SciencesPo
DEPARTMENT OF ECONOMICS

ScPoEconometrics

Session 3

Florian Oswald
SciencesPo Paris
2019-09-23

Data On Cars

- Suppose we had data on car `speed` and stopping `dist` (ance):

```
head(cars)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

- How are `speed` and `dist` related?
- What is a good summary of their relationship?

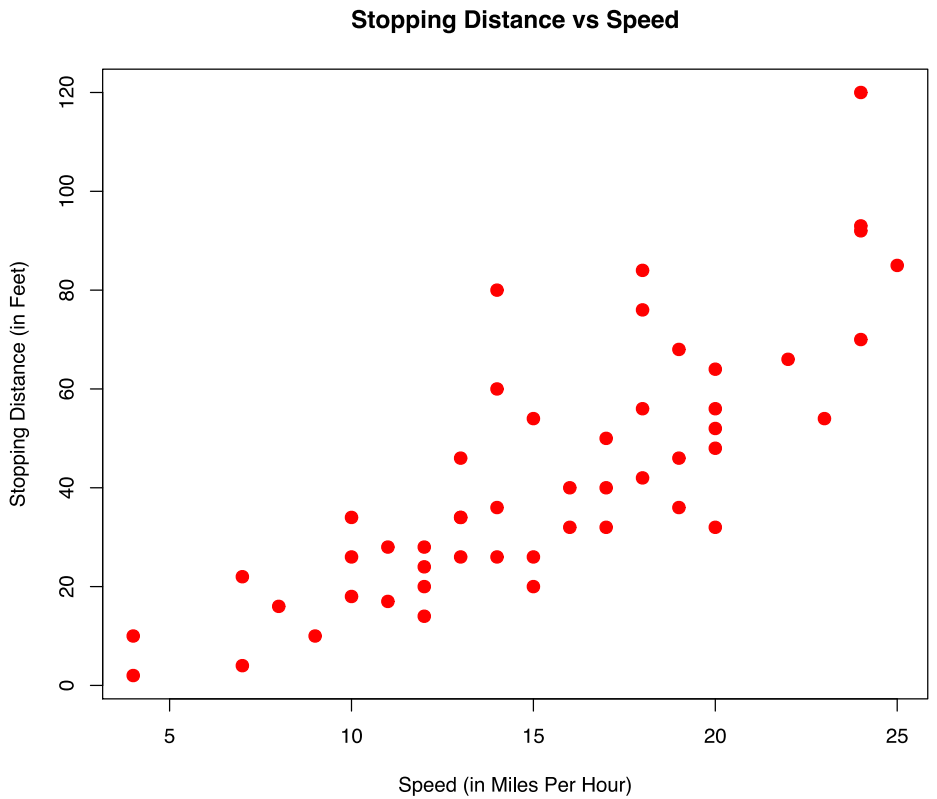
Data On Cars

- Suppose we had data on car `speed` and stopping `dist` (ance):

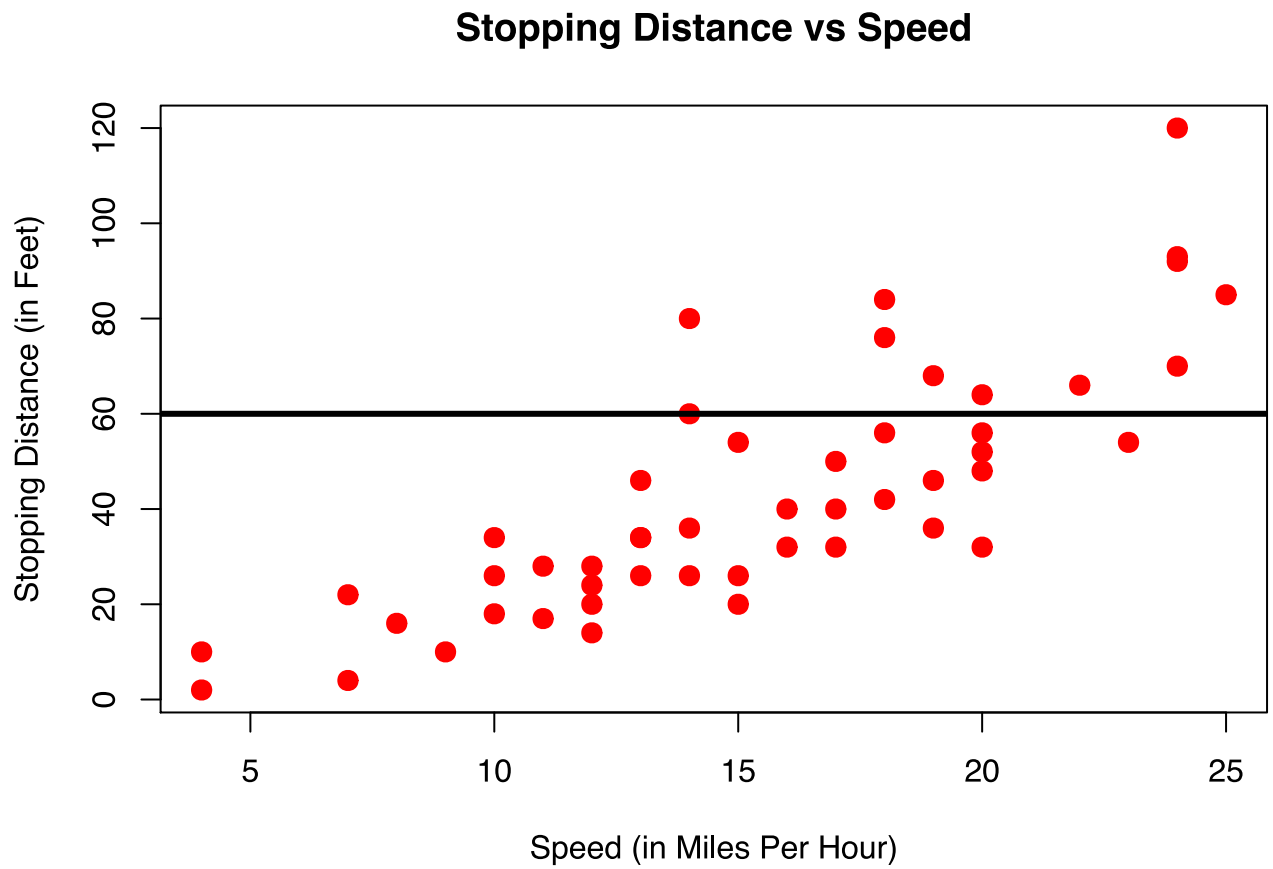
```
head(cars)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

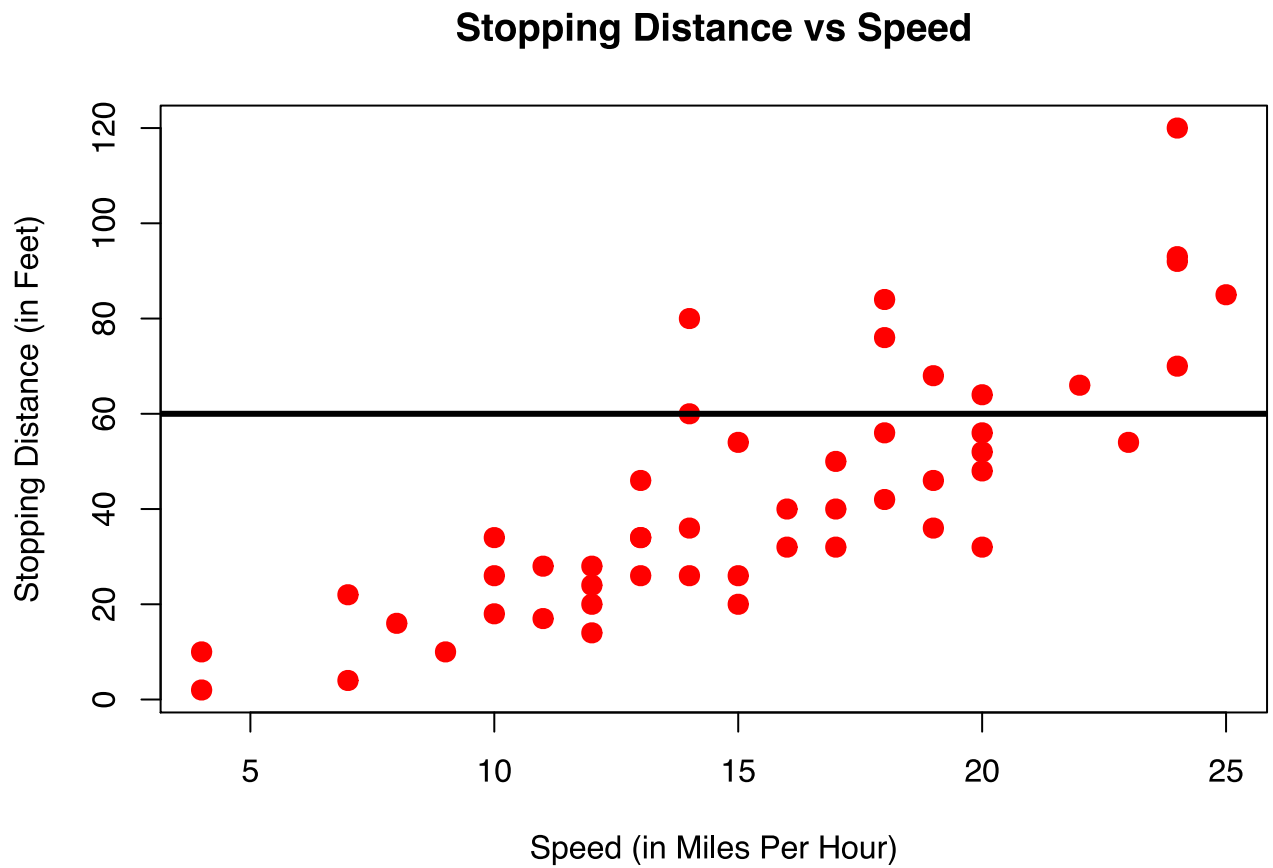
- How are `speed` and `dist` related?
- What is a good summary of their relationship?



A Line throught the Scatterplot of Cars

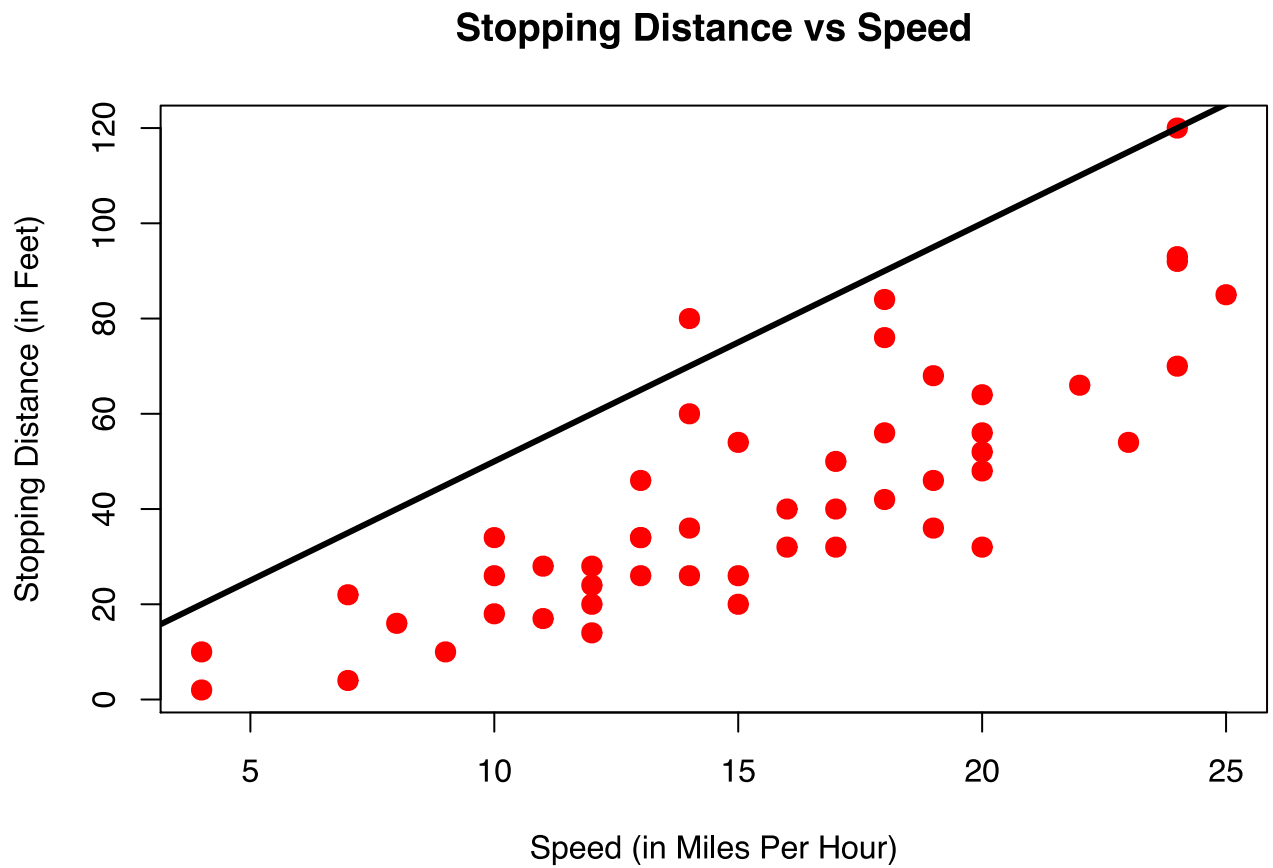


A Line throught the Scatterplot of Cars

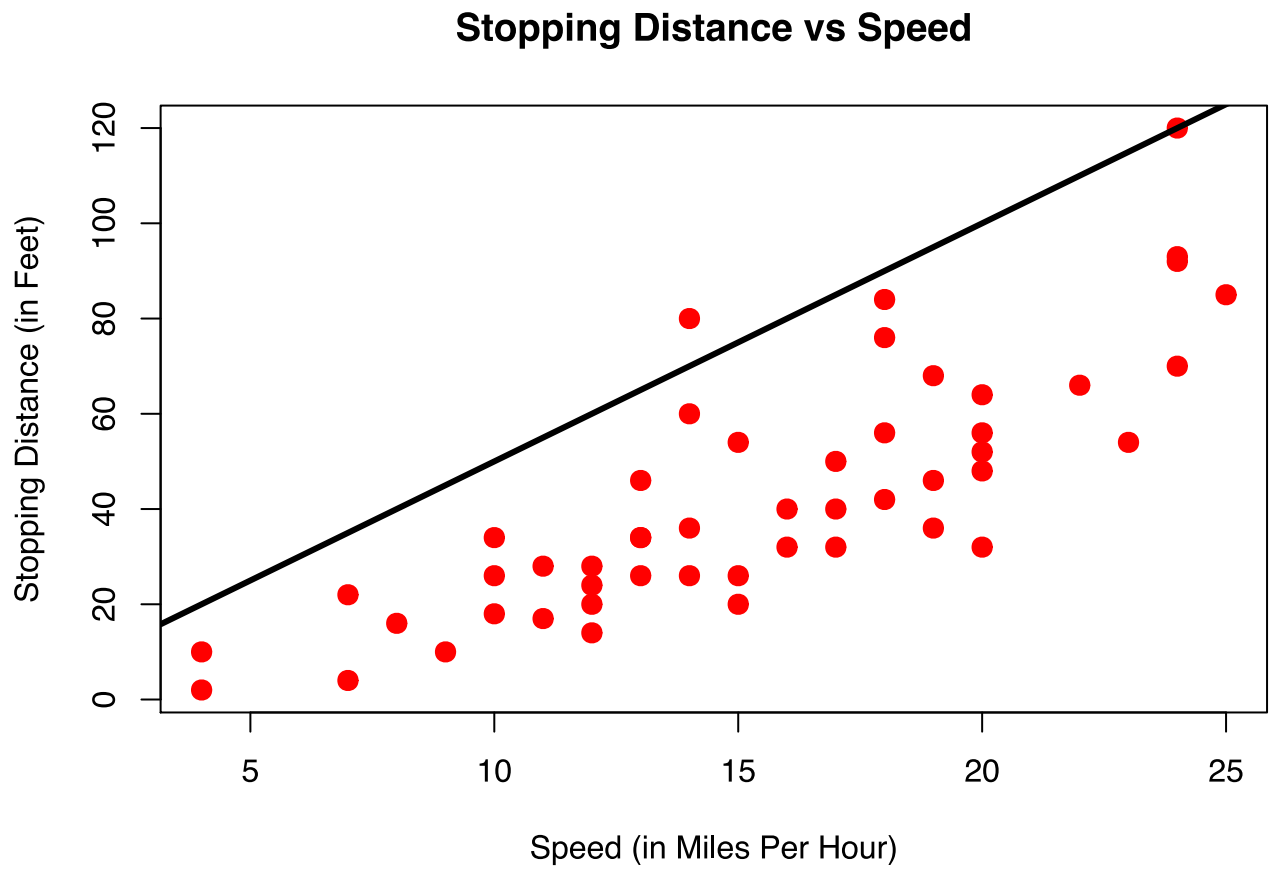


- A *line*! Great. But **which** line? This one?
- That's a *flat* line. But `dist` is increasing.
- 😩

A Line throught the Scatterplot of Cars



A Line throught the Scatterplot of Cars



- **That** one?
- Slightly better. Has a *slope* and an *intercept*.
- 😐

Writing Down A *Line*

- We observe (y_i, x_i) in the data.
- This describes a line with intercept b_0 and slope b_1:

$$\hat{y}_i = b_0 + b_1 x_i$$

- We call $\hat{y}_i$ the *prediction* for y_i.
- Most of the times, $\hat{y}_i \neq y_i$, i.e. we make an *error*.

Writing Down A *Line*

- We observe (y_i, x_i) in the data.
- This describes a line with intercept b_0 and slope b_1:

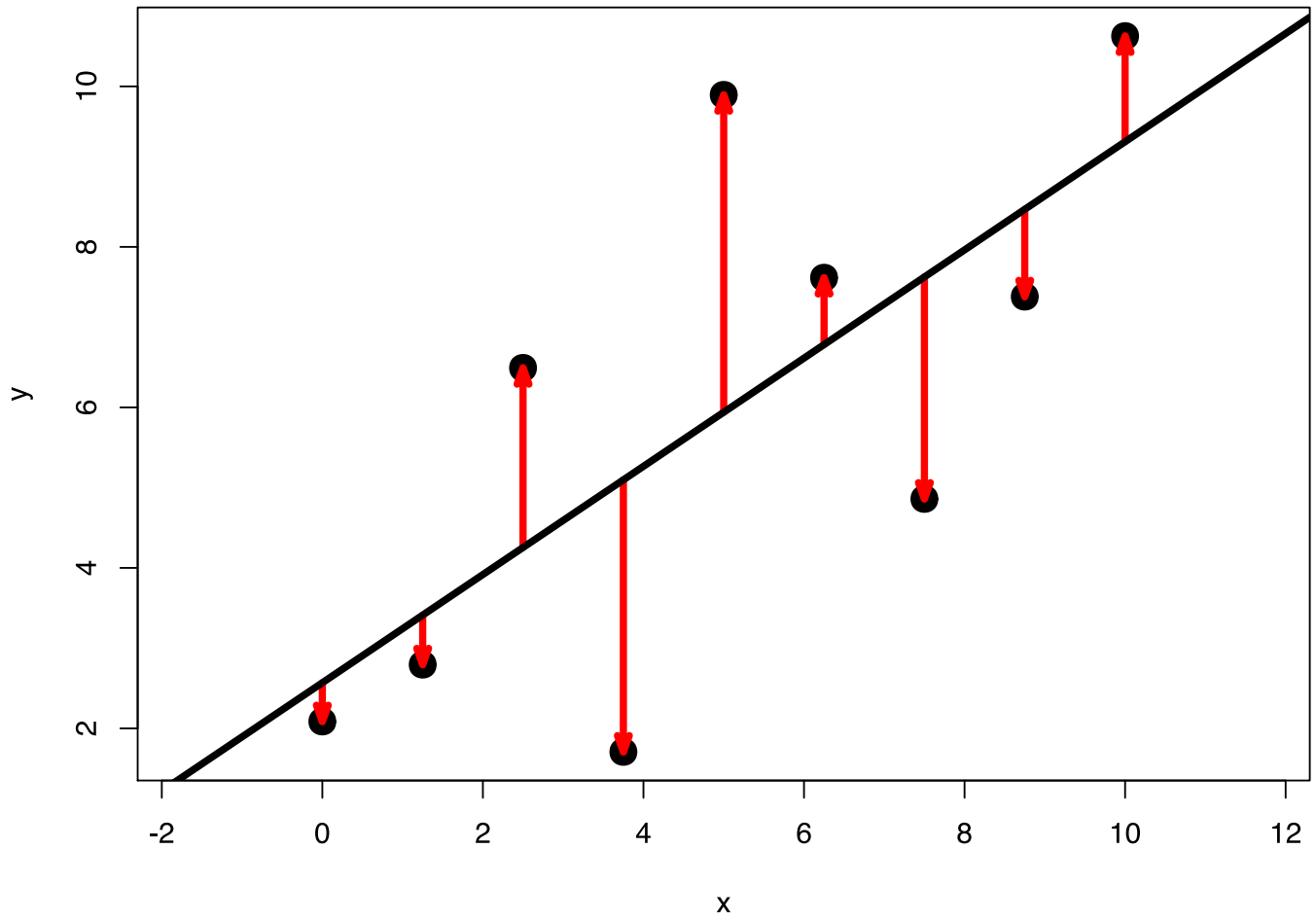
$$\hat{y}_i = b_0 + b_1 x_i$$

- We call $\hat{y}_i$ the *prediction* for y_i.
- Most of the times, $\hat{y}_i \neq y_i$, i.e. we make an *error*.

- At point x_i we make error e_i.
- Our aim will be to keep the error *as small as possible*, while at the same time giving a reasonable description of the data.
- (We could be more generally trying to fit a *curve* rather than a *line*, by the way.)
- The *actual data* (y_i, x_i) can thus be written like *prediction* + *error*:

$$y_i = b_0 + b_1 x_i + e_i$$

Making Errors



- Red Arrows are *errors* or *residuals* for each prediction.
- Often denoted u or e.
- Note that we have both $e > 0$ and $e < 0$!

App Time!

- Let's try to find the best line using only the *absolute value* of errors!

```
library(ScPoEconometrics) # load our library
launchApp('reg_simple_arrows')
aboutApp('reg_simple_arrows') # explainer about app
```

Writing Down *The Best* Line

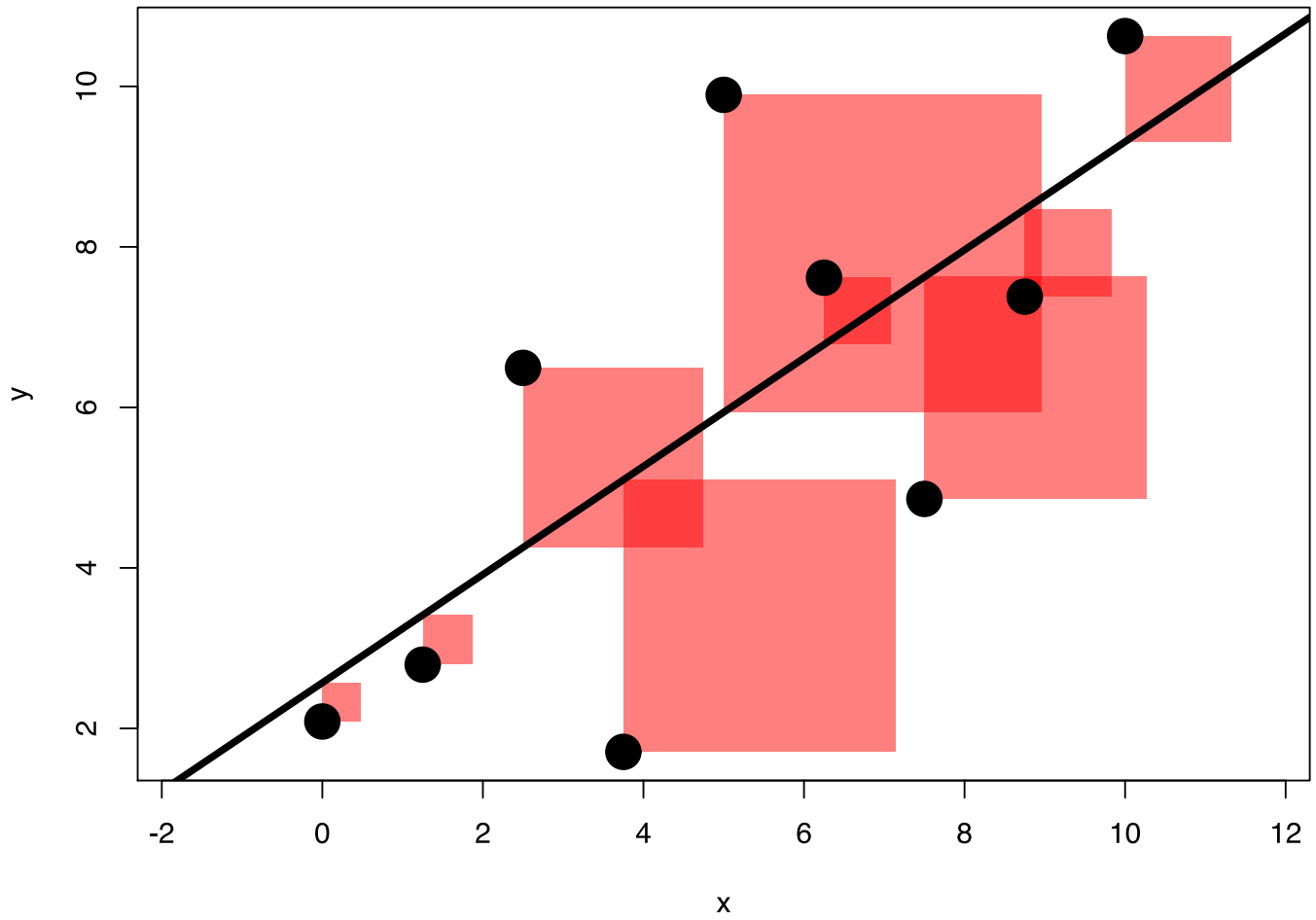
- choose (b_0, b_1) s.t. the sum $e_1^2 + \cdots + e_N^2$ is **as small as possible**
- $e_1^2 + \cdots + e_N^2$ is the *sum of squared residuals*, or SSR.
- Wait a moment... Why *squared* residuals?!

Writing Down *The Best* Line

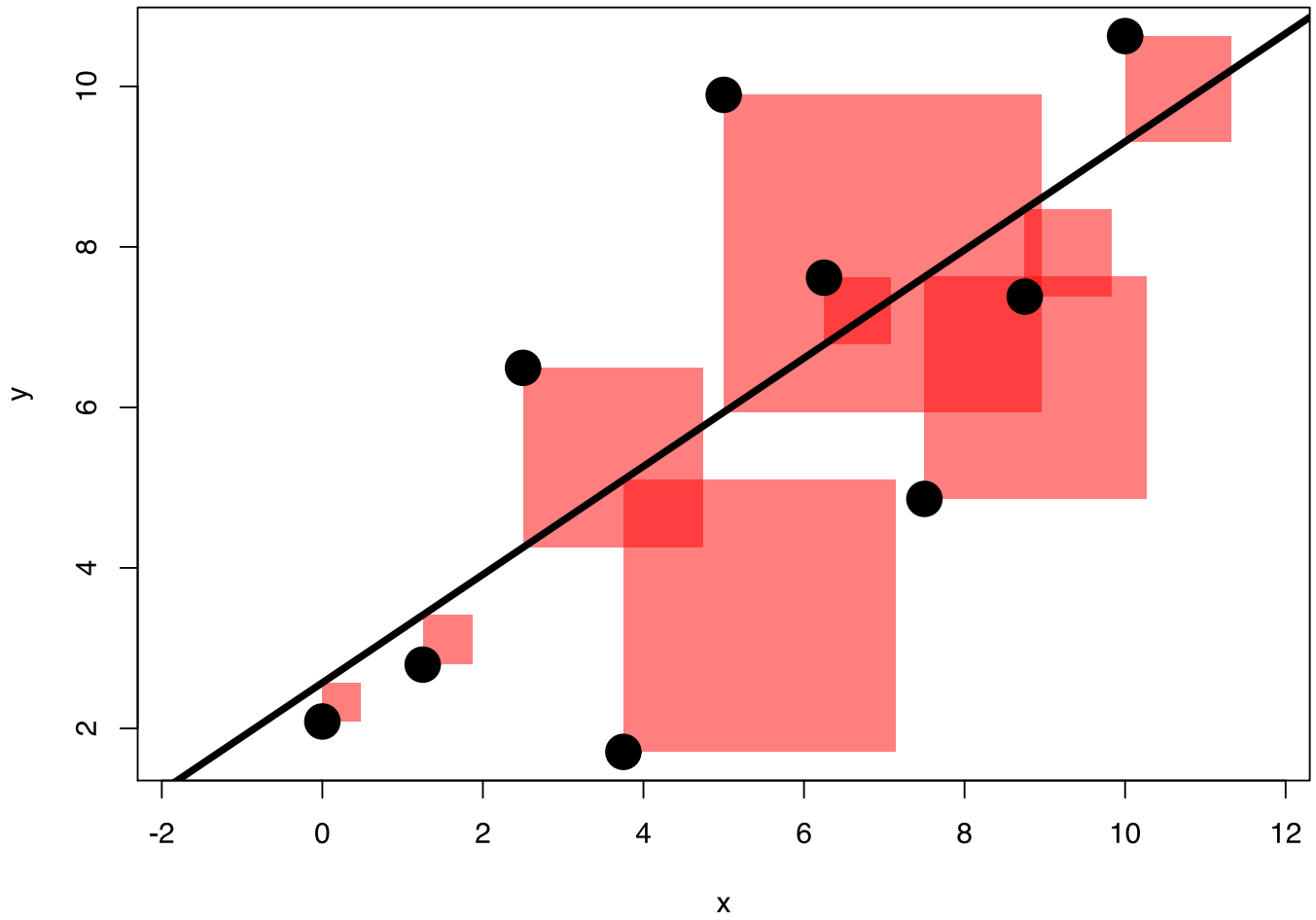
- choose (b_0, b_1) s.t. the sum $e_1^2 + \cdots + e_N^2$ is **as small as possible**
- $e_1^2 + \cdots + e_N^2$ is the *sum of squared residuals*, or SSR.
- Wait a moment... Why *squared* residuals?!

- In previous plot, errors of different sign $(+/-)$ cancel out!
- This makes it hard to find a good line.
- Squaring each e_i solves that issue as $e_i^2 \geq 0, \forall i$.

Best Line and Squared Errors



Best Line and Squared Errors



- **That's** the one!
- Perfect! Minimizes the sum of squares.
- 😌

App Time!

```
launchApp('reg_simple')
aboutApp('reg_simple')
```

Ordinary Least Squares (OLS)

- OLS estimates the best line for us.
- In our single regressor case, there is a simple formula for the slope:

$$b_1 = \frac{cov(x, y)}{var(x)}$$

- and for the intercept

$$b_0 = \bar{y} - b_1 \bar{x}$$

Ordinary Least Squares (OLS)

- OLS estimates the best line for us.
- In our single regressor case, there is a simple formula for the slope:

$$b_1 = \frac{cov(x, y)}{var(x)}$$

- and for the intercept

$$b_0 = \bar{y} - b_1 \bar{x}$$

- 🚨
- You **must** know those formulae!

App Time!

How does OLS actually perform the minimization problem?

```
launchApp('SSR_cone')
aboutApp('SSR_cone')  # after
```

App Time!

Let's do some more OLS!

```
launchApp('reg_full')
aboutApp('reg_full')  # after
```

OLS without any Regressor

- Our line is flat at level b_0:

$$y = b_0$$

- Our optimization problem is now

$$b_0 = \arg\min_{\text{int}} \sum_{i=1}^{N} [y_i - \text{int}]^2,$$

- With solution

$$b_0 = \frac{1}{N} \sum_{i=1}^{N} y_i = \bar{y}.$$

OLS without any Regressor

- Our line is flat at level b_0:

$$y = b_0$$

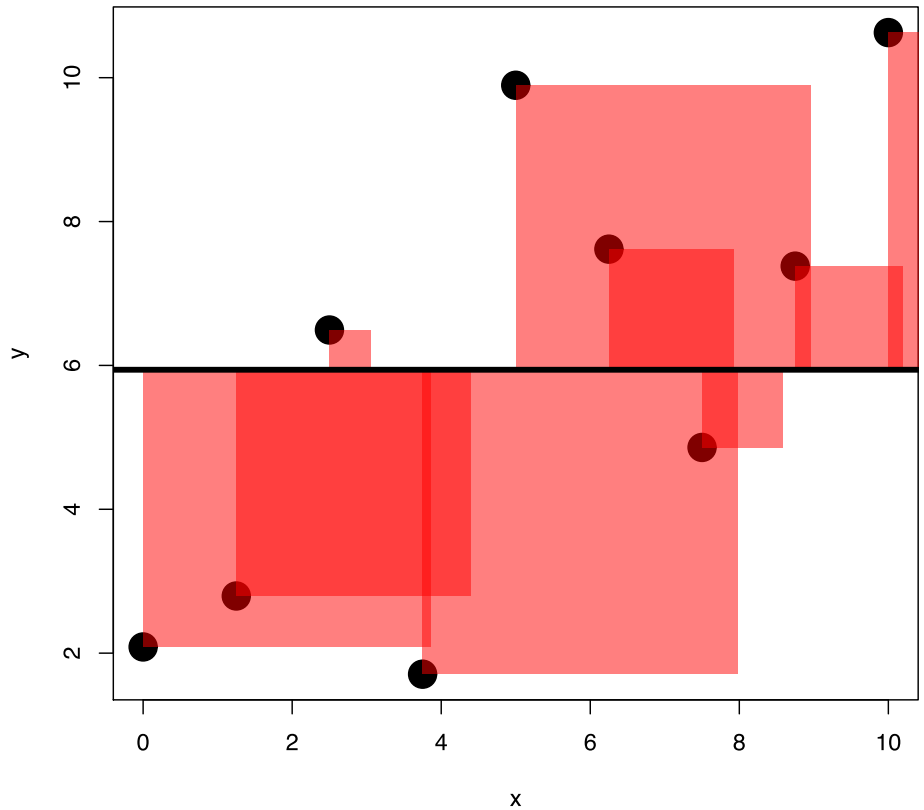
- Our optimization problem is now

$$b_0 = \arg\min_{\text{int}} \sum_{i=1}^{N} [y_i - \text{int}]^2,$$

- With solution

$$b_0 = \frac{1}{N} \sum_{i=1}^{N} y_i = \bar{y}.$$

- In other words: Estimates the **mean** of y!



Other OLS Restrictions

- There are other restrictions we can impose.
- They are described in the book. Optional 🤓.
- There is an app for each of them:

type	App
No Intercept	`launchApp('reg_constrained')`
Centered Regression	`launchApp('demeaned_reg')`
Standardized Regression	`launchApp('reg_standardized')`

Predictions and Residuals

1. The error is $e_i = y_i - \hat{y}_i$
2. The average of $\hat{y}_i$ is equal to $\bar{y}$.

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}\hat{y}_i &= \frac{1}{N}\sum_{i=1}^{N} b_0 + b_1 x_i \\ &= b_0 + b_1\bar{x} = \bar{y}\end{aligned}$$

Predictions and Residuals

1. The error is $e_i = y_i - \hat{y}_i$
2. The average of $\hat{y}_i$ is equal to $\bar{y}$.

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}\hat{y}_i &= \frac{1}{N}\sum_{i=1}^{N} b_0 + b_1 x_i \\ &= b_0 + b_1\bar{x} = \bar{y}\end{aligned}$$

3. Then,

$$\frac{1}{N}\sum_{i=1}^{N} e_i = \bar{y} - \frac{1}{N}\sum_{i=1}^{N}\hat{y}_i = 0$$

   i.e. the average of errors is zero.

Properties of Residuals

1. The average of $\hat{y}_i$ is the same as the mean of y.
2. The average of the errors should be zero.
3. Prediction and errors should be *uncorrelated* (i.e. orthogonal).

Let's look at the data behind our *arrows* plot above:

Properties of Residuals

1. The average of $\hat{y}_i$ is the same as the mean of y.
2. The average of the errors should be zero.
3. Prediction and errors should be *uncorrelated* (i.e. orthogonal).

Let's look at the data behind our *arrows* plot above:

	x	y	y_hat	error
	0.00	2.09	2.57	-0.48
	1.25	2.79	3.41	-0.62
	2.50	6.49	4.25	2.24
	3.75	1.71	5.10	-3.39
	5.00	9.89	5.94	3.95
	6.25	7.62	6.78	0.83
	7.50	4.86	7.63	-2.77
	8.75	7.38	8.47	-1.09
	10.00	10.63	9.31	1.32
Means	5.00	5.94	5.94	-0.00

Properties of Residuals

1. The average of $\hat{y}_i$ is the same as the mean of y.

2. The average of the errors should be zero.

3. Prediction and errors should be *uncorrelated* (i.e. orthogonal).

Properties of Residuals

1. The average of $\hat{y}_i$ is the same as the mean of y.
2. The average of the errors should be zero.
3. Prediction and errors should be *uncorrelated* (i.e. orthogonal).

```
# 1.
all.equal(mean(sd$y_hat), mean(sd$y))
```

```
## [1] TRUE
```

```
# 2.
all.equal(mean(sd$error), 0)
```

```
## [1] TRUE
```

```
# 3.
all.equal(cov(sd$error,sd$y_hat), 0)
```

```
## [1] TRUE
```

Linear Statistics

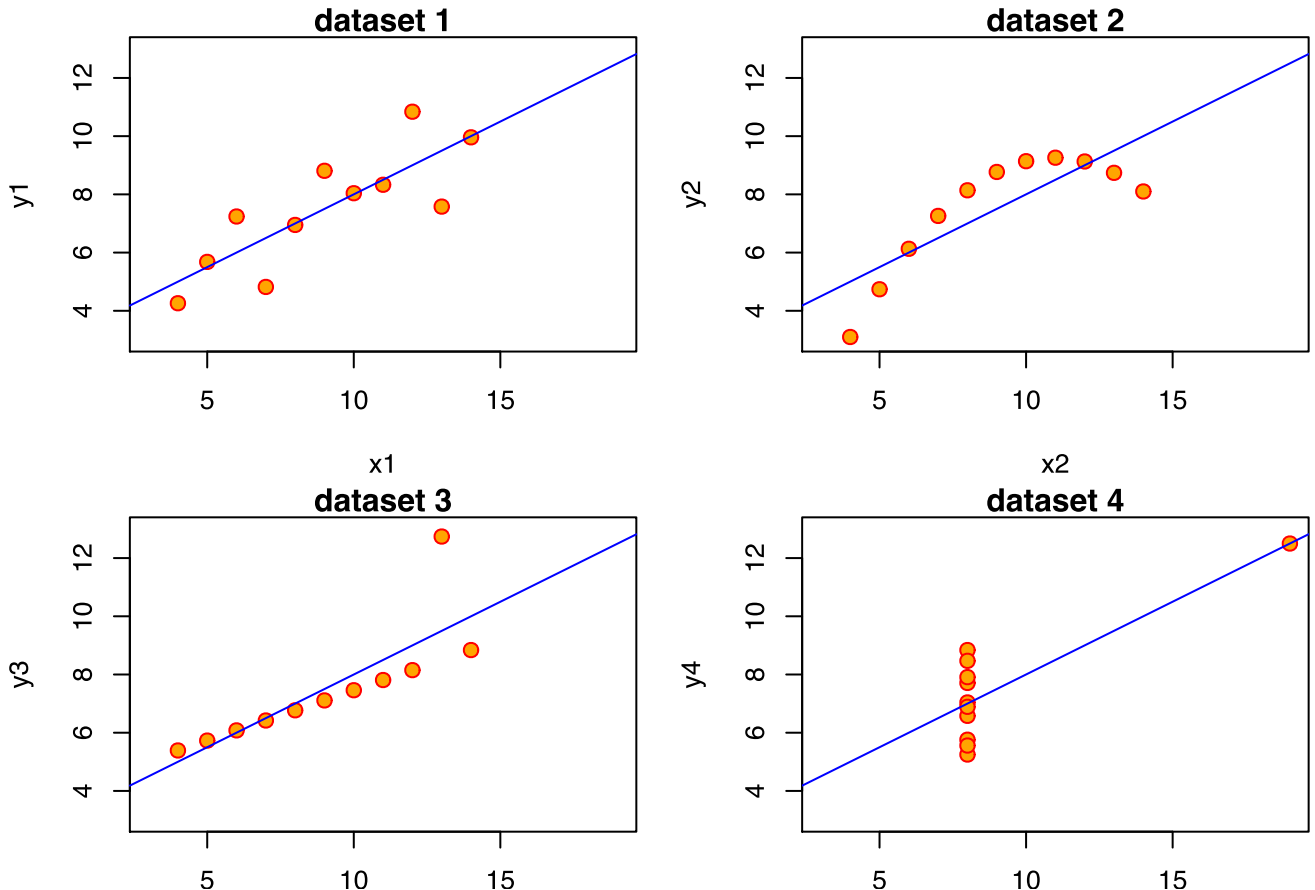
- It's important to keep in mind that Var, Cov, Corr and Regression measure **linear relationships** between two variables.
- Two datasets with *identical* correlations could look *vastly* different.
- They would have the same regression line.
- Same correlation coefficient.

Linear Statistics

- It's important to keep in mind that Var, Cov, Corr and Regression measure **linear relationships** between two variables.
- Two datasets with *identical* correlations could look *vastly* different.
- They would have the same regression line.
- Same correlation coefficient.
- Is that even possible?

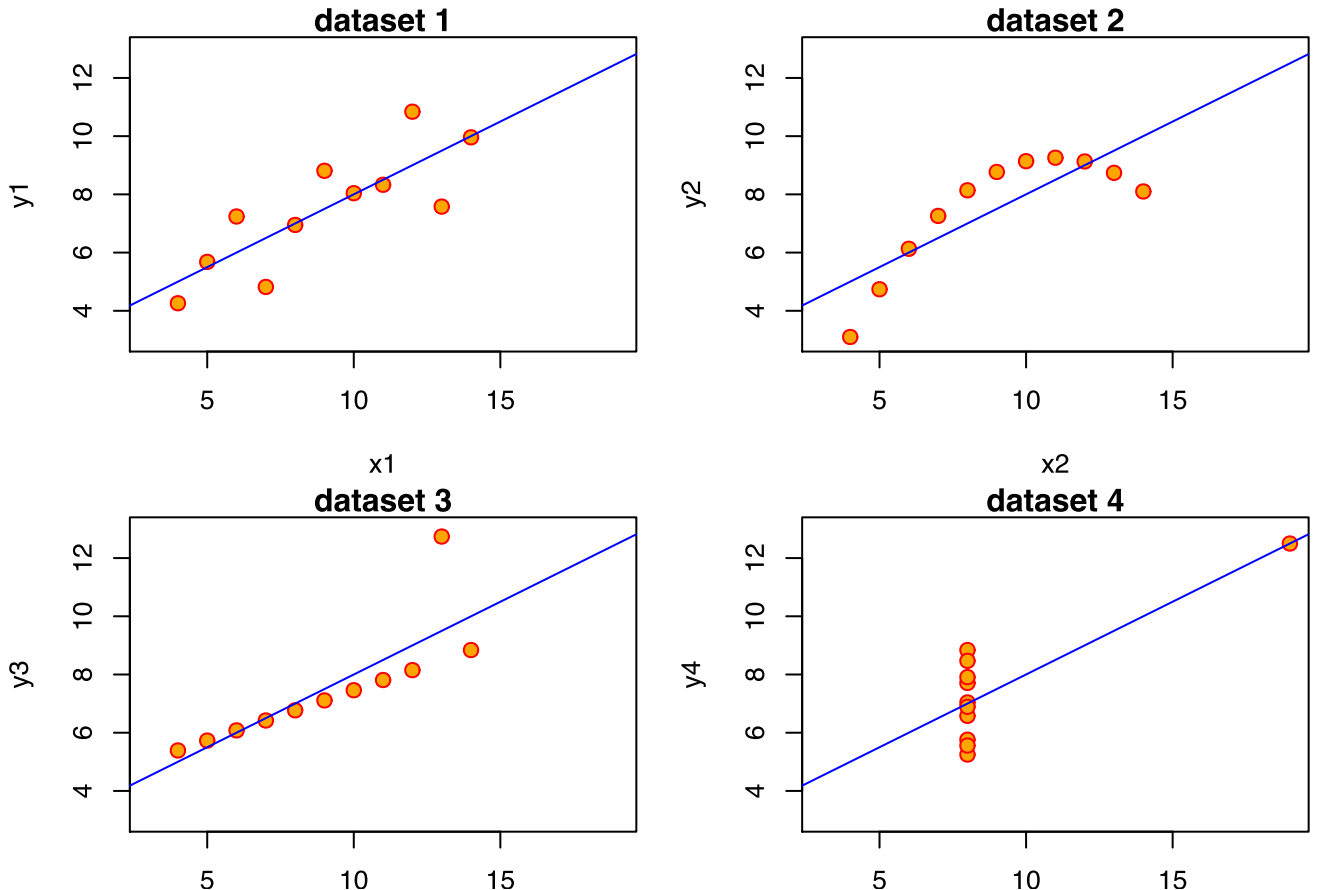
Linear Statistics: Anscombe

- Francis Anscombe (1973) comes up with 4 datasets with identical stats. But look!



Linear Statistics: Anscombe

- Francis Anscombe (1973) comes up with 4 datasets with identical stats. But look!



dataset	cov	var(y)	var(x)
1	5.50	4.13	11.00
2	5.50	4.13	11.00
3	5.50	4.12	11.00
4	5.50	4.12	11.00

Dinosaurs in your Data?

- So, be wary of only looking a linear summary stats.
- Also look at plots.

- Dinosaurs?

```
launchApp("datasaurus")
aboutApp("datasaurus")
```

Nonlinear Relationships in Data?

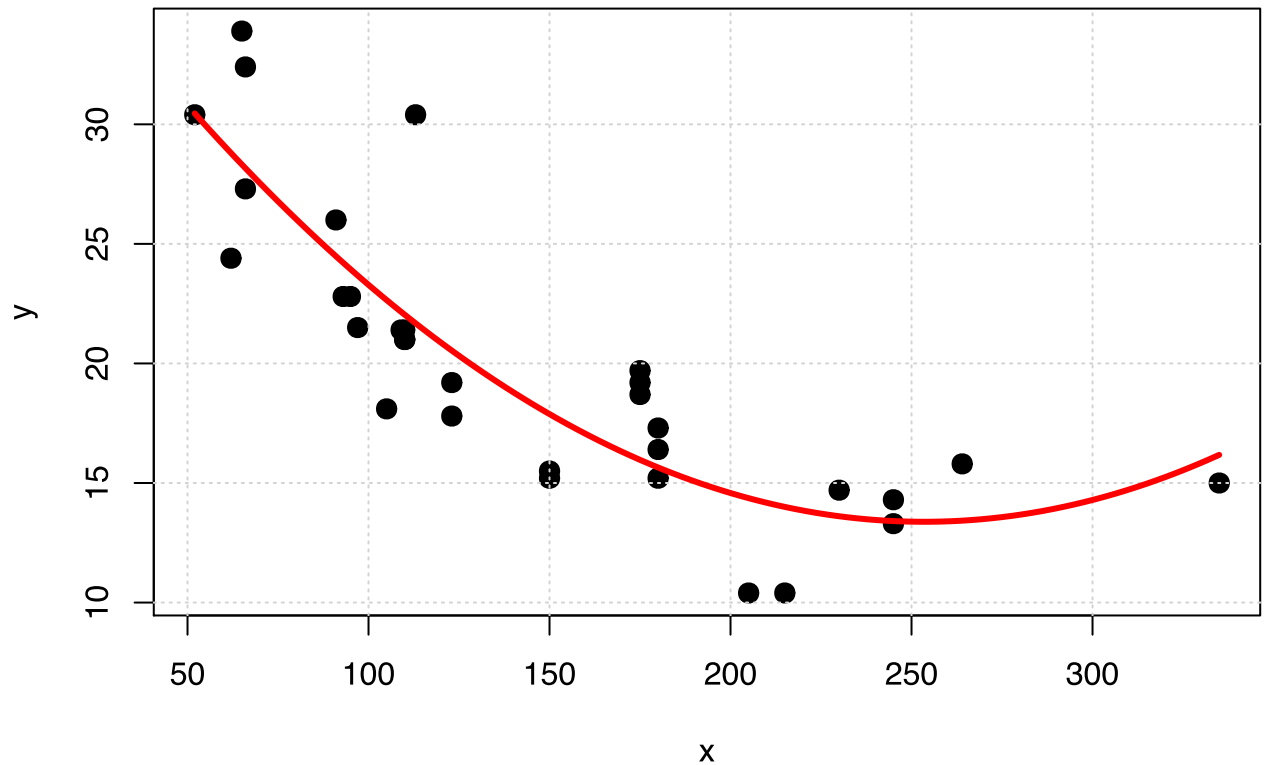
- We can accomodate non-linear relationships in regressions.
- We'd just add a higher order term like this:

$$y_i = b_0 + b_1 x_i + b_2 x_i^2 + e_i$$

- This is *multiple regression* (next chapter!)

Nonlinear Relationships in Data?

- For example, suppose we had this data and fit the above regression:



Analysis of Variance

- Remember that $y_i = \hat{y}_i + e_i$.
- We have the following decomposition:

$$\begin{aligned} Var(y) &= Var(\hat{y} + e) \\ &= Var(\hat{y}) + Var(e) + 2Cov(\hat{y}, e) \\ &= Var(\hat{y}) + Var(e) \end{aligned}$$

- Because: $Cov(\hat{y}, e) = 0$
- Total variation (SST) = Model explained (SSE) + Unexplained (SSR)

Assessing the Goodness of Fit

- The R^2 measures how good the model fits the data.
- $R^2 = 1$ is very good, $R^2 = 0$ is very poorly.

$$R^2 = \frac{\text{variance explained}}{\text{total variance}} = \frac{SSE}{SST} = 1 - \frac{SSR}{SST} \in [0, 1]$$

- Small R^2 doesn't mean it's a useless model!

An Example - California Test Scores

- Let's look at the worked example in the book!

Rescaling Regressions

- Suppose outcome y is *income in Euros*
- x be years of schooling

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

- Assume that $\beta_1 = 2000$, s.t. each additional year of schooling gives 2000 Euros more income.
- What is β_1 if we measure y in *thousands of euros* instead?

Rescaling Regressions

- The result depends on whether we rescale y, x or both.
- If we have rescale x by c and y by d, one can show that

$$\begin{aligned} b_1' &= \frac{Cov(dx, cy)}{Var(dx)} \\ &= \frac{d}{c} b \end{aligned}$$

- and the intercept

$$\begin{aligned} b_0' &= \mathbb{E}[d \cdot Y] - \frac{Cov(cX, dY)}{Var(cX)} \mathbb{E}[c \cdot X] \\ &= d b_0 \end{aligned}$$

App: Rescaling Regressors

```
library(ScPoEconometrics)
launchApp('Rescale')
```

```
library(ScPoEconometrics)
runTutorial('rescaling')
```

SciencesPo
DEPARTMENT OF ECONOMICS

END

	florian.oswald@sciencespo.fr
	Slides
	Book
	@ScPoEcon
	@ScPoEcon