How good is our straight line?

Bob O'Hara

Last Week

Last week we learned about regression: fitting straight lines

(show plot with residuals)

How good is my model? A Summary

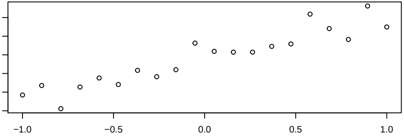
- Model as fit + residuals
- R^2: How much variation does the model explain?
- Residual plots
  - curvature
  - outliers
  - heteroscedasticity
- Normal Probability Plots
- Influential Points
- What to do to improve models

Exerise: looking at some models

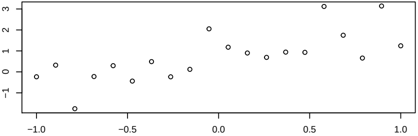
Here are some simulated data sets. For all of them I used the the same errors, but manipulated the data in different ways. For each one, you should decide

- if you think a straight line would be a good fit to the data, and
- if it is not, can you do something simple to improve the fit? (for some you cannot, for some you can)

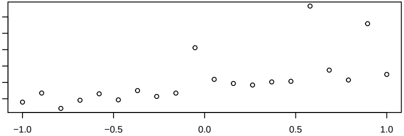
Data Set 1



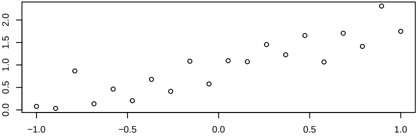
Data Set 2



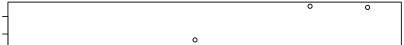
Data Set 3



Data Set 4

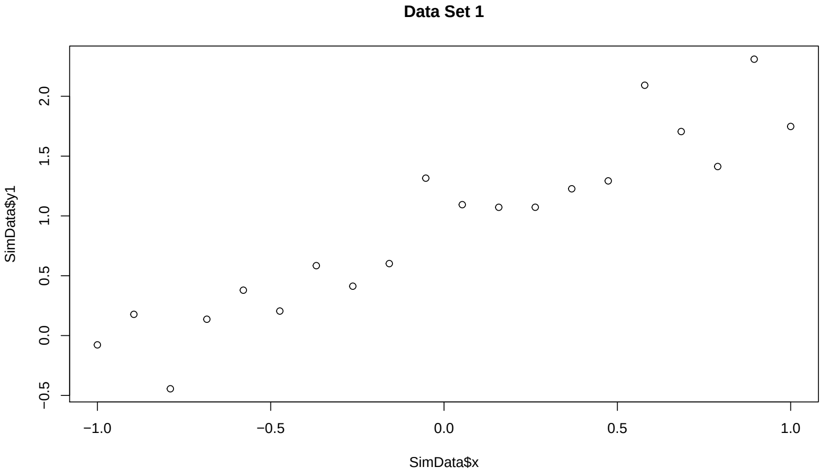


Data Set 5



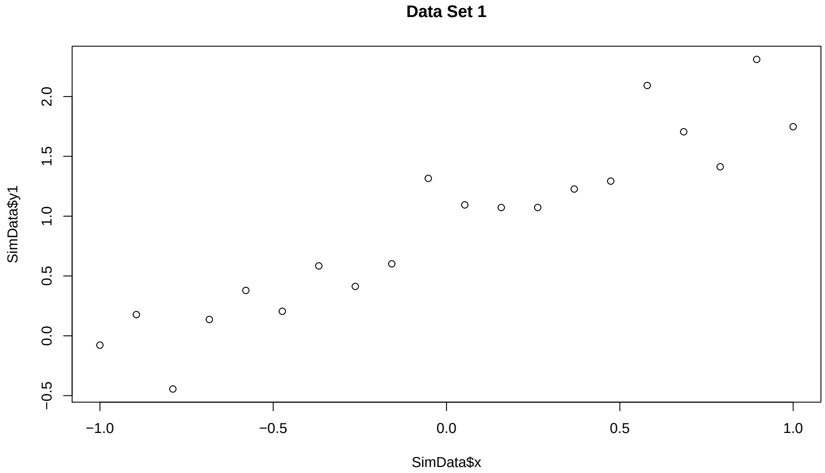
Data Set 6

Exercise: Data set 1



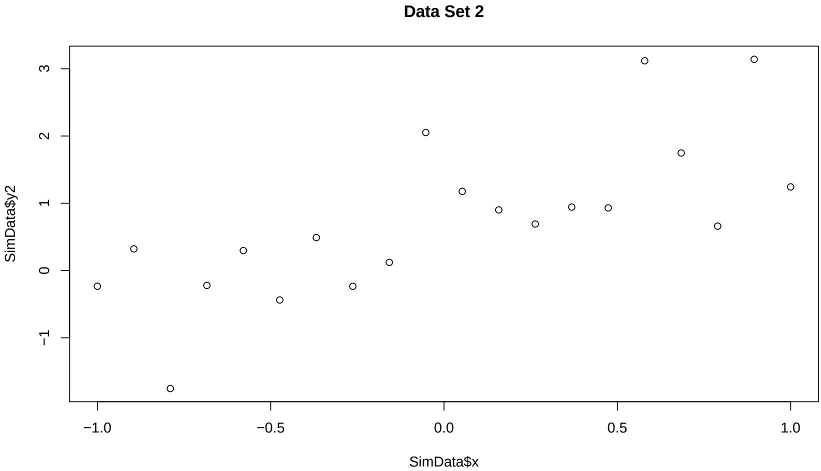
Comments?

Exercise: Data set 1

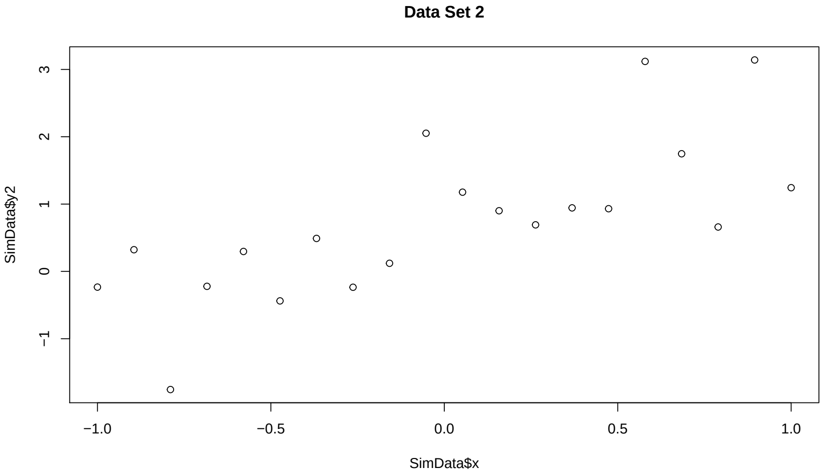


This looks OK, with a decent slope.

Exercise: Data set 2

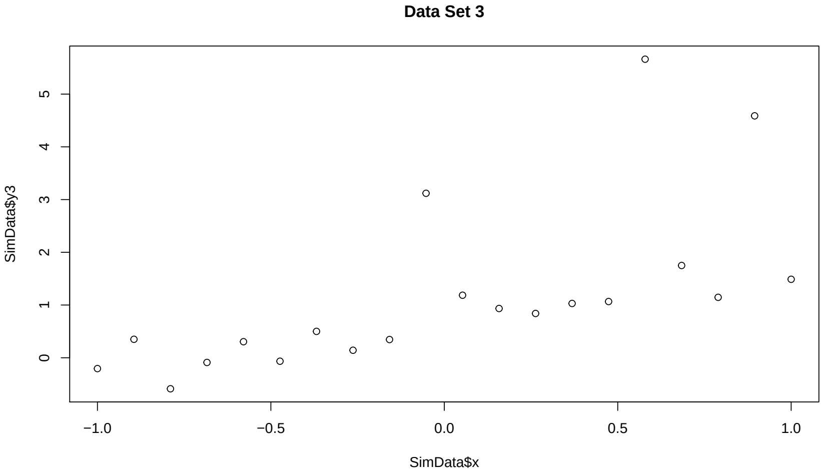


Exercise: Data set 2



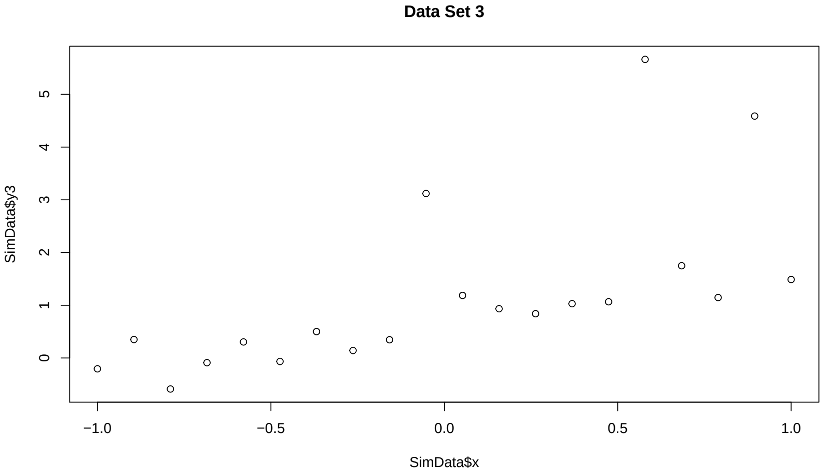
This is similar to data set 1, but x explain as much of the variation in y.

Exercise: Data set 3



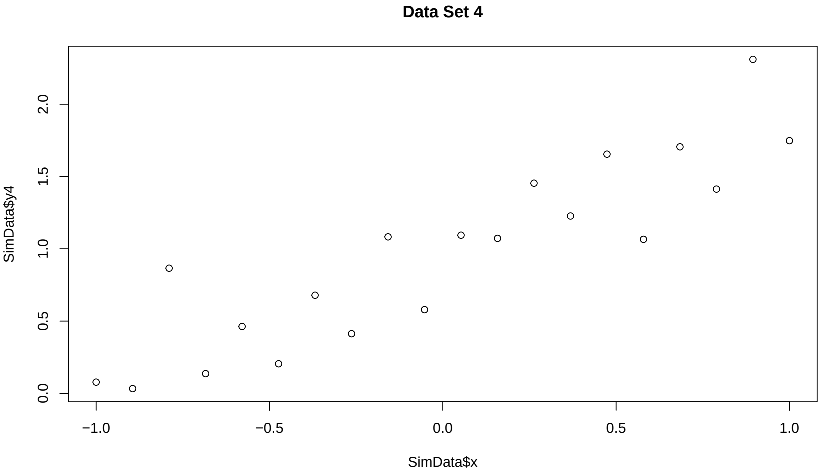
Comments?

Exercise: Data set 3



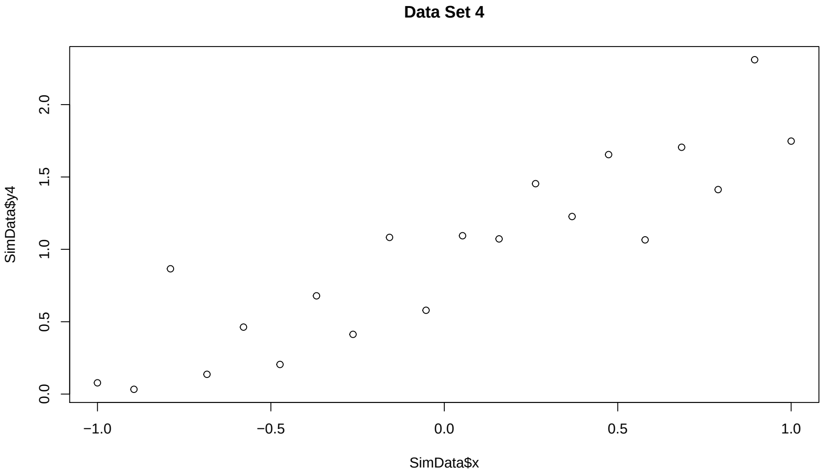
This looks OK, but there are 3 values that look too big.

Exercise: Data set 4



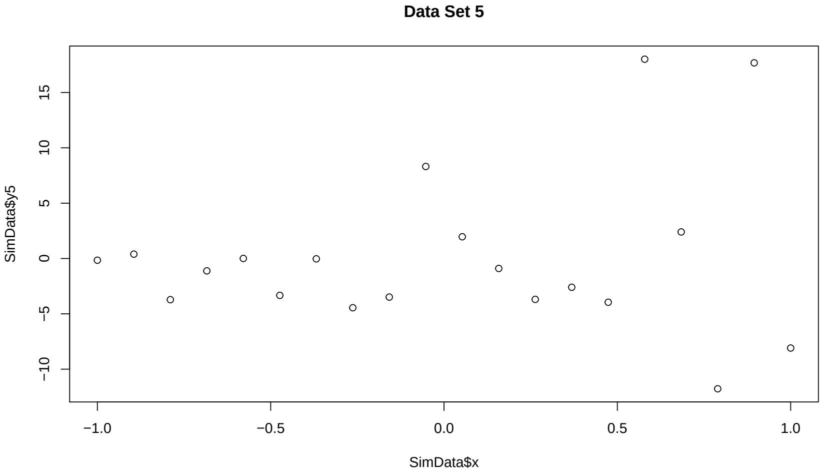
Comments?

Exercise: Data set 4



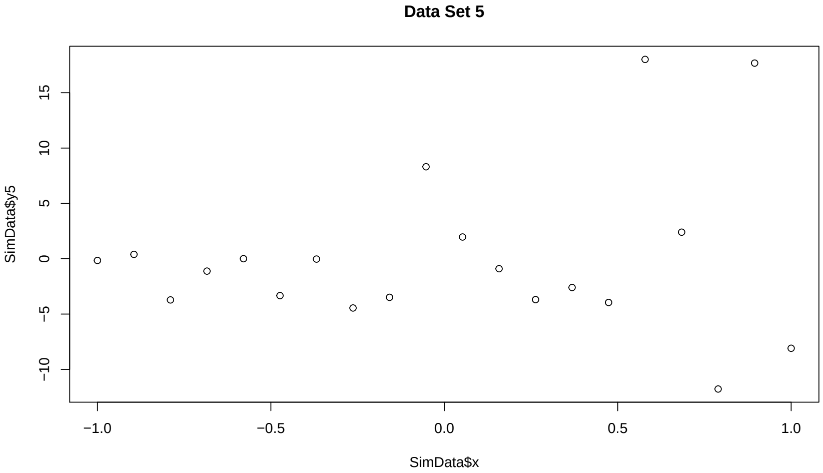
This looks OK, like Data Set 1. There is a issue here, but it's really subtle.

Exercise: Data set 5



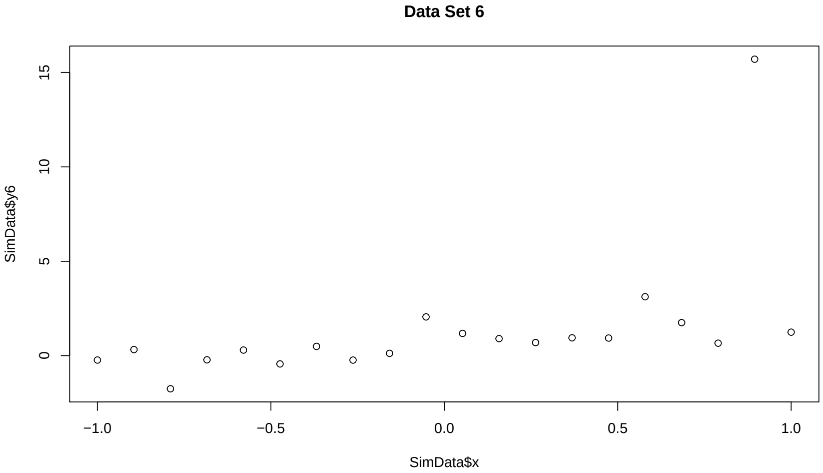
Comments?

Exercise: Data set 5



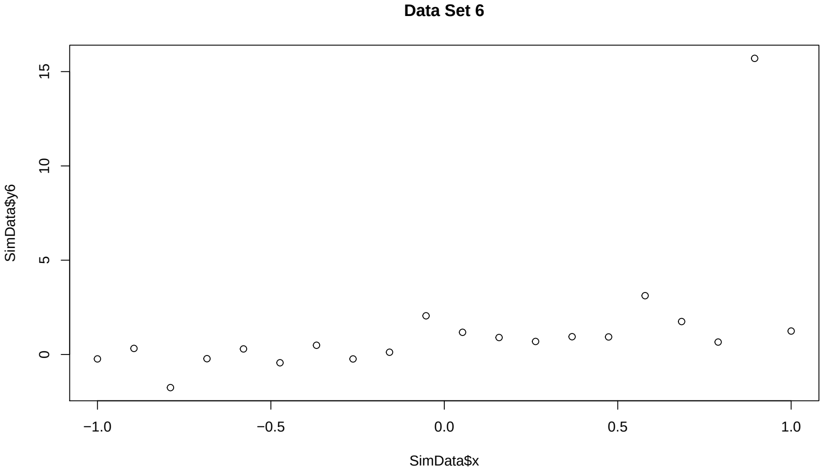
This looks OK, but as we move to the right the variation increases

Exercise: Data set 6



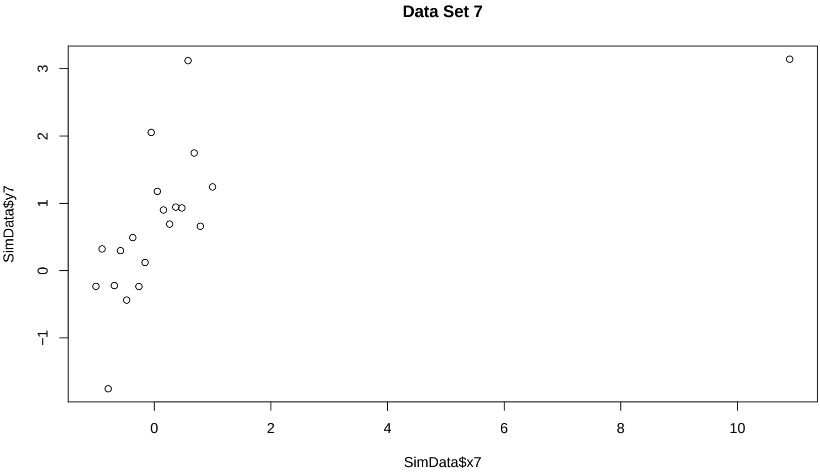
Comments?

Exercise: Data set 6



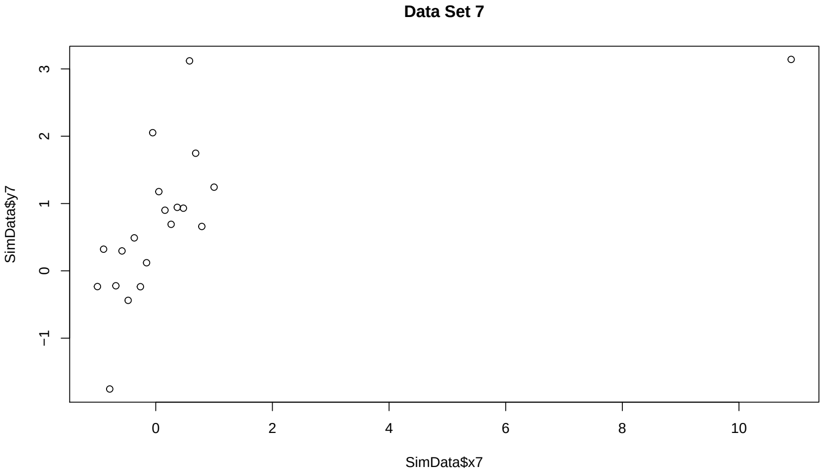
This looks OK, exept for that one point that is far too large.

Exercise: Data set 7



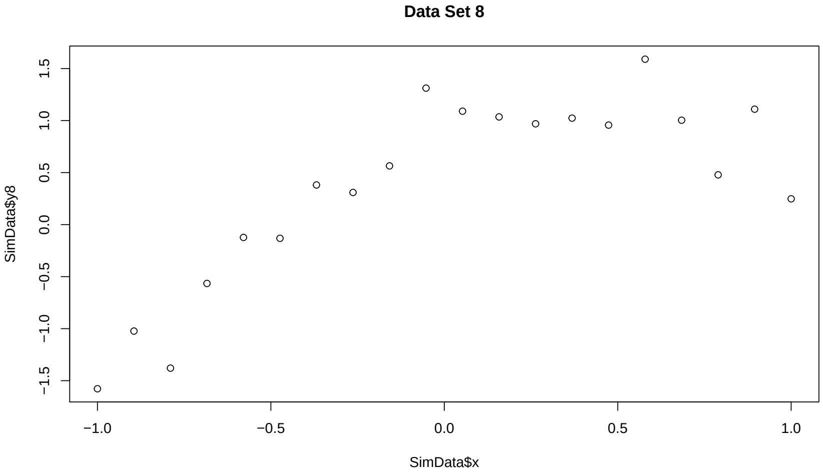
Comments?

Exercise: Data set 7



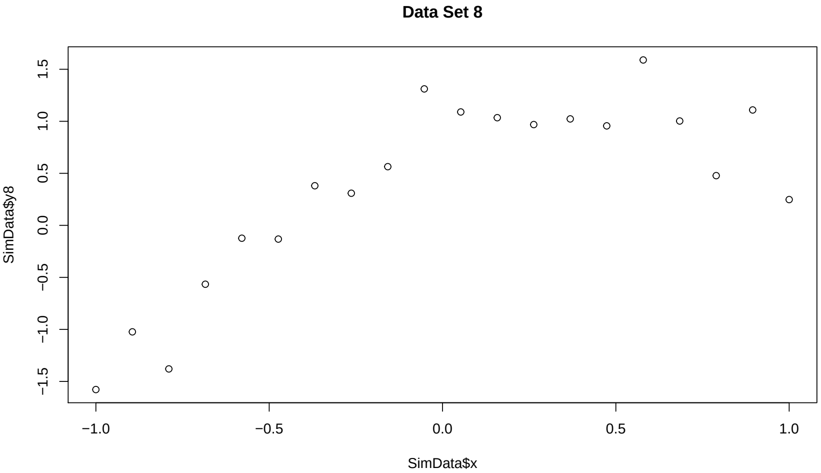
Err, what is that point doing over on the right?

Exercise: Data set 8



Comments?

Exercise: Data set 8



This looks OK, but it seems to flatten off: the amount of curvature changes

Another View of Regression

Model is systematic part + random part

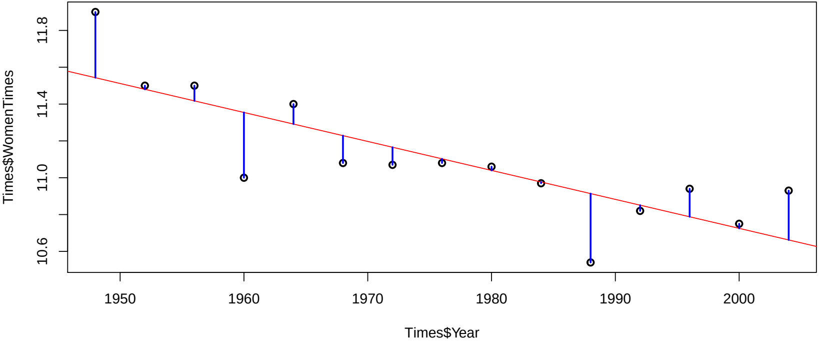
$$\begin{aligned} y_i &= \mu_i &&+ \varepsilon_i \\ &= \alpha + \beta x_i &&+ \varepsilon_i \end{aligned}$$

- Systematic part of model: a straight line
- Random part of model: residual error

All of the models we will see have this general form, but both parts can be more complicated

Women's times

```
Times <- read.csv("https://www.math.ntnu.no/emner/ST2304/20
WomenMod <- lm(WomenTimes~Year, data=Times)
plot(Times$Year, Times$WomenTimes, lwd=2)
abline(WomenMod, col=2)
segments(Times$Year, fitted(WomenMod), Times$Year, Times$Wo
```



How much variation does the model explain?

The total variation is

$$\begin{aligned}\mathrm{Var}(y_i) &= \mathrm{Var}(\alpha + \beta x_i) + \mathrm{Var}(\varepsilon_i) \\ &= \beta^2 \mathrm{Var}(x_i) + \sigma^2\end{aligned}$$

- σ^2 is the residual variation

So we can ask how much of the total total variation is explained by the model

- if it only explains 4% then the model is not good

A poor model might be because it is wrong, or because the data come from a problem that is just too noisy

The Proportion of variance explained: R^2?

We can calculate the proportion of the total variation explained by the model

$$R^2 = \frac{\text{Variance Explained}}{\text{Total Variance}} = 1 - \frac{\text{Residual Variance}}{\text{Total Variance}}$$

After a bit of maths, we get

$$R^2 = 1 - \frac{\sum(y_i - \mu_i)^2}{\sum(y_i - \bar{y})^2}$$

- $\sum(y_i - \mu_i)^2$ is the residual variance
  - squared difference from expected value
- $\sum(y_i - \bar{y})^2$ is the total variance
  - squared difference from grand mean

How do we calculate R^2 in R?

R calculates R^2 in a summary, so we can get it from this

```
R2 <- summary(WomenMod)$r.squared
R2
```

```
## [1] 0.6723703
```

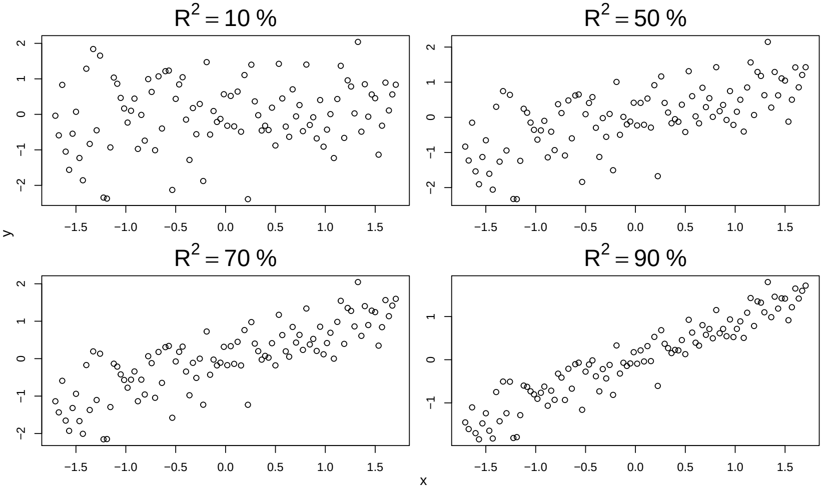
```
round(100*R2, 1)
```

```
## [1] 67.2
```

- we usually write R^2 as a percentage

What is a good R^2?

It depends!



Exercise

Exercise: calculate the R^2 for the 8 plots

You will need to read in the data, and fit the models.

x is the same for all y's *except* y7

```
Data <- read.csv("https://www.math.ntnu.no/emner/ST2304/201
mod1 <- lm(y1 ~ x, data=Data)
mod7 <- lm(y7 ~ x7, data=Data)

summary(lm(y1 ~ x, data=Data))$r.squared
```

```
## [1] 0.8708701
```

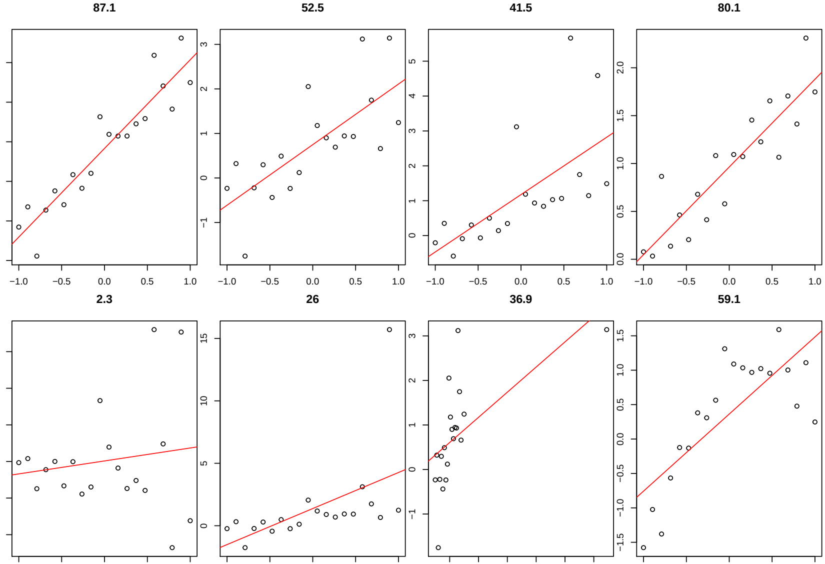
Exercise Solutions

```
Models <- list(mod1 = lm(y1 ~ x, data=Data),
               mod2 = lm(y2 ~ x, data=Data),
               mod3 = lm(y3 ~ x, data=Data),
               mod4 = lm(y4 ~ x, data=Data),
               mod5 = lm(y5 ~ x, data=Data),
               mod6 = lm(y6 ~ x, data=Data),
               mod7 = lm(y7 ~ x7, data=Data),
               mod8 = lm(y8 ~ x, data=Data))

(Rsq <- round(100*unlist(lapply(Models, function(mod) summa
```

```
## mod1 mod2 mod3 mod4 mod5 mod6 mod7 mod8
## 87.1 52.5 41.5 80.1  2.3 26.0 36.9 59.1
```

Exercise Solutions



Regression Assumptions

Model is systematic part + random part

$$\begin{aligned} y_i &= \mu_i &&+\varepsilon_i \\ &= \alpha + \beta x_i &&+\varepsilon_i \end{aligned}$$

- straight line
- errors are independent
- errors have the same variance
- errors are normally distributed
- errors have zero mean

How can these be wrong? (zero mean is forced by the maximum likelihood)

How can we check these?

This will get more complicated later

We need some tools!

Residuals

The model is

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

We can mimic this with the fitted model

$$y_i = \hat{\alpha} + \hat{\beta} x_i + e_i$$

e_i are the ***residuals***

$\hat{\alpha}$ and $\hat{\beta}$ are the parameter estimates: $\hat{\alpha} + \hat{\beta} x_i$ is the prediction for y_i

Residuals

Residuals are estimates of the error

- they should have no structure
- they should be normally distributed

We often use standardised residuals

We also sometimes standardise them:

$$t_i = \frac{r_i}{\sqrt{var(r_i)}}$$

Residuals and Fitted Values

We can extract them in R like this:

```
Women.res <- residuals(WomenMod)
round(Women.res, 2)[1:5]
```

```
##     1     2     3     4     5
##  0.36  0.02  0.08 -0.35  0.11
```

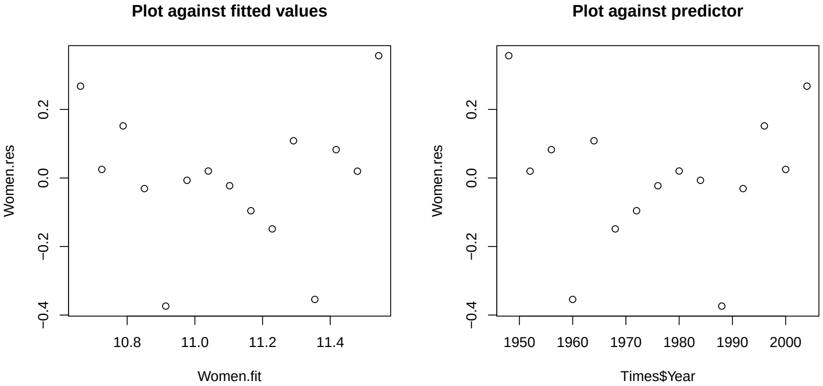
```
Women.fit <- fitted(WomenMod)
round(Women.fit, 2)[1:5]
```

```
##     1     2     3     4     5
## 11.54 11.48 11.42 11.35 11.29
```

We can stare at them, but it is more useful if we plot them

Residual plots

```
par(mfrow=c(1,2))
plot(Women.fit, Women.res, main = "Plot against fitted valu
plot(Times$Year, Women.res, main = "Plot against predictor"
```



(yes, these do look similar)

What Residual plots show

Residuals should not have any structure

With them we can see

- curvature
- outliers
- heteroscedasticity (variance changing)

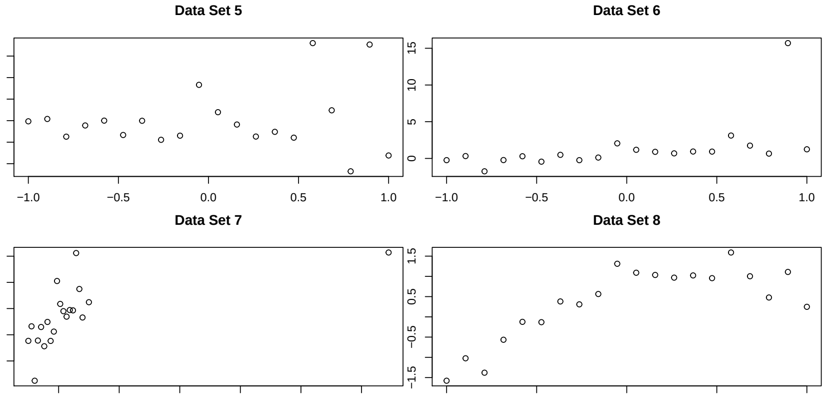
Residual Exercise

Plot the residuals against the fitted values for all 8 plots.

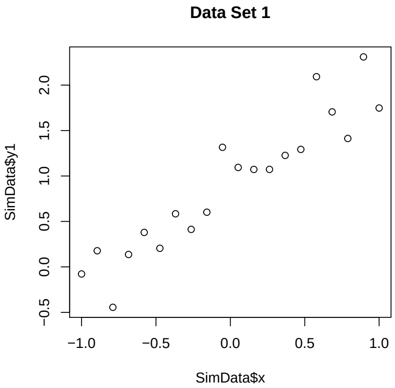
- For which data do they suggest a problem?
- What is the problem?
- Can you think of ways to improve these models?
  - no, you haven't been given the tools yet! So you can be creative

Exercise

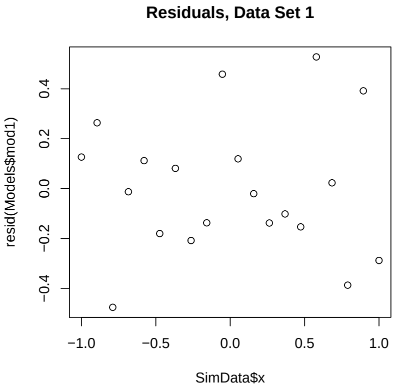
For the data sets, which assumptions are wrong?



Exercise: Data set 1

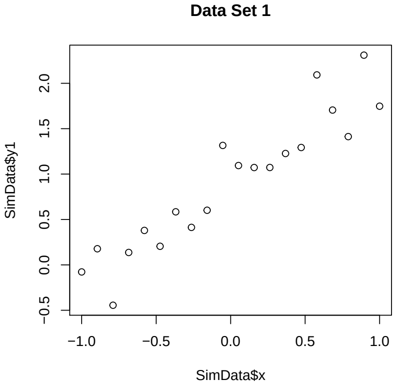


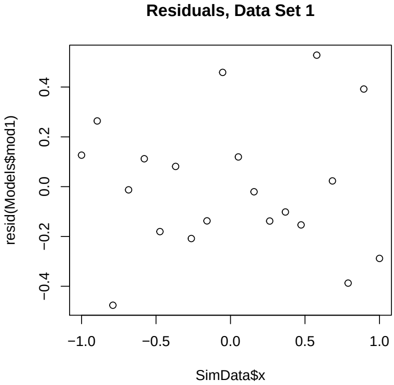
Data Set 1



Residuals, Data Set 1

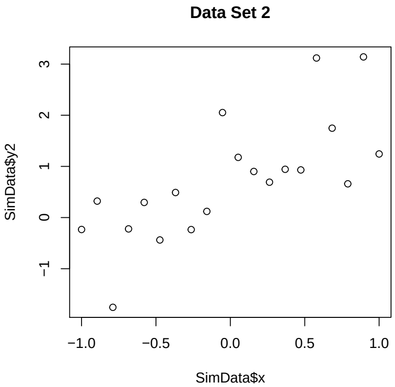
Exercise: Data set 1

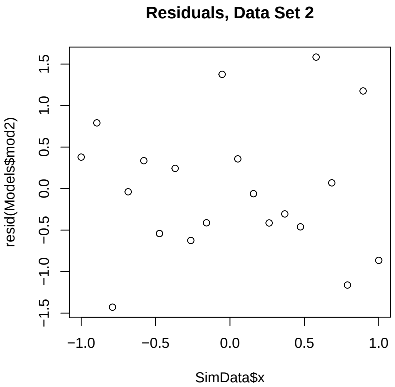




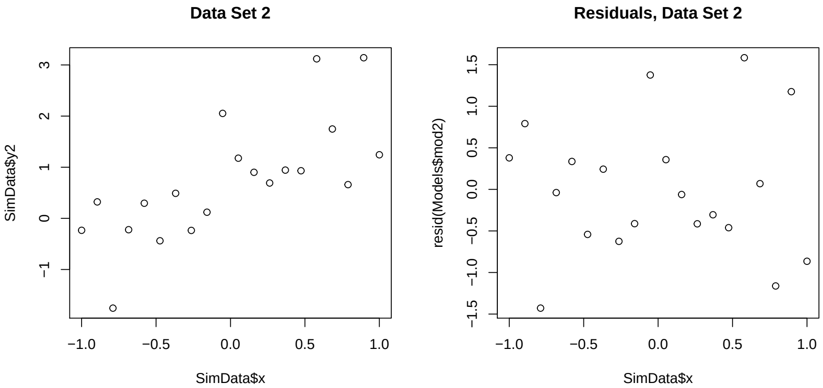
The residuals look OK

Exercise: Data set 2



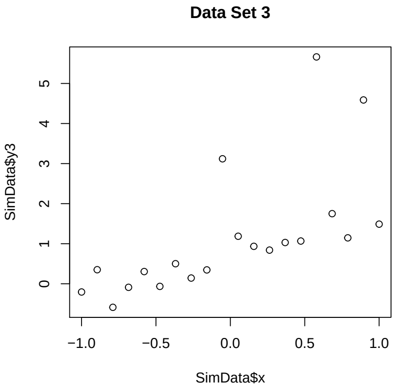


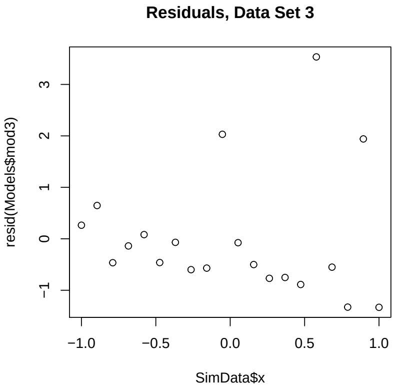
Exercise: Data set 2



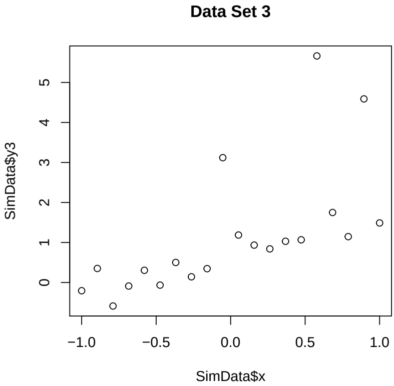
The residuals look OK

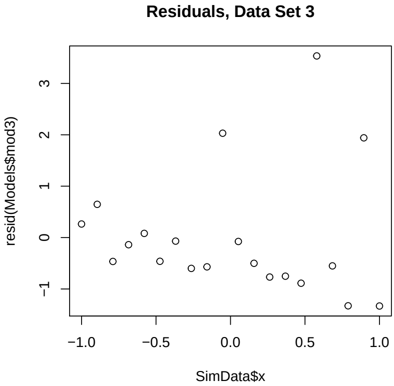
Exercise: Data set 3


Data Set 3


Residuals, Data Set 3

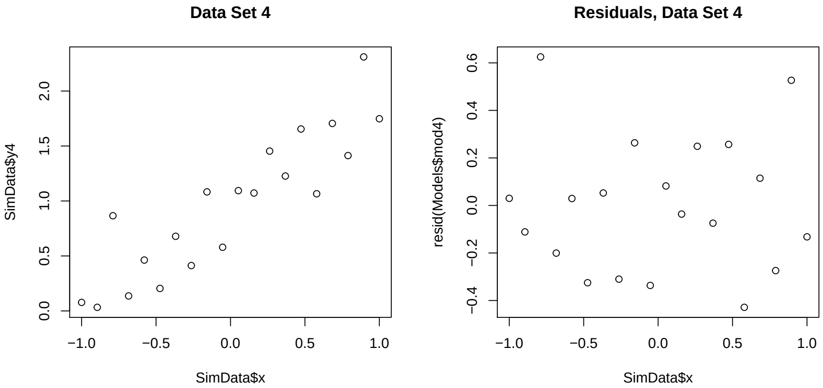
Exercise: Data set 3



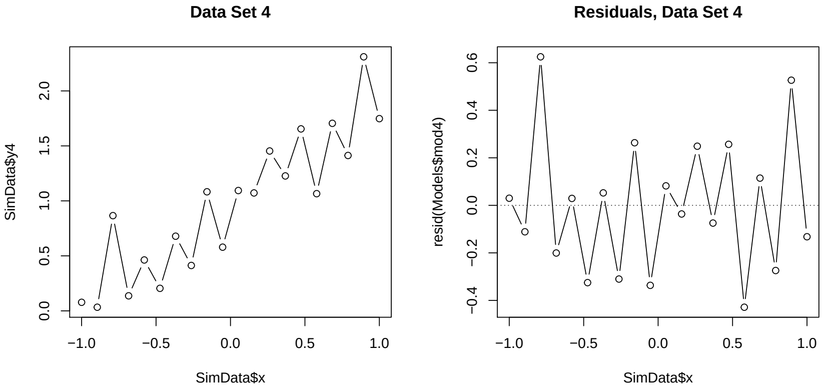


The constant variance and normality assumptions are wrong: normality might be difficult to see if you don't know what to expect.

Exercise: Data set 4



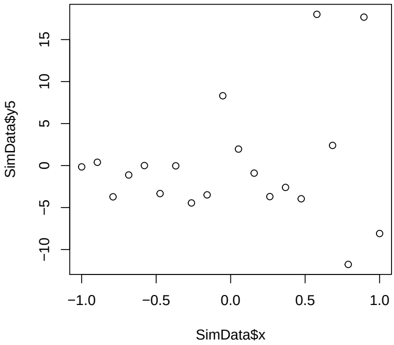
Exercise: Data set 4



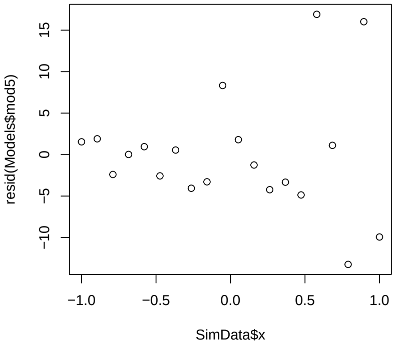
This looks OK, but the independence of errors is wrong. The signs of the residuals reverse

- I'll be surprised if anyone noticed that

Exercise: Data set 5

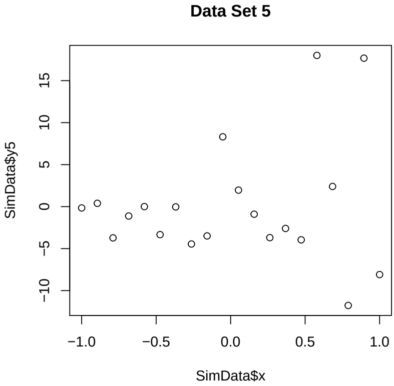


Data Set 5

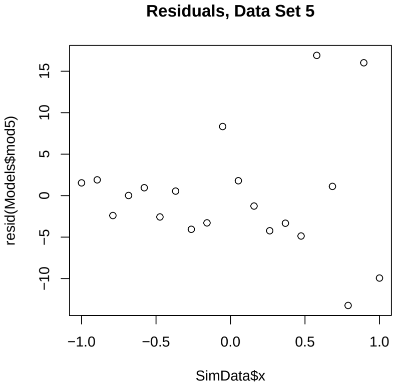


Residuals, Data Set 5

Exercise: Data set 5



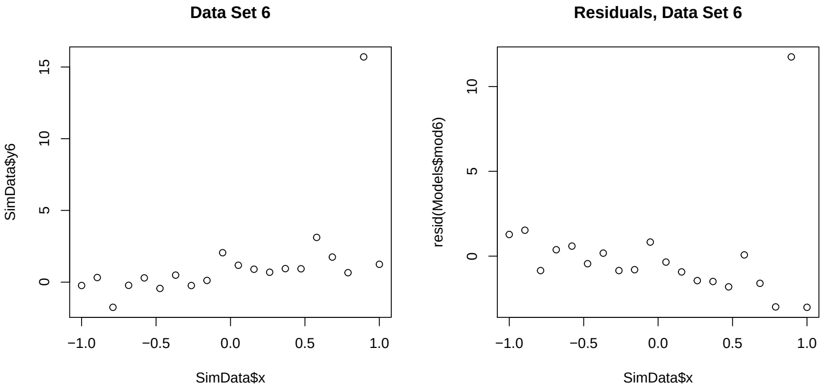
Data Set 5



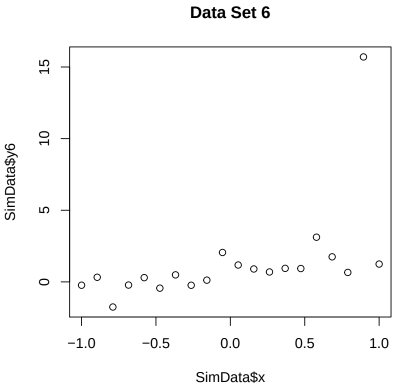
Residuals, Data Set 5

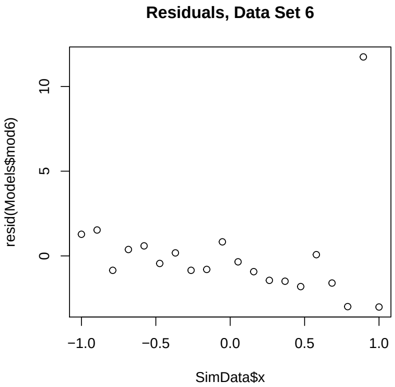
The constant variance assumption looks wrong

Exercise: Data set 6



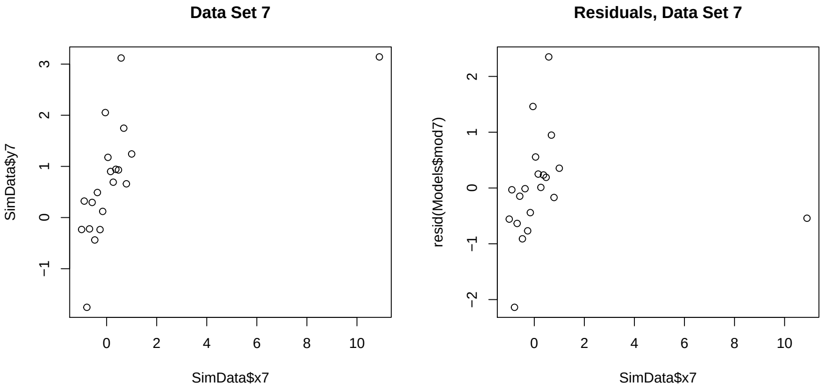
Exercise: Data set 6



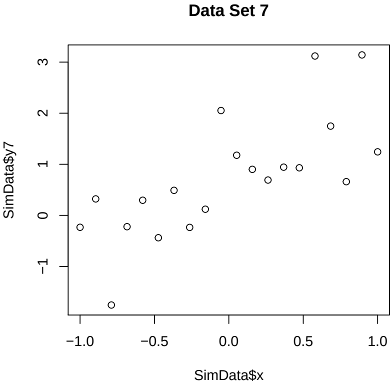


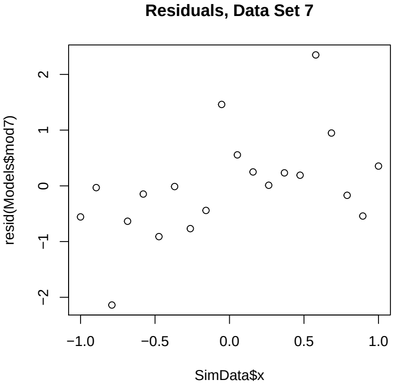
Either the normality assumption is wrong, or the constant variance. Take your pick

Exercise: Data set 7



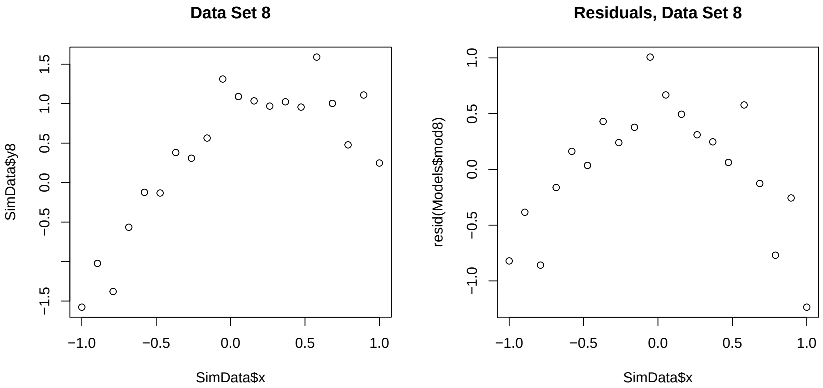
Exercise: Data set 7



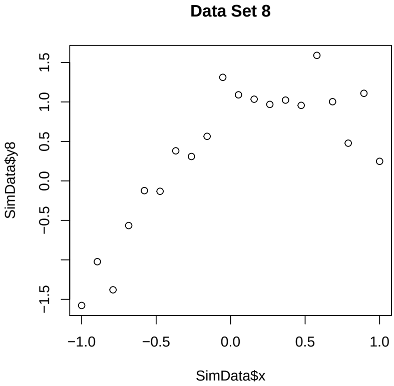


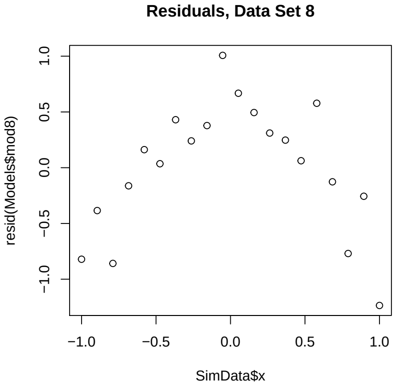
Either the normality assumption is wrong, or linearity is wrong.
Take your pick

Exercise: Data set 8



Exercise: Data set 8



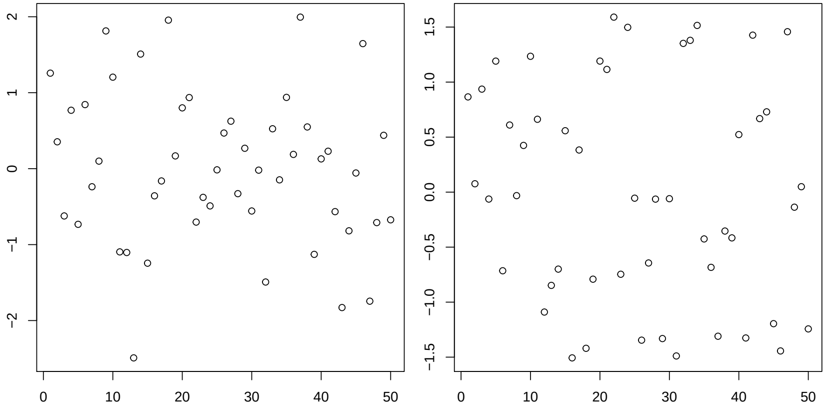


The straight line assumption is wrong. Horribly wrong

Normal Probability Plots

Residual plots can show some deviant patterns

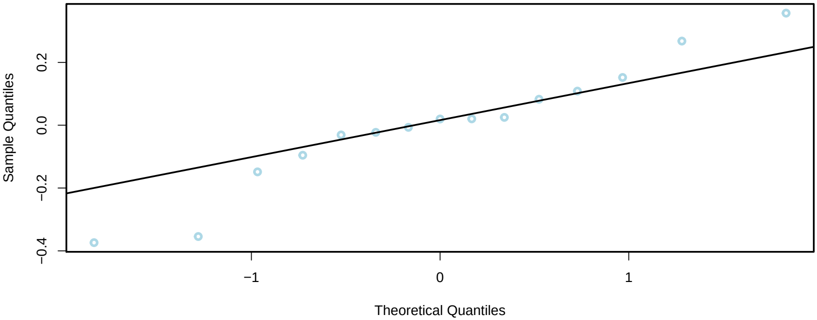
But they are poor as a test of normality



Normal Probability Plots

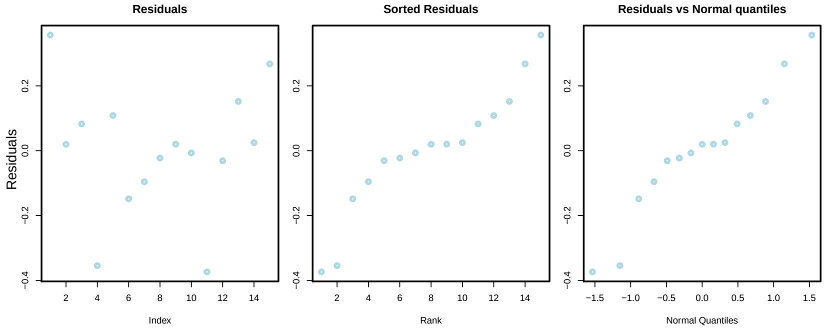
If we sort the data (smallest to largest), we can plot them against their expected values, i.e. plot r_i against the normal quantile

```r
par(mar=c(4.1,4.1,1,1), lwd=2)
qqnorm(resid(WomenMod), main="", lwd=3, col="lightblue")
qqline(resid(WomenMod))
```

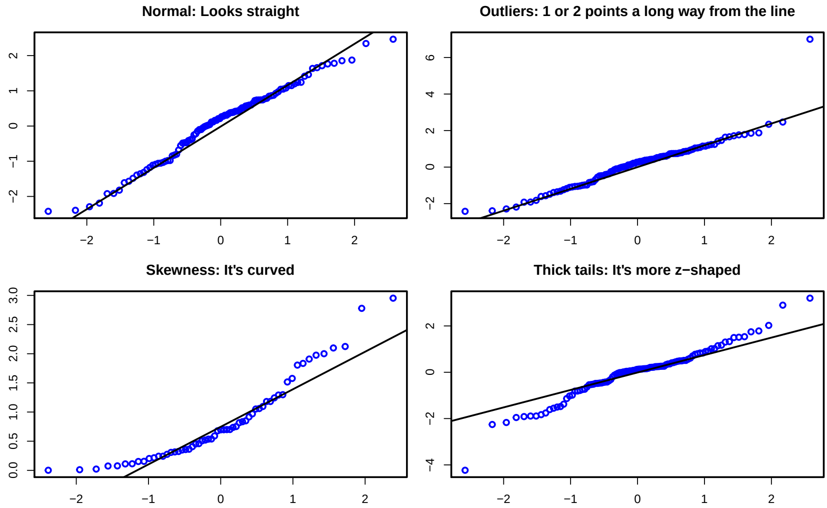


Constructing Probability Plots

```
NormQuants <- qnorm(1:length(Women.res)/
                      (1+length(Women.res)))
par(mfrow=c(1,3), mar=c(4.1,2.1,3,1), oma=c(0,2,0,0), lwd=2
plot(Women.res, lwd=3, col="lightblue", main="Residuals", y
plot(sort(Women.res), lwd=3, col="lightblue", main="Sorted
plot(NormQuants, sort(Women.res), lwd=3, col="lightblue", m
mtext("Residuals", 2, outer=TRUE)
```



What you can see



Normal: Looks straight

Outliers: 1 or 2 points a long way from the line

Skewness: It's curved

Thick tails: It's more z−shaped

You Turn. . .

- Draw normal probability plots for the 8 data sets. Do any suggest problems?
- Try to draw normal probability plots that are normal, and then have outliers, skewness and thick tails
  - you will need to simulate data (e.g. with `rnorm()`), and then add points, or transform the data

Thursday

Yesterday covered R^2, residuals vs fitted plots, and began normal probability

Today:

- we will look at normal probability a bit more
- making bad data
- leverage
- Box-Cox and transformations

Recap: R^2

A measure of how much variance is explained by our model

Recap: Residual vs fitted plots

Plots our residuals against our fitted values (residuals on y-axis and fitted on x-axis)

Checks: **that residuals meet linearity and constant variance assumptions, also check outliers.**

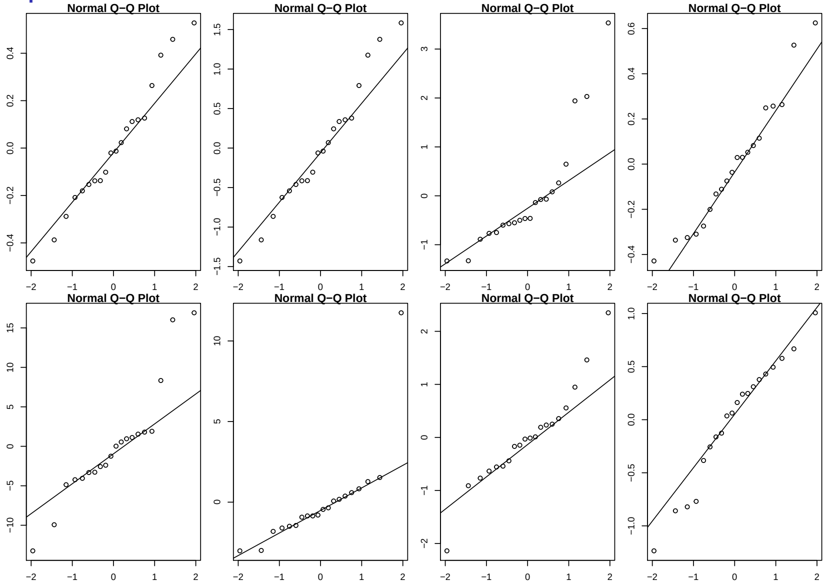
Recap: normal probability (normall QQ)

Normal QQ = normal quantile-quantile

Plots the quantiles of your residuals against a theoretical normal

Checks: **that residuals meet the normality assumption**

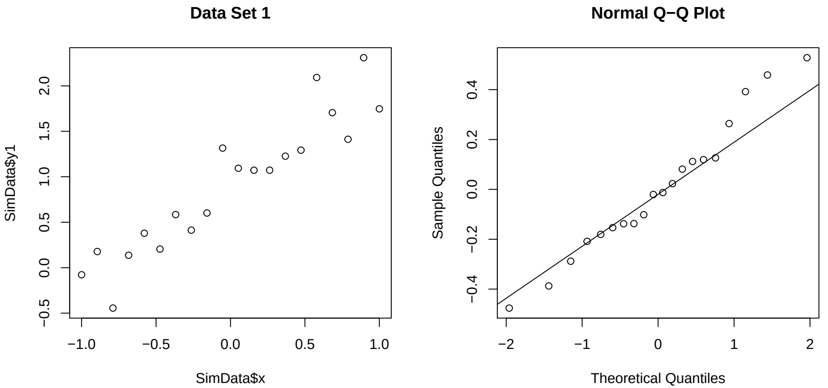
The plots



```
## $mod1
## NULL
```

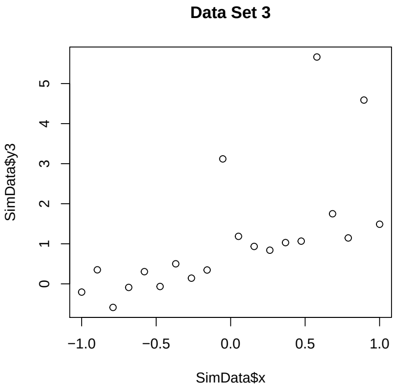
Exercise: Data set 1

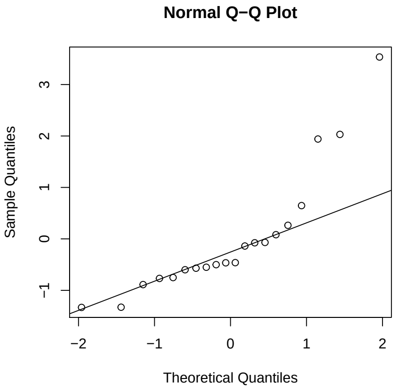
A good example



The residuals look OK

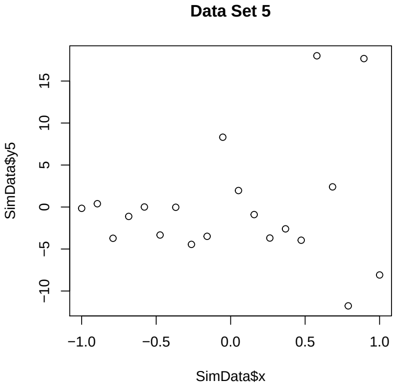
Exercise: Data set 3

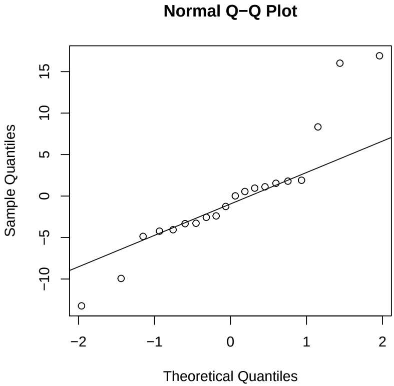




This looks skewed: the plot curves up

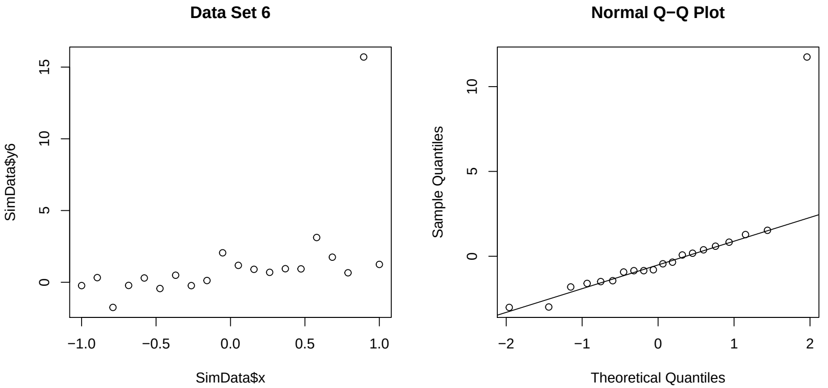
Exercise: Data set 5





The heteroscedasticity (variance is not even) makes outliers at both ends, so the tails look thick

Exercise: Data set 6



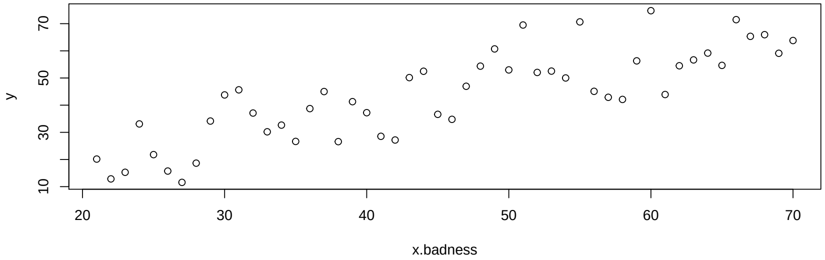
Look! Huge outlier!

Exercise: make your own bad data

There are lots of ways to do this! Let's start with some "good" data

`length()` tells you how long an object is.

```r
x.badness <- 21:70
y <- rnorm(length(x.badness), x.badness, 10)
plot(x.badness,y)
```



Exercise: make your own bad data, outliers

This is easy: add a “bad” point

Remember indexing with [] (some notes in this week’s exercise)

Code in module.

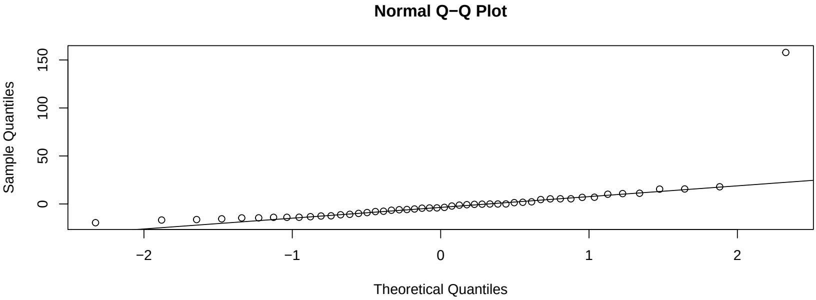
```
y.outlier <- y
y.outlier[20] <- 200
mod.outlier <- lm(y.outlier~x.badness)
#qqnorm(resid(mod.outlier))
#qqline(resid(mod.outlier))
```

Exercise: make your own bad data, outliers

This is easy: add a “bad” point

Remember indexing with [] (some notes in this week’s exercise)

```
y.outlier <- y
y.outlier[20] <- 200
mod.outlier <- lm(y.outlier~x.badness)
qqnorm(resid(mod.outlier))
qqline(resid(mod.outlier))
```

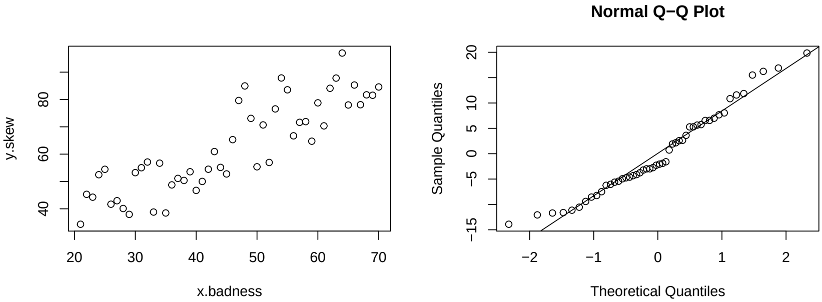


Exercise: make your own bad data, skewness

```r
err.skew <- rnorm(length(x.badness), 4, 1)^2
y.skew <- x.badness + err.skew
mod.skew <- lm(y.skew~x.badness)
#par(mfrow=c(1,2))
#plot(x.badness, y.skew)
#qqnorm(resid(mod.skew))
#qqline(resid(mod.skew))
```

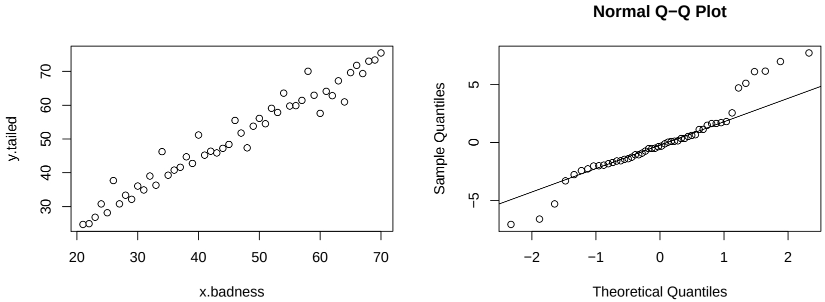
Exercise: make your own bad data, skewness

```
err.skew <- rnorm(length(x.badness), 4, 1)^2
y.skew <- x.badness + err.skew
mod.skew <- lm(y.skew~x.badness)
par(mfrow=c(1,2))
plot(x.badness, y.skew)
qqnorm(resid(mod.skew))
qqline(resid(mod.skew))
```



Exercise: make your own bad data, thick tails

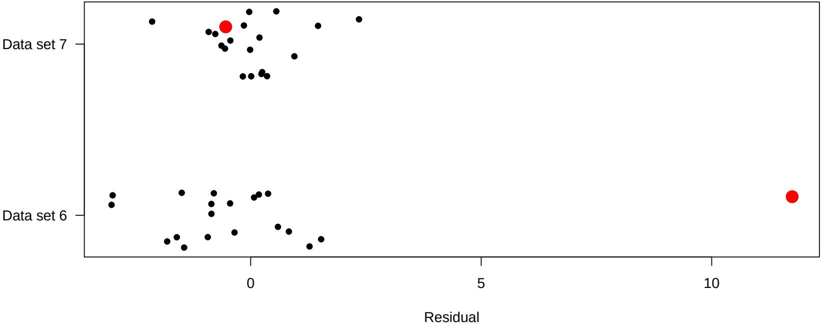
```r
err.tailed <- rnorm(length(x.badness), 4, c(1,5)) # repeat
y.tailed <- x.badness + err.tailed
mod.tailed <- lm(y.tailed~x.badness)
par(mfrow=c(1,2))
plot(x.badness, y.tailed)
qqnorm(resid(mod.tailed)); qqline(resid(mod.tailed))
```



Leverage

This is less well know, but can be a problem.

Let's look at the residuals for data sets 6 & 7:



In data set 7 there is an obvious weird point, but the residuals don't see it

Influence and Leverage: Cook's D

The general problem is with points that have a big influence on the regression. We call this **leverage**: like a good lever, these points can shift the regression line a long way.

Influence and Leverage: Cook's D

We can generalise this idea by asking how much the fitted values for the other points change if we remove a data point

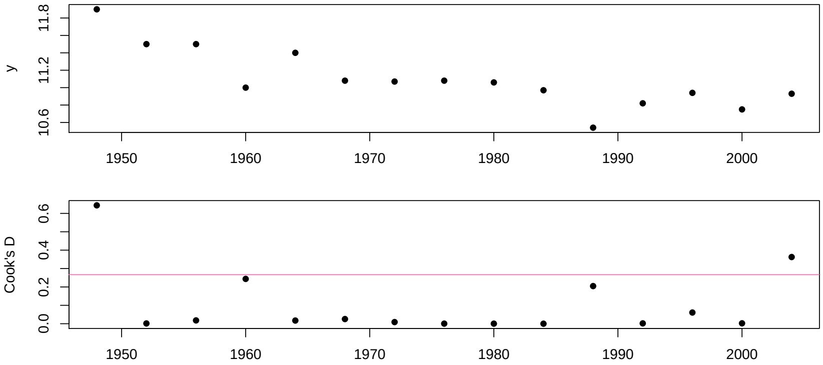
$$D_i = \frac{\sum_{j=1}^{n} (\hat{y}_j - \hat{y}_{j(-i)})^2}{s^2}$$

- $\hat{y}_j$ - prediction for full model
- $\hat{y}_{j(-i)}$ - prediction for model with data point i removed
- s^2 - residual variance
- for each data point take the difference in the predicted value for that point between the full model, and the model with that point removed
- sum the squares, and standardise by the residual variance

What is influential?

Large values of D_i mean a large influence

- $D_i > 1$, or $4/n$



Influence

Your task (in module)

Fit the model with and without the weird point

You can remove the point like this:

```
DataNotWeird <- Data[Data$x7<10,]
```

Look at the fitted models. How similar are they?

- check the parameter estimates
- plot the fitted lines on the data (with `abline()`)

Calculate Cook's D for the different data sets, and plot them against `x`. Do you see any influential points?

Look at the fitted models. How similar are they?

Influence: How similar are the parameter estimates

```
DataNotWeird <- Data[Data$x7<10,]
mod7 <- lm(y7 ~ x7, data=Data)
mod7NW <- lm(y7 ~ x7, data=DataNotWeird)
coef(mod7)
```

```
## (Intercept)          x7
##    0.606328    0.282340
```

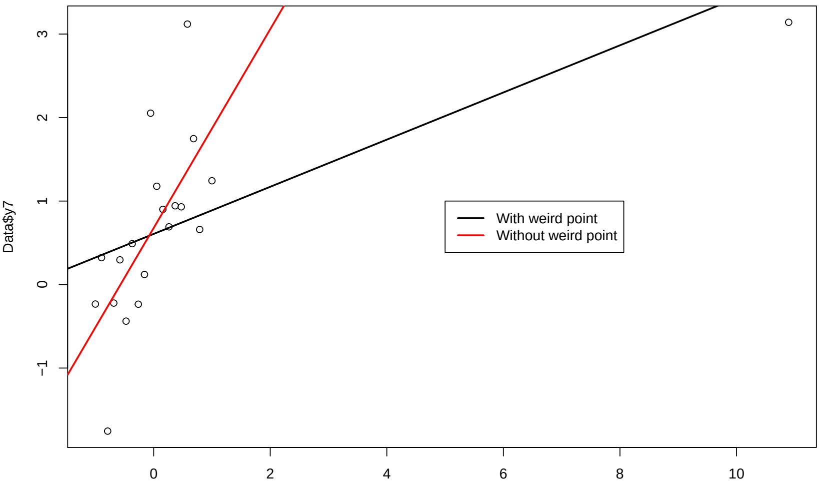
```
coef(mod7NW)
```

```
## (Intercept)          x7
##   0.6776091   1.1905997
```

We can see that removing the odd point changfes the slope (`x7`) by a huge amount, from 0.28 to 1.19.

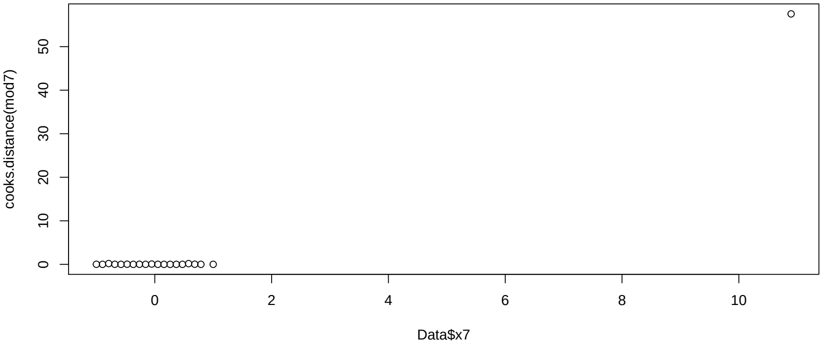
Influence: plots with abline())

```
plot(Data$x7, Data$y7)
abline(mod7, lwd=2)
abline(mod7NW, col=2, lwd=2)
legend(5,1, c("With weird point", "Without weird point"), c
```



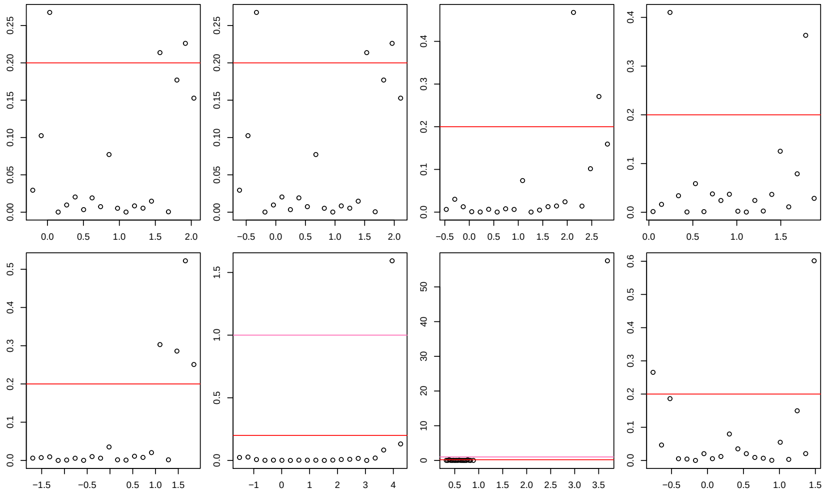
Influence: Cook's D

```
plot(Data$x7, cooks.distance(mod7))
```



We can see the hugely influential point, with a value of 57.5. For the other data...

Thing



There are a few influentiual points, largely ouliers.

How good is my model? A Summary

- Model as fit + residuals
- R^2: How much variation does the model explain?
- Residual plots
  - curvature
  - outliers
  - heteroscedasticity
- Normal Probability Plots
- Influential Points (Cook's D)

How can we improve the model?

First, check the data and model for silly mistakes

- typos are common

Then, ask if if any misfit is a problem

- does it change the conclusions?
- will it change predictions?

Individual Data Points

Is your data point wrong?

- typos?
- real but unique

If it is wrong, correct, if it is right, might want to remove it & see if that makes a big difference

- if it does, be careful!

Possible Solutions

Transform the covariate

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

e.g. $\sqrt(x_i)$, x_i^2, $\log(x_i)$,

Add more terms

- quadratic

$$y_i = \alpha + \beta x_i + \gamma x_i^2 + \varepsilon_i$$

More about this later

Transformations

Transform the response

e.g. $\sqrt(x_i)$, x_i^2, $\log(x_i)$

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

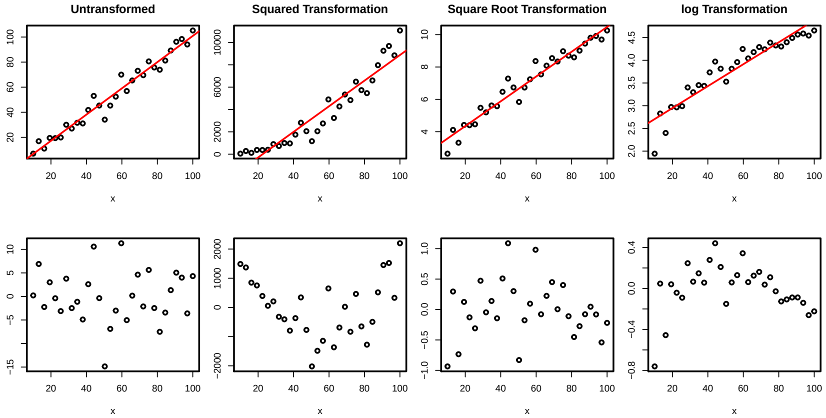
Box-Cox transformations

General Class of transformations

$$y_i \rightarrow y_i^p$$

if $p = 0$, use $log(y_i)$

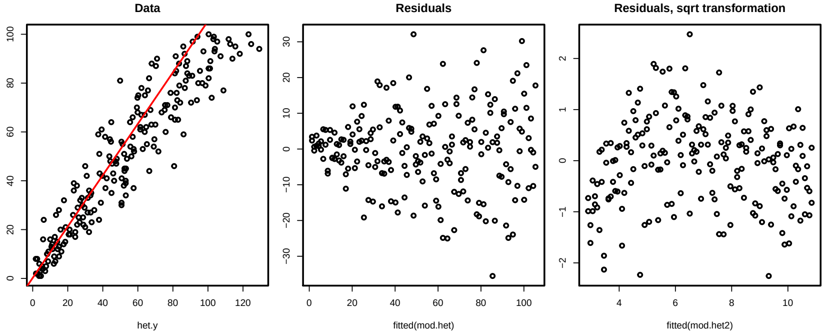
Using Box-Cox transformations



Heteroscedasticity

Variance changes with the mean

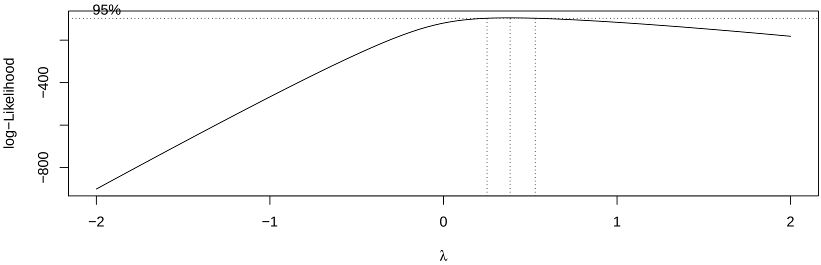
- Box-Cox can also solve this (or make it worse)



Box-Cox in R

R has a function to find the best Box-Cox transformation

```r
library(MASS)
x <- 1:50
y <- rnorm(50,0.1*x, 1)^2
boxcox(lm(y ~ x)) # 0.5 is true transformation
```



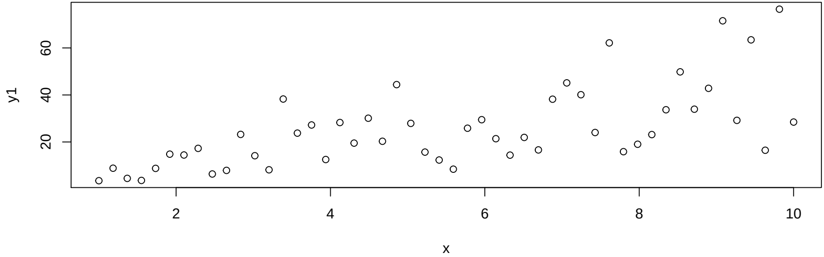
Your Turn

Follow the exercise!

Has an example in this week's exercise or in the module.

# Your Turn: y1

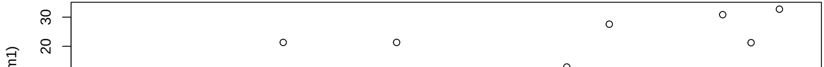
```
x <- seq(1,10, length=50)
y1 <- rgamma(length(x), shape=5, scale=x)
plot(x,y1)
```



## Your Turn: y1 residuals

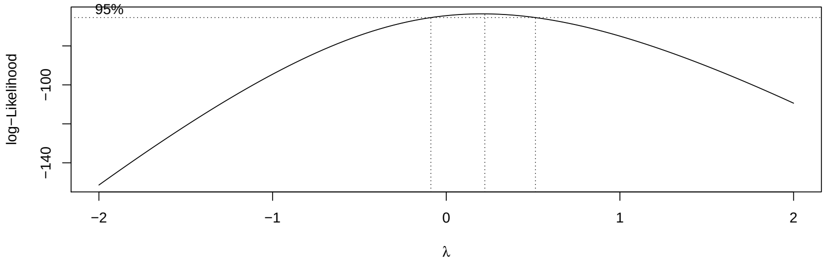
There seems to be a positive slope, but the data are heteroscedastic.

```
modsim1 <- lm(y1~x)
plot(x, resid(modsim1))
```



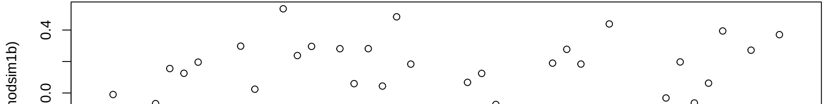
Your Turn: y1 Box-Cox

```
library(MASS)
boxcox(modsim1)
```



The Box-Cox function suggests a transformation just under 0.5, so

```
modsim1a <- lm(sqrt(y1)~x)
modsim1b <- lm(y1^0.25 ~x)
plot(x, resid(modsim1b))
```



Summary

We now know how to asses the model fit

- R^2 show how much variation the model explains
- Residual plots and Normal Probability Plots can show curvature, outliers, and varying variance
- Influential Points can be detected using Cook's D. These may not be large outliers!
- We should check outliers & other odd points - are they typos?
- We can try to transform the response to get a better model