Polynomials (mainly)

Bob O'Hara

February 20, 2019

Yesterday: Multiple Regression

We can now write a multiple regression model

$$y_i = \alpha + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i$$

We can fit it in R

```
lm(Dust ~ GapeSize + BodySize, data=Schey)
```

We know what a design matrix looks like

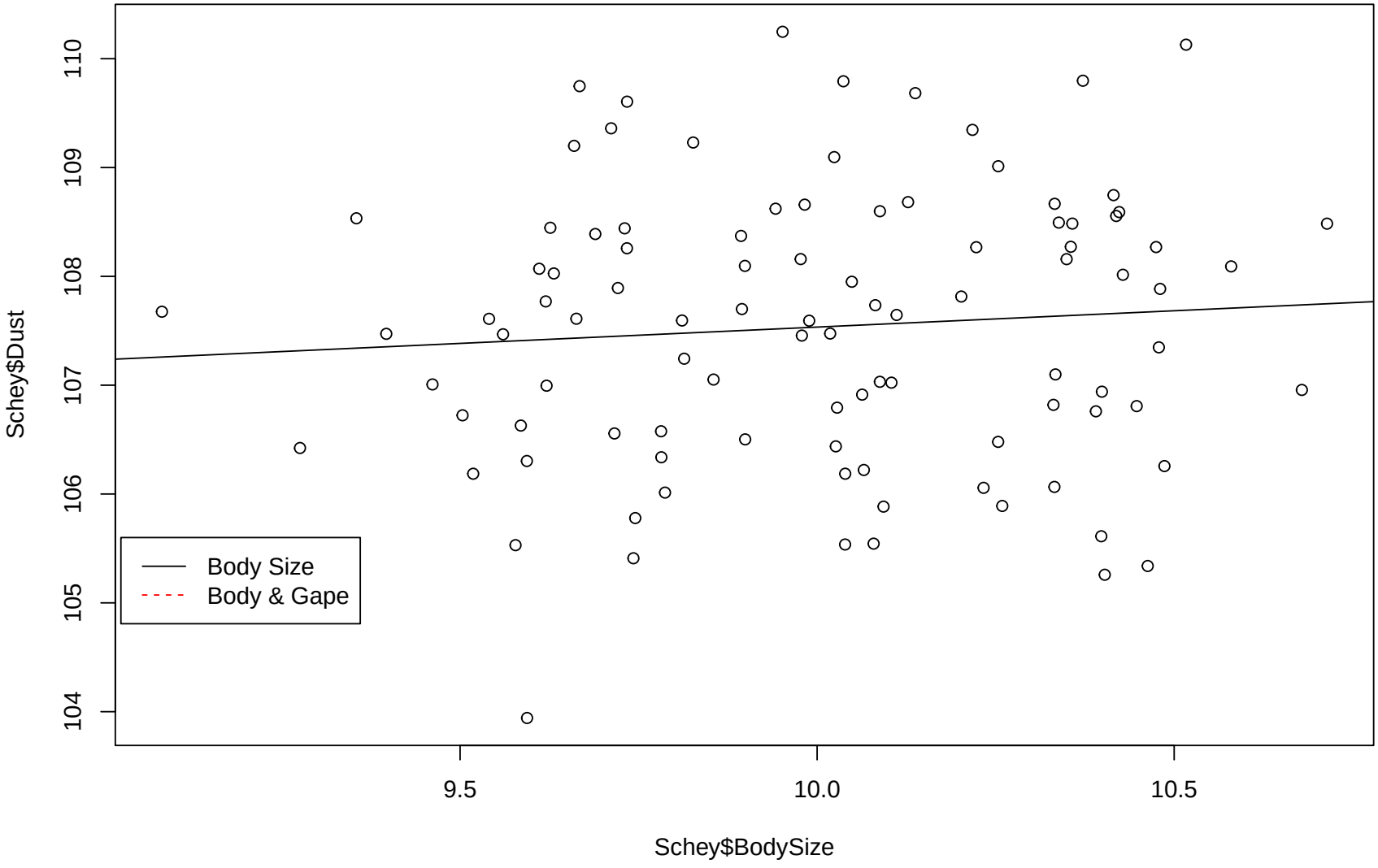
$$X = \begin{pmatrix} 1 & 2.3 & 3.0 \\ 1 & 4.9 & -5.3 \\ 1 & 1.6 & -0.7 \\ \vdots & \vdots & \vdots \\ 1 & 8.4 & 1.2 \end{pmatrix}$$

Today

- centring and scaling (and understanding a model)
- how to fit a polynomial model

A Question from Yesterday

"Where's the line?"



Getting the Line I

The model that was fitted was

$$y_i = \hat{\alpha} + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \varepsilon_i$$

(x_{i1} is Body Size, x_{i2} is Gape Size. The hats on Greek letters show that we are using the estimates of the parameters)

This code

```
abline(a = coef(BSModel)["(Intercept)"],
       b = coef(BSModel)["BodySize"])
```

draws the line

$$y_i = \hat{\alpha} + \hat{\beta}_1 x_{i1}$$

Getting the Line II

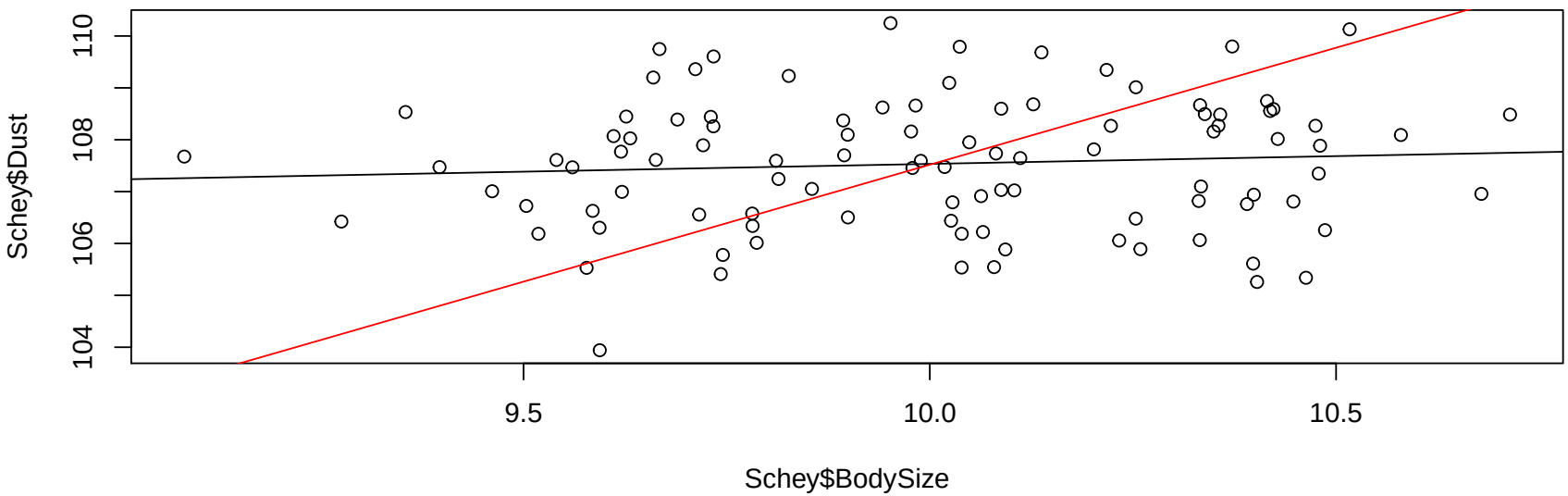
As we are plotting against x_{i1}, we have to do something with x_{i2}

$$y_i = \hat{\alpha} + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2}$$

Getting the Line II

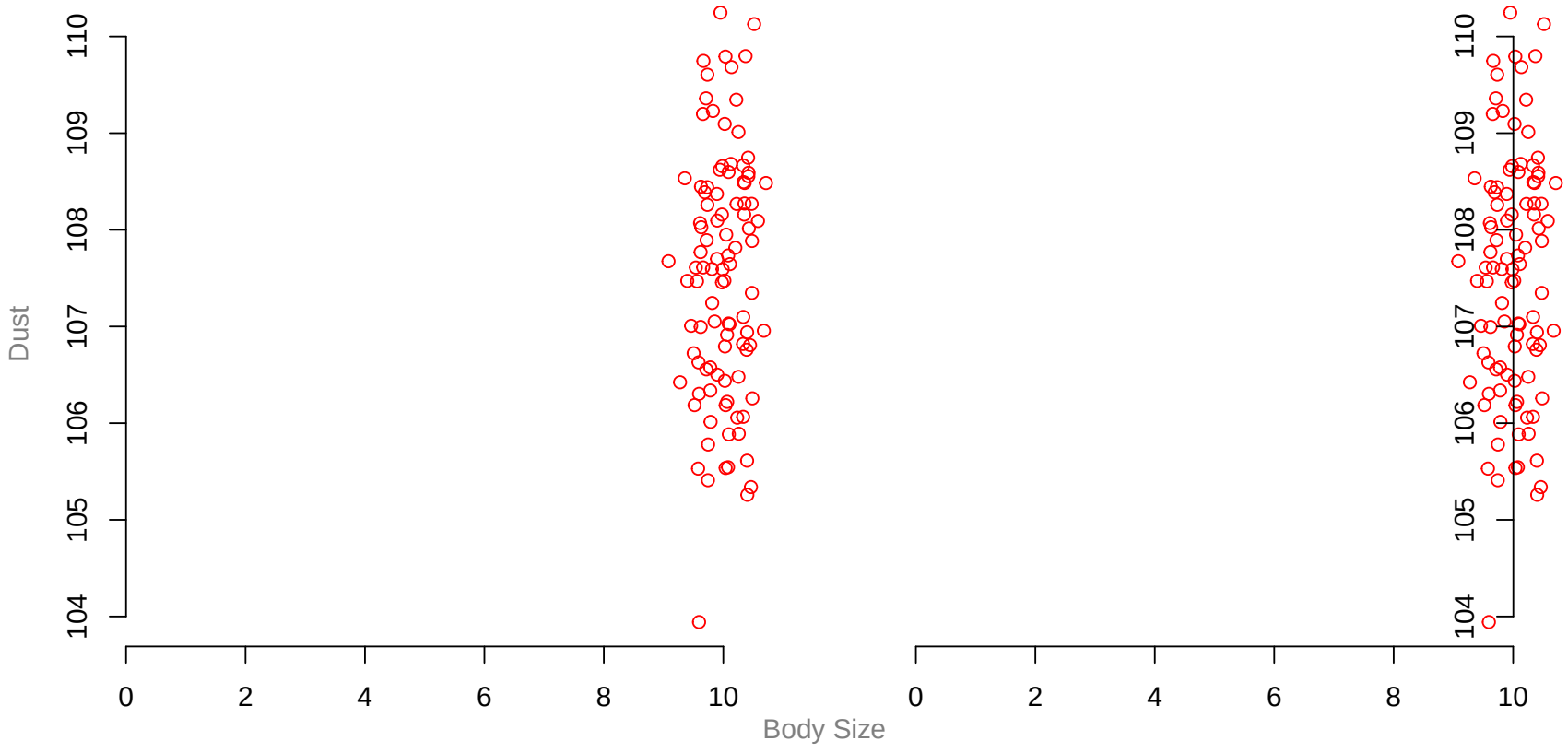
A simple remedy is to set it to the mean:

```
Better.a <- coef(FullModel)["(Intercept)"] +
  coef(FullModel)["GapeSize"]*mean(Schey$GapeSize)
plot(Schey$BodySize, Schey$Dust)
abline(a=coef(BSModel)["(Intercept)"],
       b = coef(BSModel)["BodySize"])
abline(a=Better.a, b = coef(FullModel)["BodySize"], col=2)
```



Mean Centering: getting the line

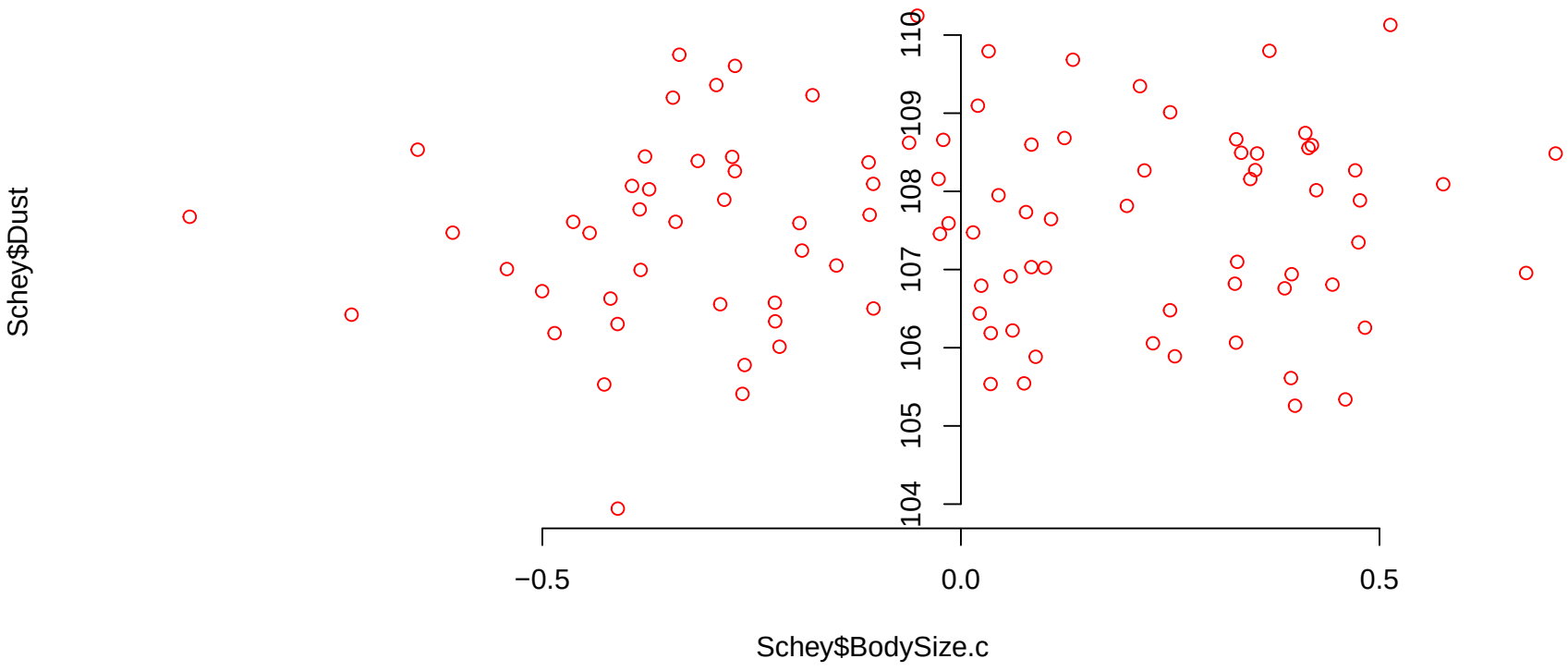
Another approach is to move the intercept



Mean Centring: getting the line

In practice this just means subtracting the mean from Body Size:

```
Schey$BodySize.c <- Schey$BodySize - mean(Schey$BodySize)
plot(Schey$BodySize.c, Schey$Dust, col=2,
     yaxt="n", bty="n")
axis(2, pos=0)
```



Your task

```
Schey$BodySize.c <- Schey$BodySize - mean(Schey$BodySize)
Schey$GapeSize.c <- Schey$GapeSize - mean(Schey$GapeSize)

FullModel <- lm(Dust ~ GapeSize + BodySize,
                data=Schey)
FullModel.c <- lm(Dust ~ GapeSize.c + BodySize.c,
                  data=Schey)
```

Fit the models with the un-centred and centred Body Size and Gape Size. Look at the parameters (with `coef()`), and discuss any differences.

Can you interpret the parameters?

Scaling

I mentioned that we could measure body size in kg:

```
Schey$BodySize.kg <- Schey$BodySize/1000
mod.kg <- lm(Dust ~ GapeSize + BodySize.kg, data=Schey)

round(coef(mod.kg), 2)
```

```
## (Intercept)    GapeSize BodySize.kg
##        9.38       10.62     4509.23
```

The effect of body size is massive!

Discussion

Why is the effect so massive?

How do you interpret the regression coefficients? They say something about the change in Dust when body size changes, but can you say what?

- yes, they are the slope, but what do they say biologically?
- can you interpret the slopes in terms of predictions?

Standardisation

As well as centring the predictors, we can standardise them.

```
Schey$BodySize.s <- (Schey$BodySize - mean(Schey$BodySize))
  sd(Schey$BodySize)
Schey$GapeSize.s <- scale(Schey$GapeSize)
```

The first does it "by hand", the second uses an R function. Both do the same thing

Interpreting the Standardised Model

```
FullModel.s <- lm(Dust ~ GapeSize.s + BodySize.s,
                  data=Schey)
round(coef(FullModel.s), 3)
```

Fit the model with the standardised coefficients

Can you interpret the standardised coefficients?

When might you prefer to use the standardised or un-standardised models?

Addendum: the models are the same

A quick bit of maths. The standardised model (for one variable) is

$$y_i = \alpha + \beta \frac{(x_i - \bar{x})}{s_x} + \varepsilon_i$$

where $\bar{x}$ is the mean of x and s_x is the standard deviation of x. We can expand the brackets and re-arrange to get

$$y_i = \alpha + \beta x_i / s_x - \beta \bar{x} + \varepsilon_i$$

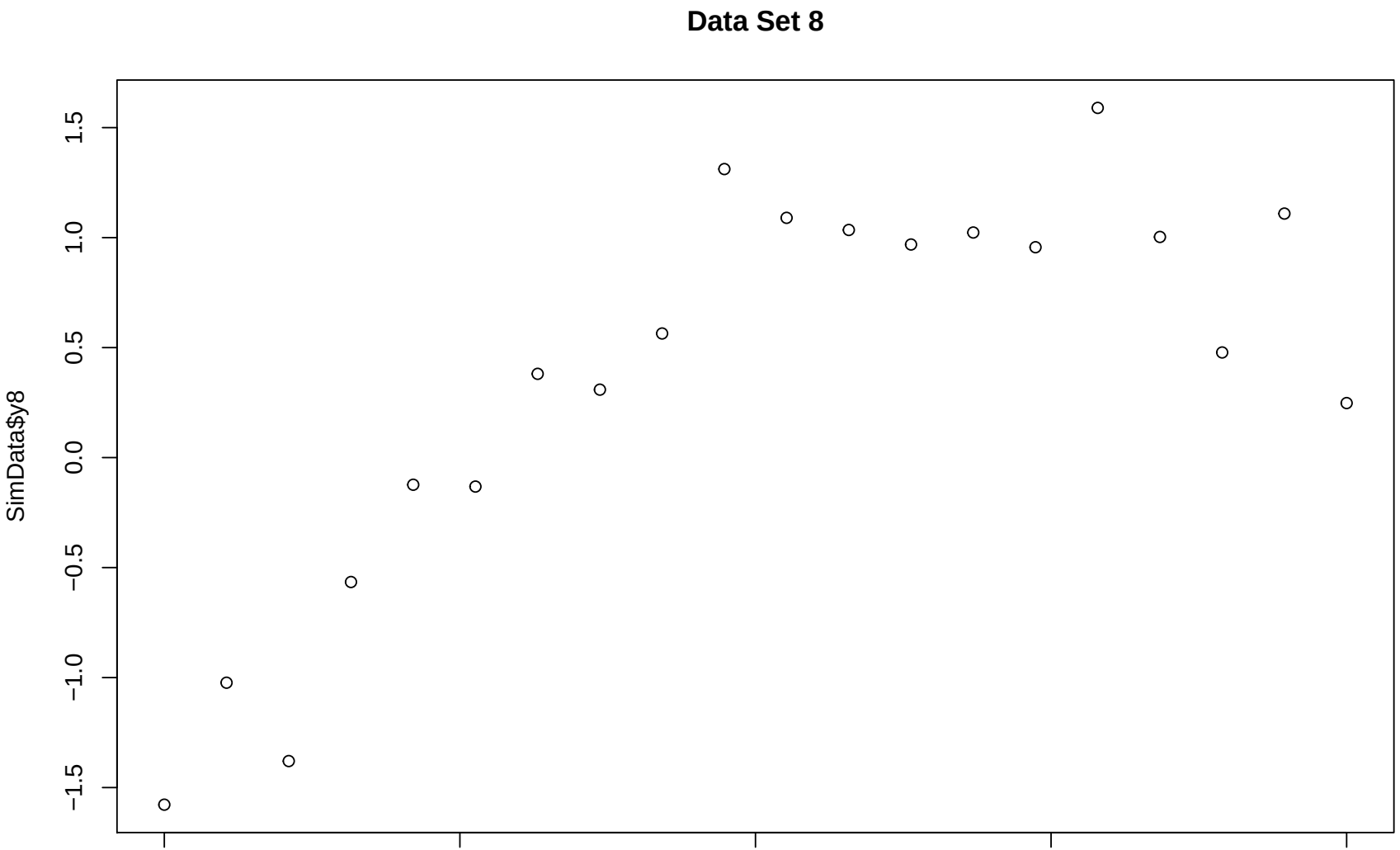
But $\bar{x}_{.j}$ is a constant - it does not vary for different y's, so we have the same model, but with

$\alpha^* = \alpha - \sum_{j=1}^{p} \frac{\beta_j}{s_j} \bar{x}_{.j}$ and $\beta_j^* = \frac{\beta_j}{s_j}$

Polynomials

Back to Data Set 8 last week. . .

```
SimData <- read.csv("https://www.math.ntnu.no/emner/ST2304/
plot(SimData$x, SimData$y8, main="Data Set 8")
```



Approximating curves

We can approximate any reasonable curves with a Taylor series:

$$f(x) \approx \beta_0 + \beta_1(x - \bar{x}) + \beta_2(x - \bar{x})^2 + \beta_3(x - \bar{x})^3 + \cdots + \beta_p(x - \bar{x})^p$$

So we can fit an approximate curve by regressing Y against X, X^2, x^3 etc.

(we don't have to centre, of course)

Fitting in R

We can simply treat the extra terms as additional variables

```
linmod <- lm(y8 ~ x, data=SimData)
quadmod <- lm(y8 ~ x + I(x^2), data=SimData)
```

Your tasks:

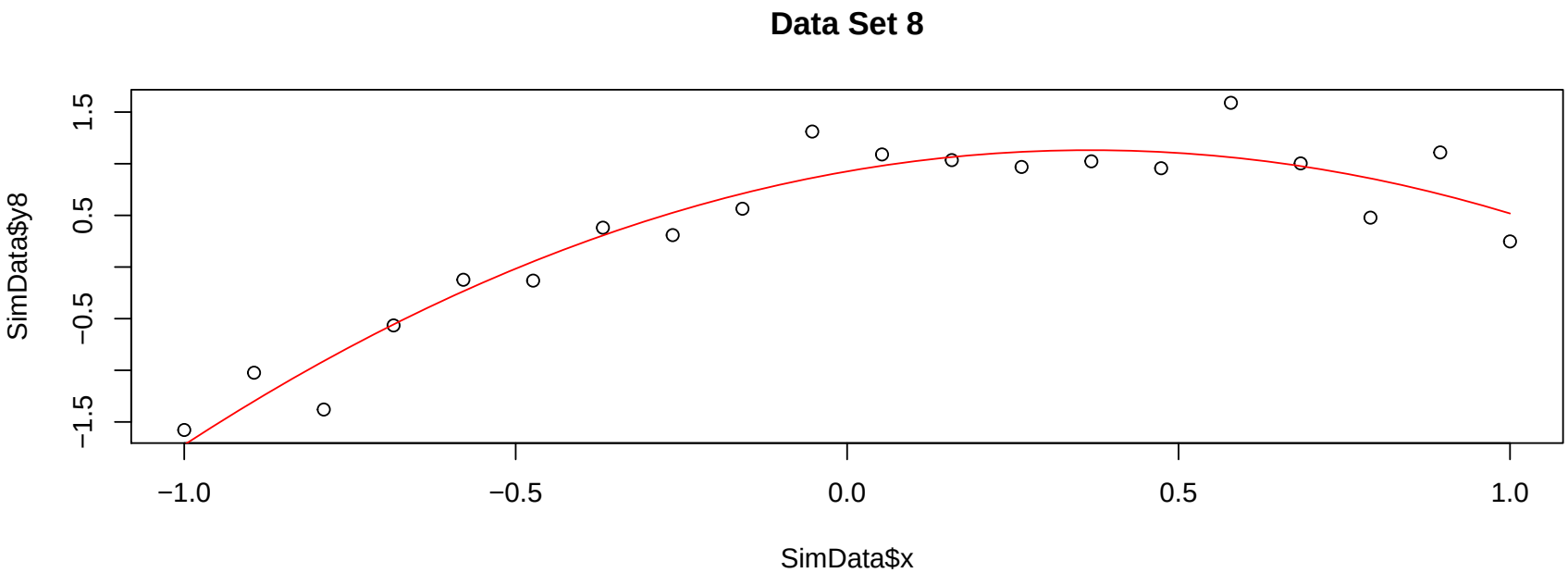
- fit the linear and quadratic models
- fit the linear and quadratic models after standardising x

Does the quadratic model fit better? Are the parameters different? What happens if you add an x^3 term?

Plotting a polynomial

Unfortunalely `abline()` won't work. Instead we can predict new data, and plot that:

```
PredData <- data.frame(x=seq(min(SimData$x),
                             max(SimData$x), length=50))
PredData$y.quad <- predict(quadmod, newdata = PredData)
plot(SimData$x, SimData$y8, main="Data Set 8")
lines(PredData$x, PredData$y.quad, col=2)
```



Today: a summary

- centring and scaling (and understanding a model)

We can now centre and scale models. This can make interpretation easier

- how to fit a polynomial model

We can fit polyomial model: `lm(y ~ x + I(x^2))`