Multiple Regression

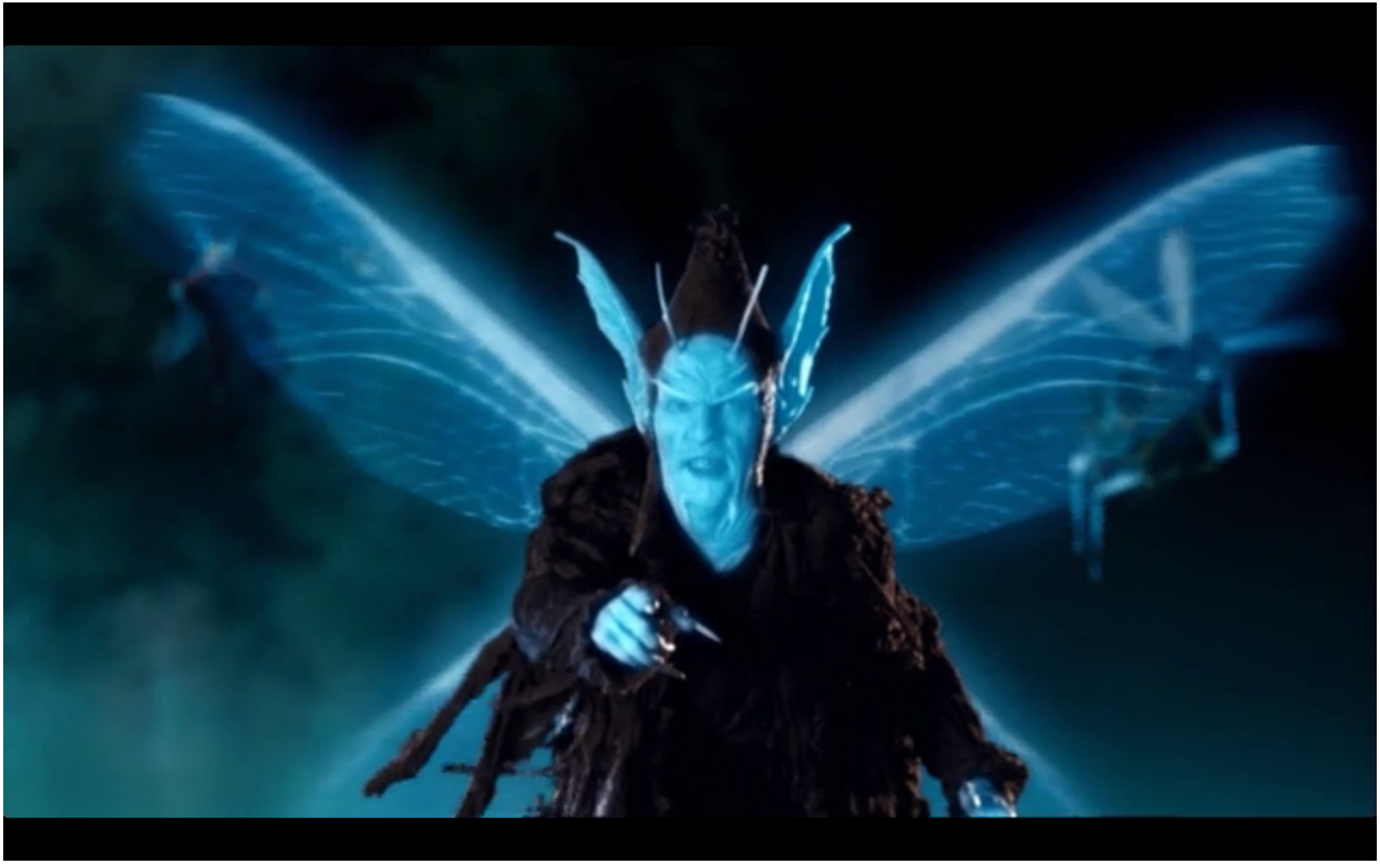
Bob O'Hara

This week: Multiple Regression

We will look at

- explaining our dependent variable with more than one explanatory variable
- how to fit these models in R
- what a design matrix is (this will be helpful later)
- how to fit a polynomial model

More Monsters



More Monsters

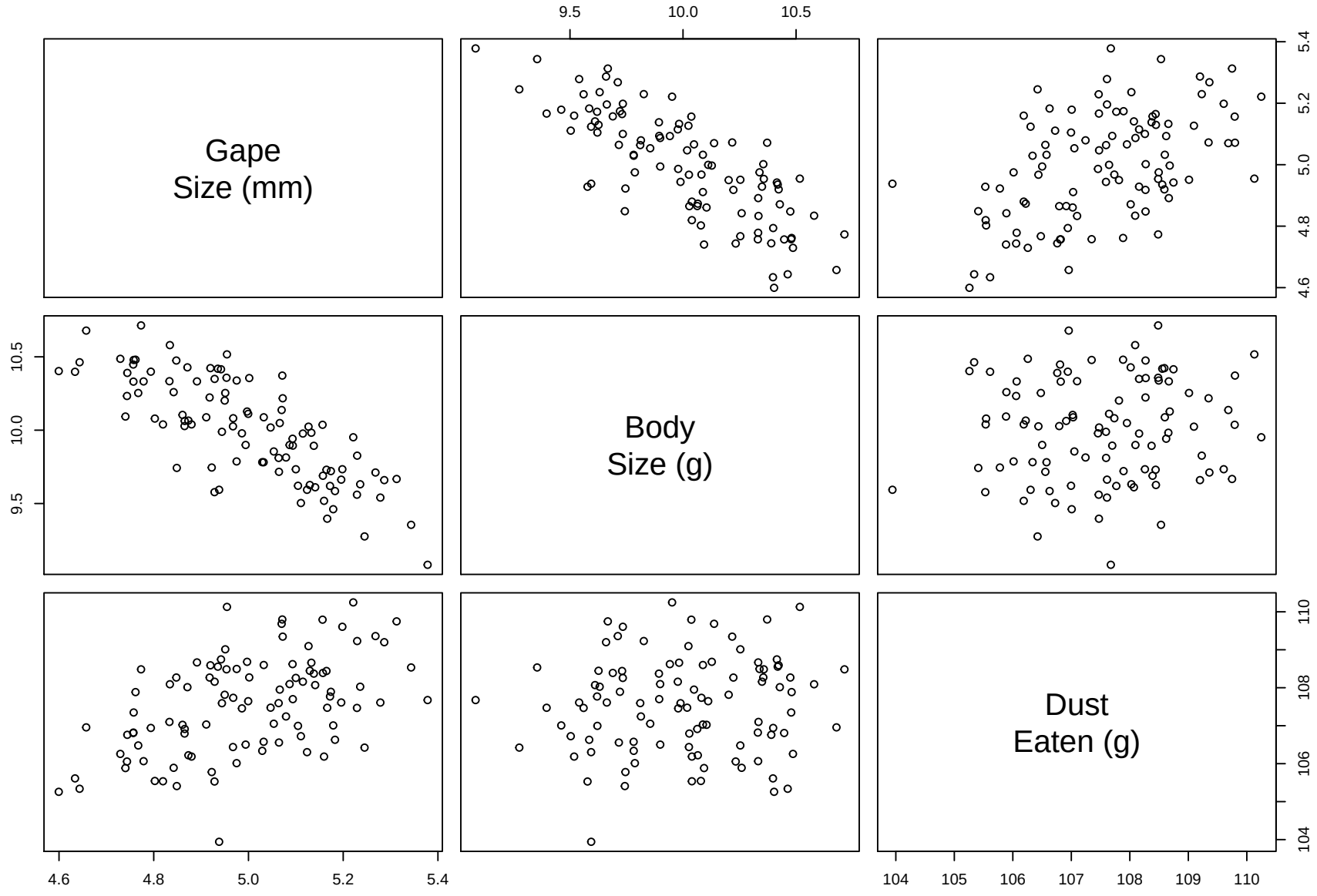
In the cellar of the museum in Frankfurt we had a population of Schey.

These are small creatures that lurk in the dark and eat ancient dust and stale cobwebs.

Some of us wanted to know more about them, and whether they could be trained to clean the museum collections.

We caught 100 and measured the amount of dust they could eat in 5 mins, and wanted to explain that by their body size, their gape size (i.e. how large their mouths are).

The Data


Gape
Size (mm)
Body
Size (g)
Dust
Eaten (g)
9.5
10.0
10.5
4.6
4.8
5.0
5.2
5.4
104
105
106
107
108
109
110

Simple regression

```
Dir <- "https://www.math.ntnu.no/emner/ST2304/2019v/"
File1 <- "Week7/ScheyData.csv"
Schey <- read.csv(paste0(Dir, File1))
plot(Schey, labels=c("Gape\nSize (mm)", "Body\nSize (g)",
                     "Dust\nEaten (g)"))
```

What if we have >1 predictor?

We often want to look at the effects of several variables together

- they may all have some effect
- we might be doing an experiment where factors interact
- we might want to model one variable as a polynomial

The model

This is our model for simple regression

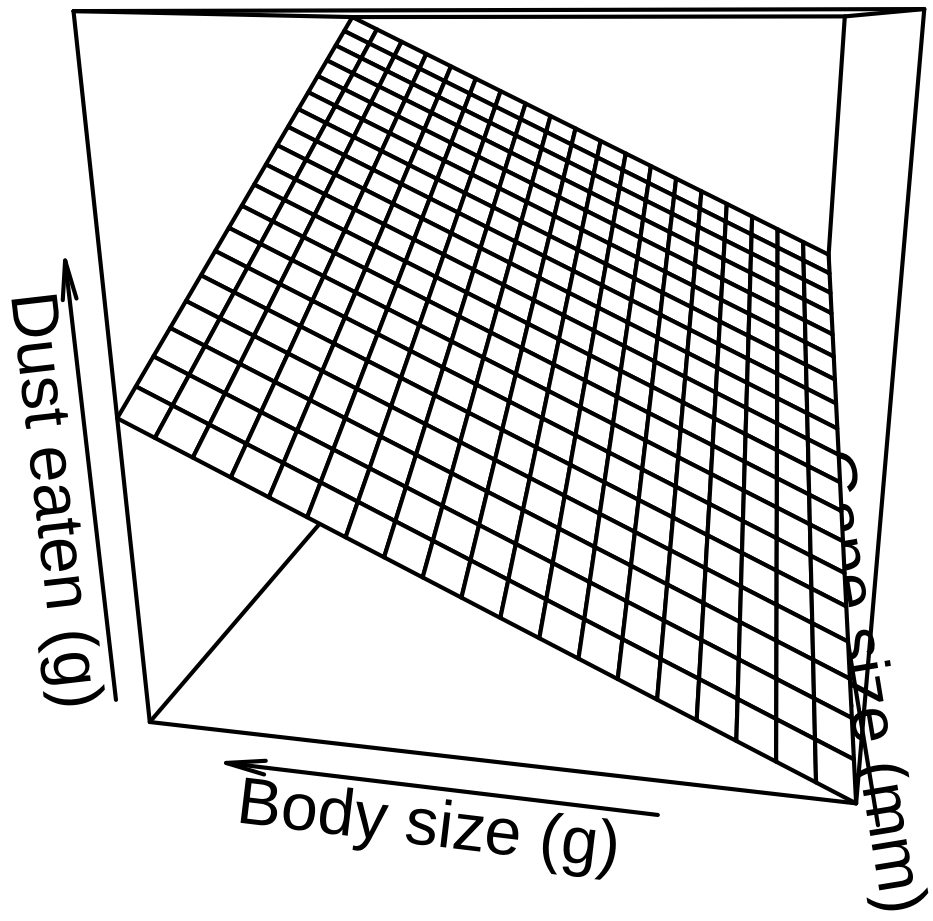
$$y_i = \alpha + \beta x_i + \varepsilon_i$$

How can we extend it to more than one variable?

The obvious model

$$E(y_i) = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i}$$

This is a plane

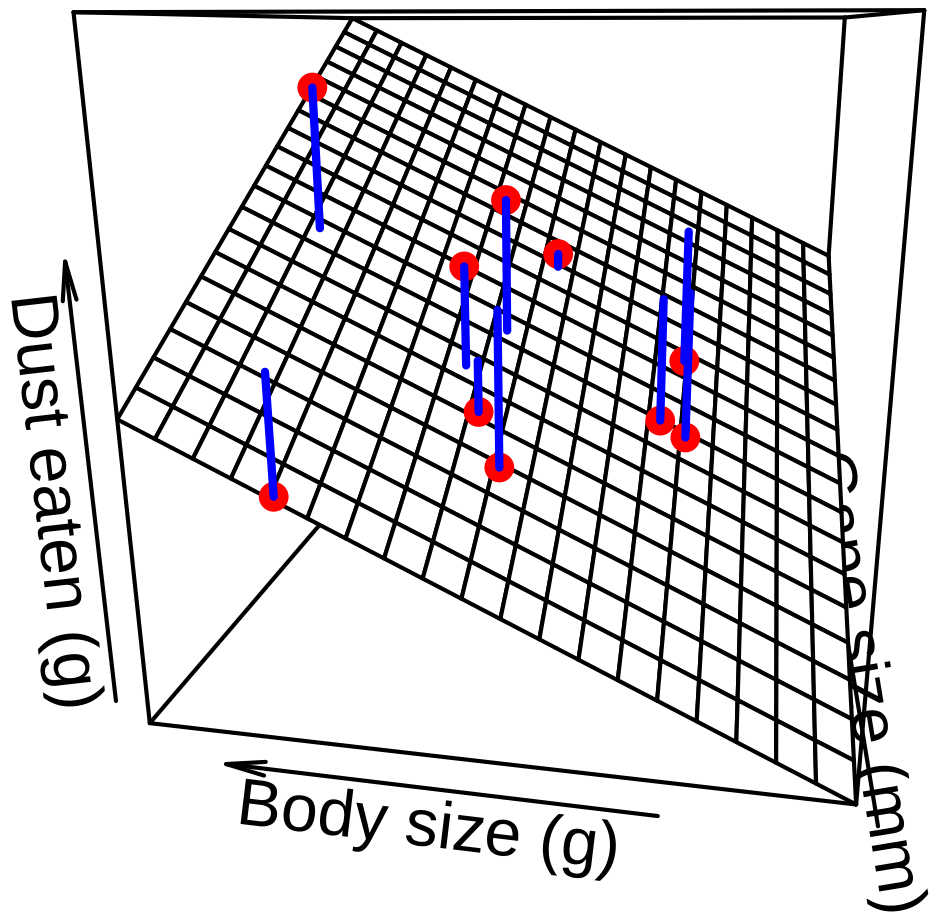


The obvious model

The model for the data is thus

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i$$

The points deviate from the plane



Fitting in R

In R we can just use the same function as we did before.

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

The only change is in the formula. It was

`Y ~ X`

now it is

`Y ~ X1 + X2`

Your Turn I

- first fit the model with each covariate individually (i.e. first explain dust eaten by gape size, then explain dust eaten by body size).
  - use `summary()` to look at the parameter estimates and R^2. Write down the regression models (i.e. plug the correct values into $E(y_i) = \alpha + \beta_1 x_i$)
  - What do the models suggest are the effects on dust eating, and how well do the variables individually explain the variation in the response?

Your Turn I: Gape Size

```
GapeMod <- lm(Dust ~ GapeSize, data=Schey)
summary(GapeMod)$coefficients
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 89.504179  3.1553839 28.365543 4.790278e-49
## GapeSize     3.608754  0.6311464  5.717776 1.168581e-07
```

```
summary(GapeMod)$r.square
```

```
## [1] 0.2501509
```

So the model is $y_i = 89.5 + 3.6\ x + \varepsilon_i$

Your Turn I: Body Size

```
BodyMod <- lm(Dust ~ BodySize, data=Schey)
summary(BodyMod)$coefficients
```

```
##                 Estimate Std. Error    t value      Pr(>|t
## (Intercept) 104.5336017  3.6719402 28.4682203 3.491845e-
## BodySize      0.3000321  0.3668415  0.8178792 4.154103e-
```

```
summary(BodyMod)$r.square
```

```
## [1] 0.006779503
```

So the model is $y_i = 104.5 + 0.3\ x + \varepsilon_i$

Your Turn I: Individual regressions

Body size seems to have little effect - R^2 is 0.68 %. Gape size seems to be more important, explaining 25 % of the variation.

The effect is positive: changing the gape size by 1 mm increases the amount of dust eaten by 3.6 g

Your Turn I: Joint regression

- fit a model with both covariates (i.e. explain dust eaten by both gape size and body size).
  - again, use `summary()` to look at the parameter estimates and R^2.Write down the regression model.
  - What does this model suggest are the effects on dust eating, and how well do the variables together explain the variation in the response?
  - How do these results compare to those from the single regression models?

Your Turn I: Joint regression

```
FullMod <- lm(Dust ~ BodySize + GapeSize, data=Schey)
summary(FullMod)$coefficients
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)  9.376014  4.5202646  2.074218 4.070794e-02
## BodySize     4.509229  0.2404605 18.752475 5.009431e-34
## GapeSize    10.617635  0.4761362 22.299577 5.827119e-40
```

```
summary(FullMod)$r.square
```

```
## [1] 0.8378814
```

So the model is $y_i = 9.4 + 4.5\ x_{1i} + 10.6\ x_{2i} + \varepsilon_i$ (x_{1i} is Body Size, x_{2i} is Gape Size)

Your Turn I: Joint regression

The joint model explains much more of the variation - R^2 is now 83.8 %. The estiamted coefficients are also much larger.

- the effect of body size has changed from 0.3 to 4.5
- the effect of gape size has changed from 3.6 to 10.6

So, the model is better, and the estimated effects are much larger.

Regression More Generally

$$y_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

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

- we have p covariates, labelled from $j = 1$ to p
- we have p covariate effects
- the jth covariate values for the ith individual is x_{ij}

Design Matrices

We can write this more compactly. First, we turn the intercept into a covariate by using a covariate with a value of 1 for every data point. Then we write all of the covariates in a matrix, X:

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

So, the first column is the intercept, the second is the first covariate, and the third is the second covariate.

This is called the *Design Matrix*: it is helpful for writing down the model

Writing the Model

Using matrix algebra, the regression model becomes

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

where $\mathbf{Y}$, β and ε are now all vectors of length n, where there are n data points. X is am $n \times p$ matrix.

We will not look at the mathematics in any detail: the point here is that the model for the effect of covariates can be written in the design matrix.

Writing the Model

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

is

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & 2.3 & 3.0 \\ 1 & 4.9 & -5.3 \\ 1 & 1.6 & -0.7 \\ \vdots & \vdots & \vdots \\ 1 & 8.4 & 1.2 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

- β_0 is the intercept

The Solution (just so you can see it)

After a bit of matrix algebra, one can find the ML solution:

$$\mathbf{b} = (X^T X)^{-1} X^T \mathbf{Y}$$

where $\mathbf{b}$ is the MLE for β.

In practice:

- you won't have to calculate this: the computer does it, and
- the computer actually doesn't use this

Multiple Regression Today

We can now write a multiple regression model

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

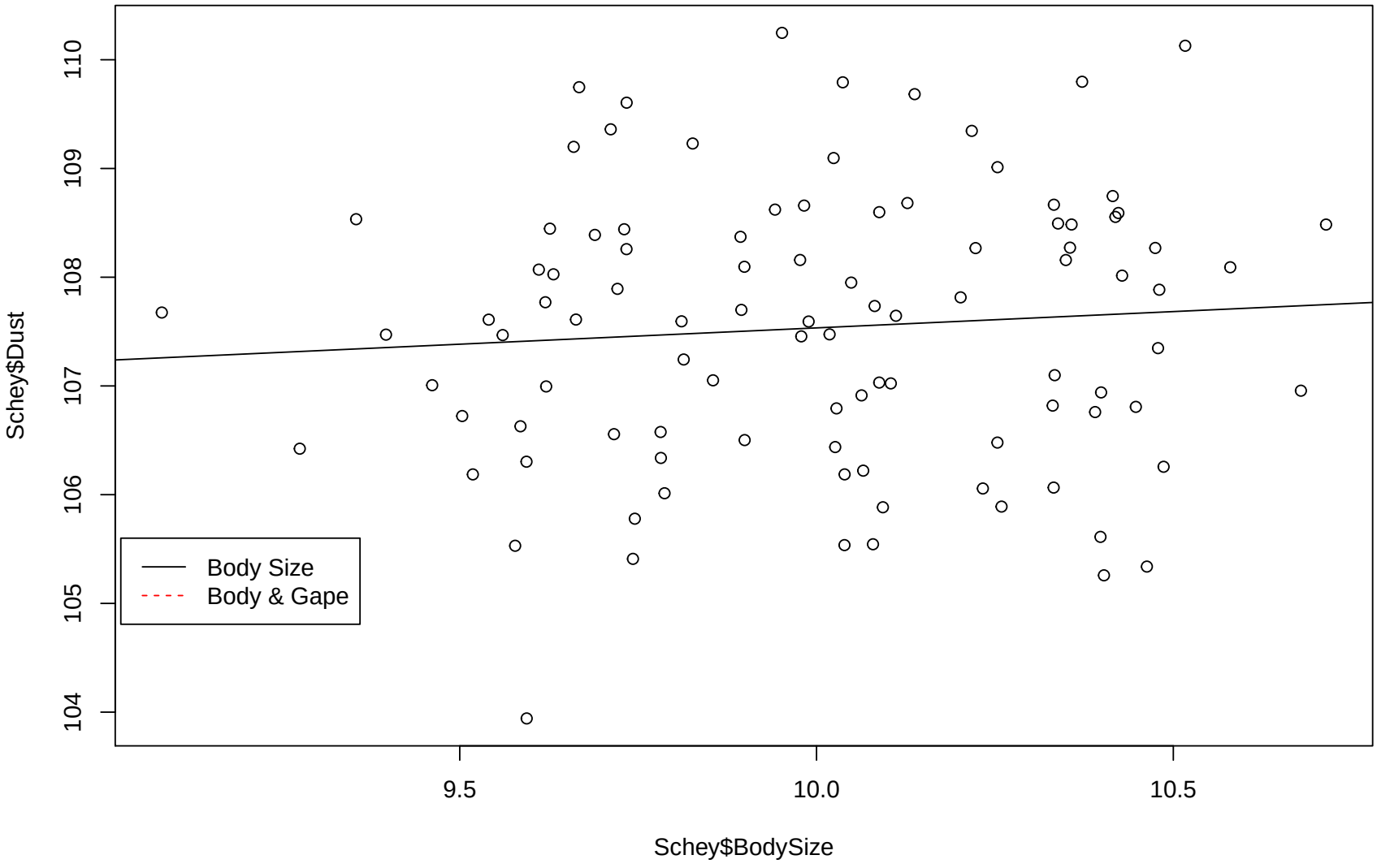
We can fit it in R

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

We know what a design matrix looks like

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

Where's the line in the regression plot?



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

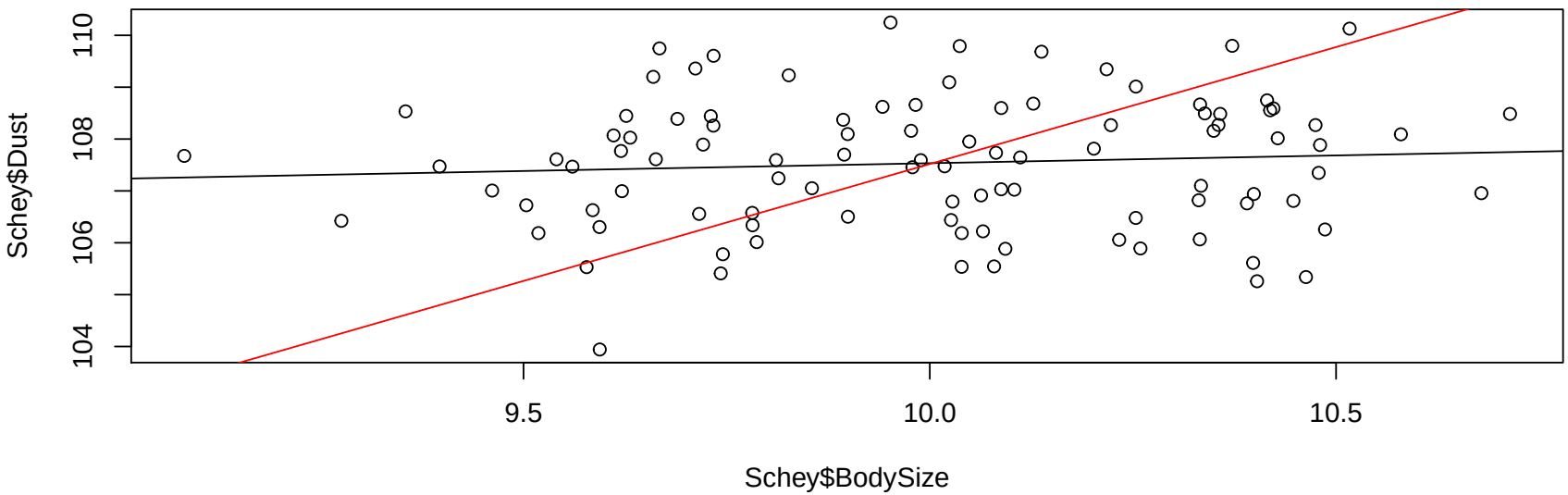
As we are plotting against x_{i1}, we have to do something with x_{i2}

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

Getting the Line II

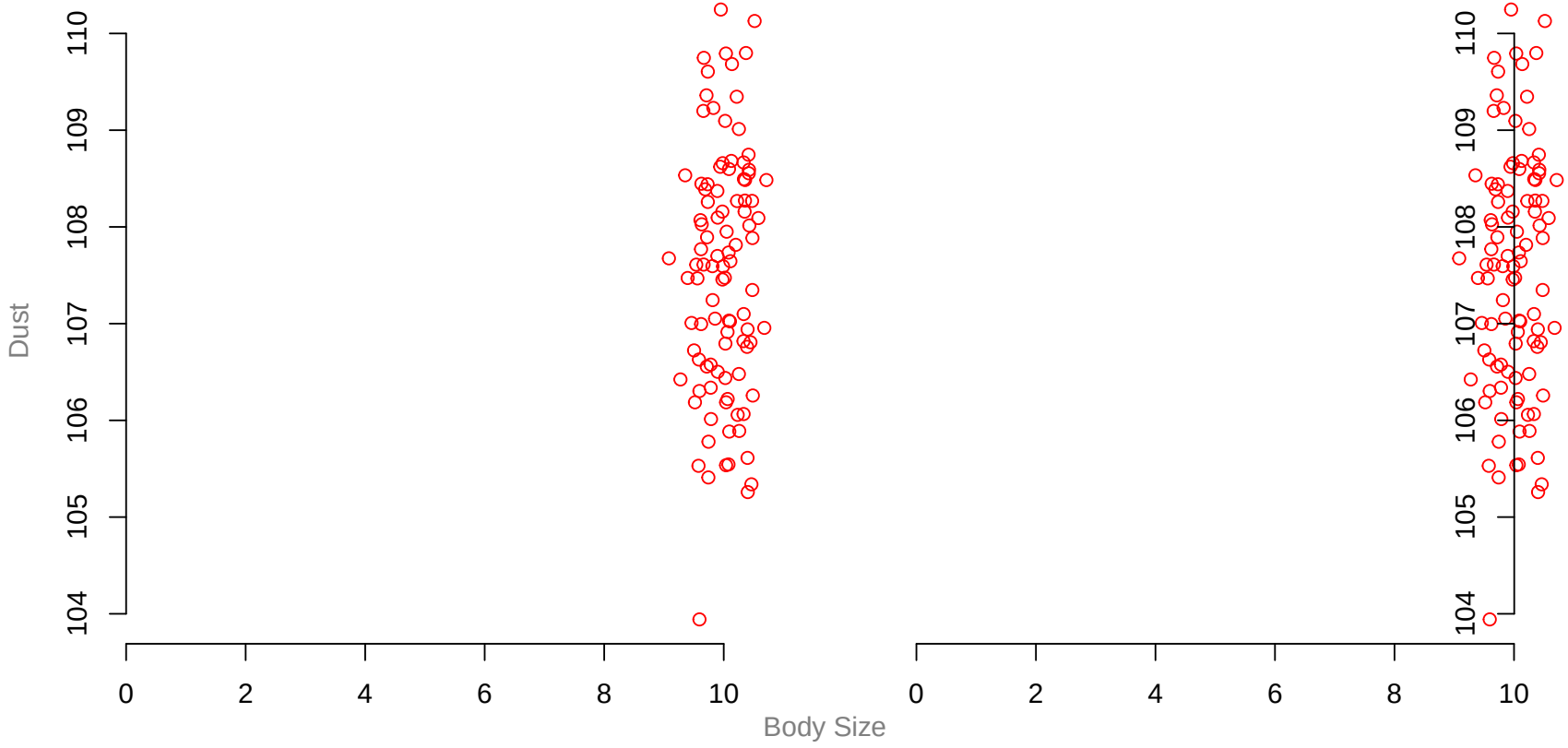
A simple remedy is to set it to the mean:

```
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

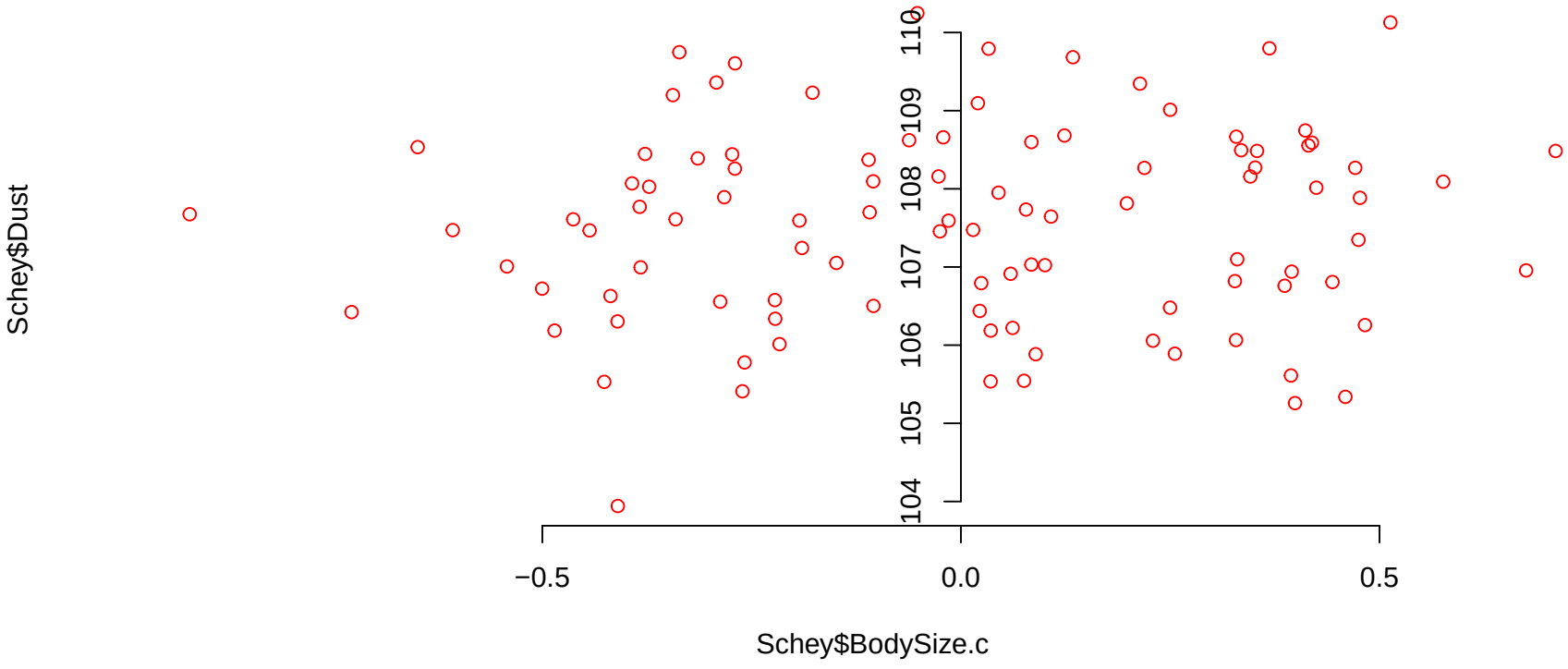
Another approach is to move the intercept



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-centered and centered 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 centering 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

Your task: Centering

- Fit the models with the un-centered and centered Body Size and Gape Size. Look at the parameters (with `coef()`), and discuss any differences.
- Can you interpret the parameters?

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

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

Your task: Centering

```
coef(FullMod)
```

```
## (Intercept)    GapeSize    BodySize
##    9.376014   10.617635    4.509229
```

```
coef(FullMod.c)
```

```
## (Intercept)  GapeSize.c  BodySize.c
##  107.535025   10.617635    4.509229
```

The slopes (i.e. the effects of body size and gape size) are the same, but the intercept has changed.

For the centered model, the coefficient is now 107.54, which is close to the mean of the response, 107.54 g

Standardisation

STOPPED HERE

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

```
## (Intercept)  GapeSize.s  BodySize.s
##     107.535       1.857       1.562
```

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

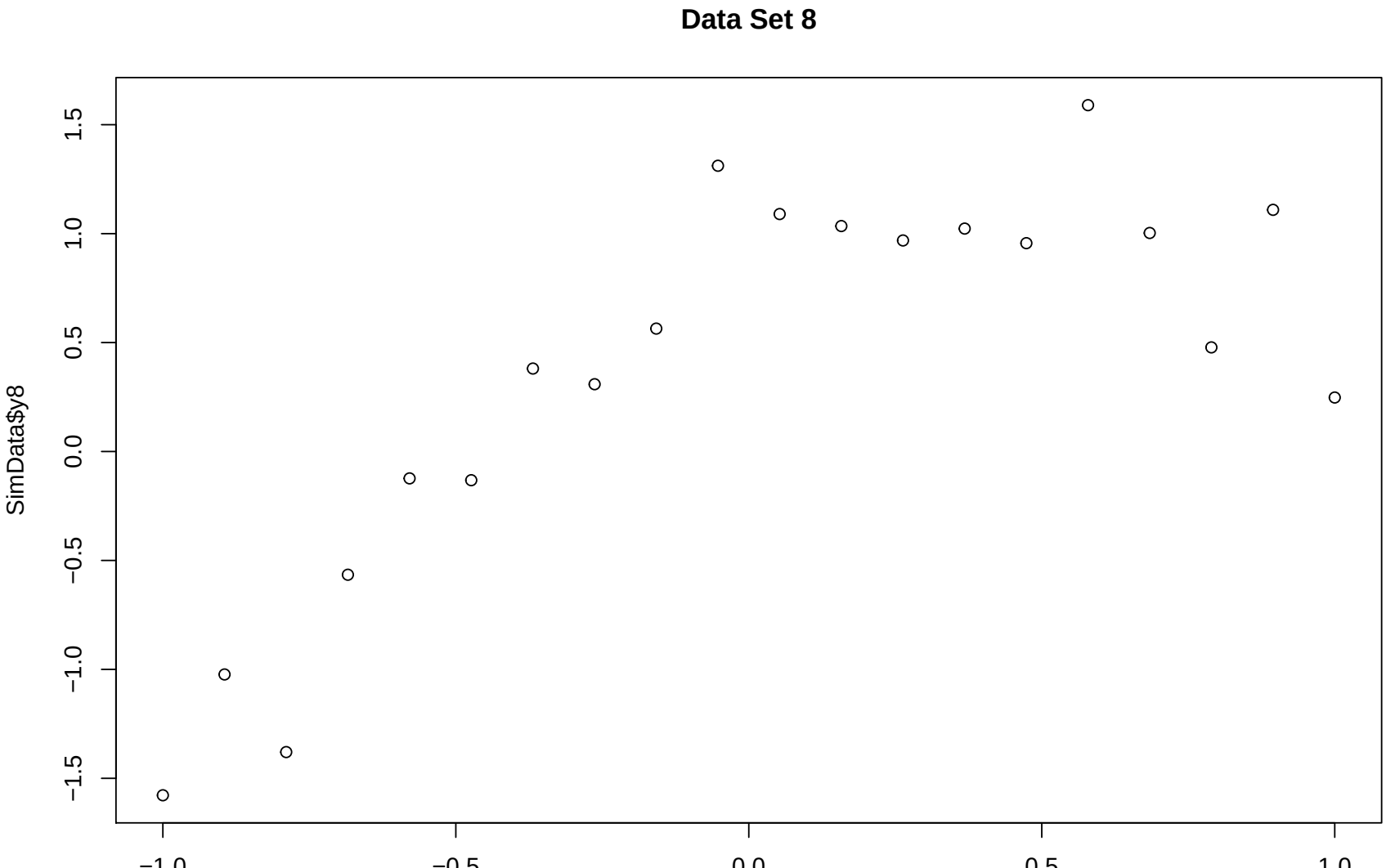
The slope say that when we change the covariate by 1 unit (e.g. from 100g to 101g), the response changed by that amount. This is why the coefficient is so massive when we convert to kilograms - the coefficient is the difference in dust consumption if between Schey that have 1kg difference in weight.

In the standardised model, the change is by one standard deviation

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:

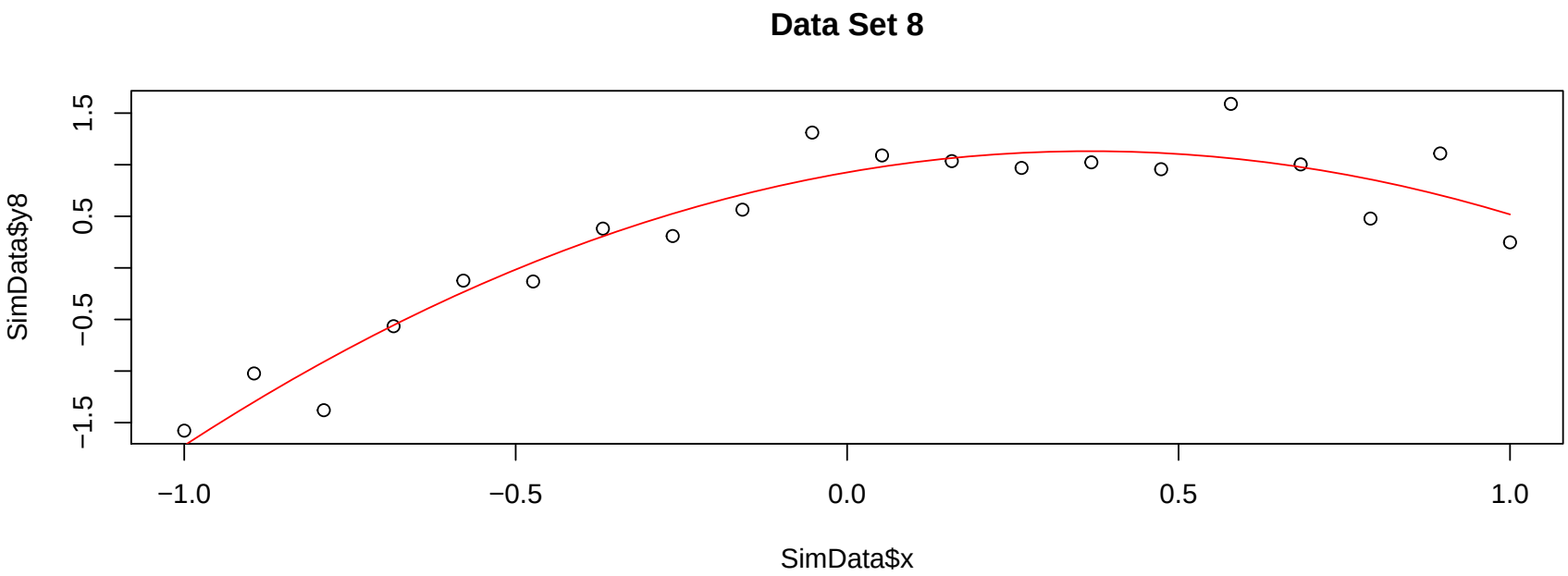
- fit the linear and quadratic models
- fit the linear and quadratic models after standardising x

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)
```



Polynomial Tasks: Does the quadratic model fit better?

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

summary(linmod)$r.square
```

```
## [1] 0.590943
```

```
summary(quadmod)$r.square
```

```
## [1] 0.9124822
```

The R^2 for the linear model is 59%, which is fairly good, but the quadratic model is much better, with an R^2 of 91%. We will see the improvement in a couple of slides' time, when we look at the plot.

Polynomial Tasks: Are the parameters different?

```
# could also look at summary()
coef(linmod); confint(linmod)
```

```
## (Intercept)           x
##   0.3632011   1.1201099
```

```
##                  2.5 %    97.5 %
## (Intercept) 0.08309298 0.6433092
## x           0.65862933 1.5815906
```

```
coef(quadmod); confint(quadmod)
```

```
## (Intercept)           x      I(x^2)
##   0.9260229   1.1201099  -1.5276592
```

```
##                  2.5 %    97.5 %
## (Intercept)  0.7247749  1.127271
## x            0.8995350  1.340685
## I(x^2)      -1.9354878 -1.119830
```

We can see that when we add the quadratic term, the intercept

Polynomial Tasks: Are the parameters different?

```
SimData$x.uc <- SimData$x - 1
linmod.uc <- lm(y8 ~ x.uc, data=SimData)
quadmod.uc <- lm(y8 ~ x.uc + I(x.uc^2), data=SimData)

# could also look at summary()
coef(linmod.uc)
```

```
## (Intercept)        x.uc
##    1.483311    1.120110
```

```
coef(quadmod.uc)
```

```
## (Intercept)        x.uc   I(x.uc^2)
##   0.5184737  -1.9352084  -1.5276592
```

```
coef(quadmod)
```

```
## (Intercept)           x      I(x^2)
##   0.9260229   1.1201099  -1.5276592
```

If we move the intercept, we see that both the intercept and the

Polynomial Tasks: What happens if you add an x3 term?

```
cubmod <- lm(y8 ~ x + I(x^2) + I(x^3), data=SimData)

summary(quadmod)$r.square
```

```
## [1] 0.9124822
```

```
summary(cubmod)$r.square
```

```
## [1] 0.9163881
```

We should look at the full summary, but the interesting bit turns out to be the R^2. Adding the cubic term increases it by 0.4%, which is basically nothing. The cubic term estimate is -0.32, with a 95% confidence interval of -1.1 to 0.46, so we don't know what direction it is going in.

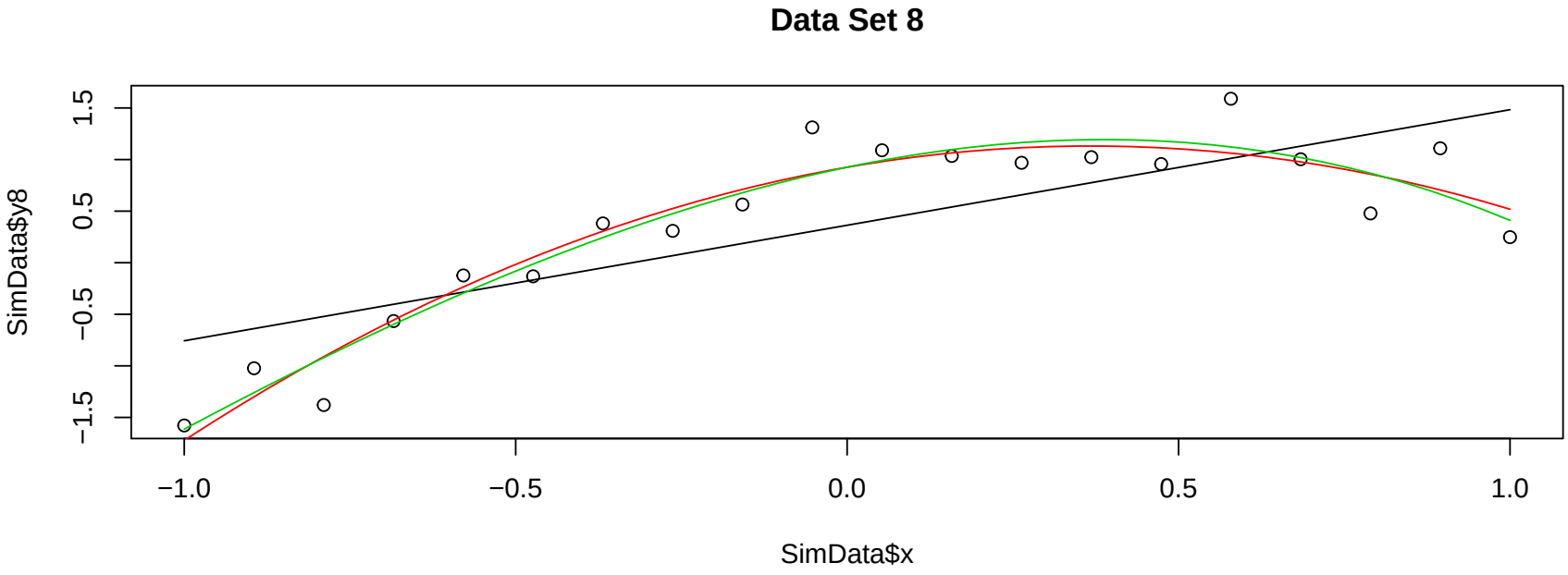
Polynomial Tasks: Plot the curves.

Here's the code. Hte plot is on the next page

```
PredData <- data.frame(x=seq(min(SimData$x),
                             max(SimData$x), length=50))
PredData$y.quad <- predict(quadmod, newdata = PredData)
PredData$y.lin <- predict(linmod, newdata = PredData)
PredData$y.cub <- predict(cubmod, newdata = PredData)

plot(SimData$x, SimData$y8, main="Data Set 8")
lines(PredData$x, PredData$y.lin, col=1)
lines(PredData$x, PredData$y.quad, col=2)
lines(PredData$x, PredData$y.cub, col=3)
```

Polynomial Tasks: Plot the curves.



The quadratic and cubic curves are very similar.

The quadratic curve is better, because it is simpler, and adding the cubic term barley improves the fit. We will find out later how to make this comparisom more formal.

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))`