mean(prestige))
      r_xy     sd_x     sd_y   mean_x   mean_y
1 0.8501769 2.728444 17.20449 10.73804 46.83333
```

$$\beta_1 = r_{xy}\frac{s_y}{s_x} = .85018 \times \frac{17.2045}{2.7284} = 5.3610$$

$$\beta_0 = \overline{y} - \beta_1\overline{x} = 46.8333 - (.85018 \times 10.7380) = -10.7331$$

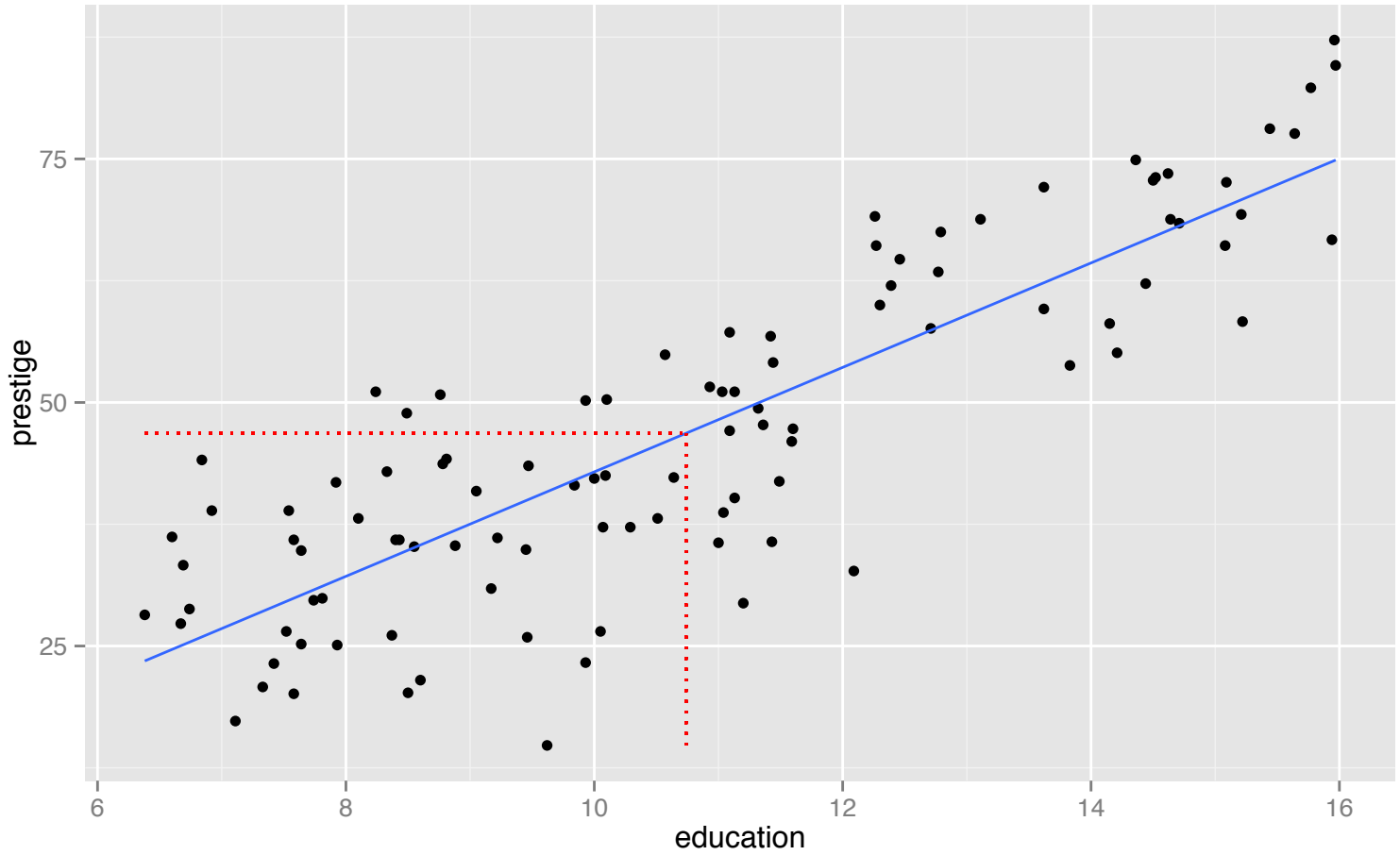
The fitted regression line

$$\hat{y} = -10.733 + 5.361x$$



The fitted regression line

$$\hat{y} = -10.733 + 5.361x$$



$$\hat{y} = -10.733 + 5.361x$$

- For a person with zero years of education, we would predict a prestige score of -10.733
  - Extrapolating past observed xs is not good
- Slope is 5.361: Each additional year of education is accompanied on average by an increase of a bit more than 5 prestige points.