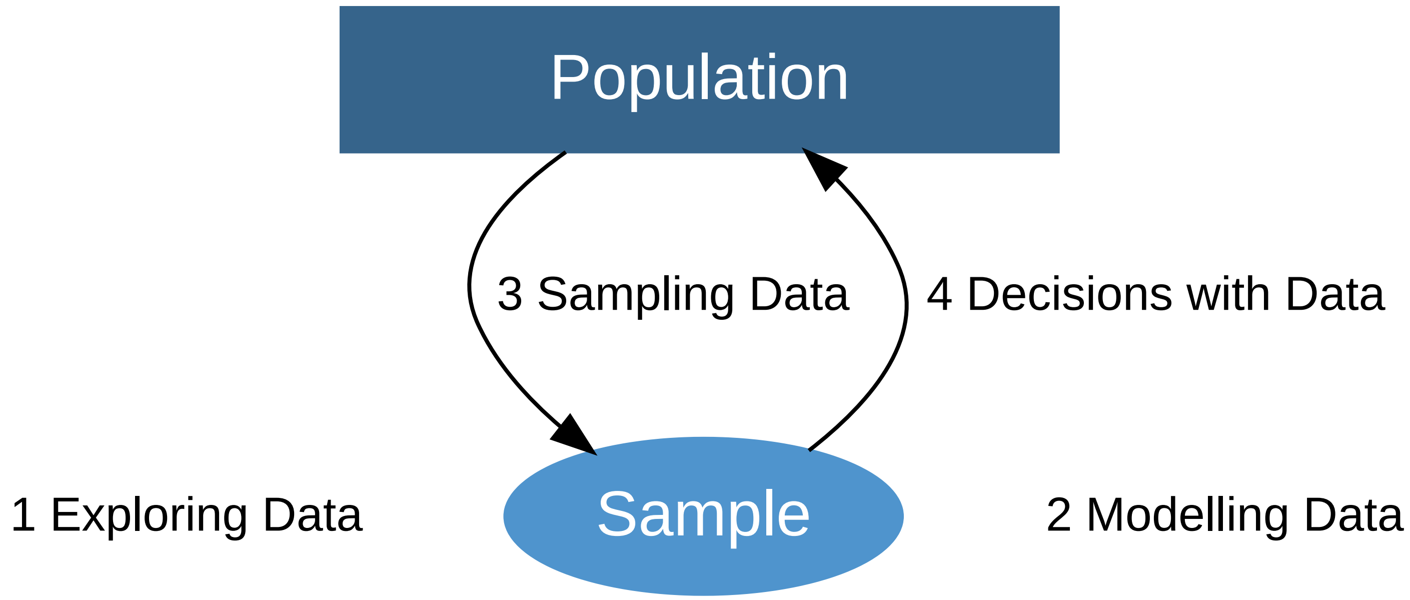
Non Linear Models

Modelling Data | Linear Model

Unit Overview



Module2 Modelling Data

Normal Model

What is the Normal Curve? How can we use it to model data?

Linear Model

How can we describe the relationship between 2 variables? When is a linear model appropriate?

Non Linear Models

Data Story | How quickly does alcohol decrease in concentration in the blood?

Linear vs Non-Linear Models

Fitting Linear Model to BAC

Fitting Non-Linear Models to BAC

Other Models

Summary

Data Story

How quickly does alcohol decrease in concentration in the blood?

Alcohol in Australia

“Alcohol has a complex role in Australian society. Most Australians drink alcohol for enjoyment, relaxation and sociability and at levels that cause few adverse effects. However, a substantial proportion of people drink at levels that increase their risk of alcohol-related harm.”

NHMRC

Blood Alcohol Concentration (BAC)

- Working out guidelines for healthy drinking is an area of ongoing research.
- A common measure is the Blood Alcohol Concentration (BAC), which is the percentage of total blood volume that is alcohol (or equivalently, grams of alcohol per litre of blood).
- For driving, the legal BAC is 0.05%, which is 0.5g/L or 0.05g/ 100ml.
- Unlike many other drugs, the rate of alcohol metabolism is roughly constant (a zero-order reaction in Chemistry), suggesting a linear model.
- The rate of alcohol metabolism is usually independent of the BAC because typical levels of alcohol consumption saturate the metabolising capacity of enzymes within the liver. It also varies between individuals, depending on factors such as age and gender.

🔗 ADF

Data for BAC

The data in `AlcoholConcentration.xlsx` is blood alcohol concentration over time, after ethanol administration in the fasting state.

🔗 Paper

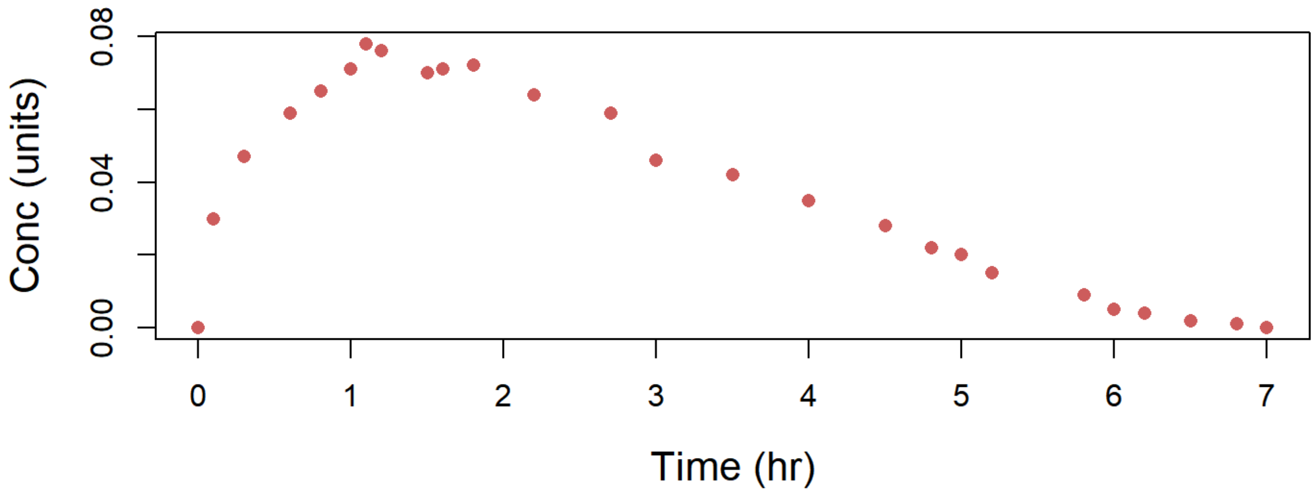
```r
library(readxl)
data = read_excel("data/AlcoholConcentration.xlsx")
head(data)
```

```
## # A tibble: 6 x 2
##   TimeHours BloodAlcoholConcentration
##       <dbl>                     <dbl>
## 1       0                      0.0001
## 2       0.1                    0.03
## 3       0.3                    0.047
## 4       0.6                    0.059
## 5       0.8                    0.065
## 6       1                      0.071
```

```r
dim(data)
```

```
## [1] 26  2
```

```r
Time = data$TimeHours
Conc = data$BloodAlcoholConcentration
plot(Time, Conc, xlab = "Time (hr)", ylab = "Conc (units)", pch = 16, cex.lab = 1.3,
    col = "indianred")
```



Would a linear model be a good fit?

Linear vs Non-Linear Models

Linear & Non-Linear models

Linear model

A (statistical) linear model takes the form

$$y = a + bx_1 + cx_2 + \ldots$$

where $a, b, c \ldots$ are constants. The **simple** linear model is $y = a + bx$.

Non-Linear model

A non-linear model cannot be written as a linear model!

Linear Model

Advantages:

- Easy to fit
- Results in a simple equation (analytical solution)
- If we increase the number of variables (x_1, x_2) or degree (eg x^2, x^3) we can fit many shapes.

Disadvantage:

- Hard to link parameters (slope, intercept) to mechanistic processes, for example in a biological or environmental context.

Non-linear models

Advantages:

- Most are linked to mechanistic processes, resulting in interpretable parameters.
    - The parameters of a nonlinear model may be directly related to processes under study, such as the carrying capacity or maximum rate of growth.
- The constraints (such as asymptotes) can easily be built into a nonlinear model.

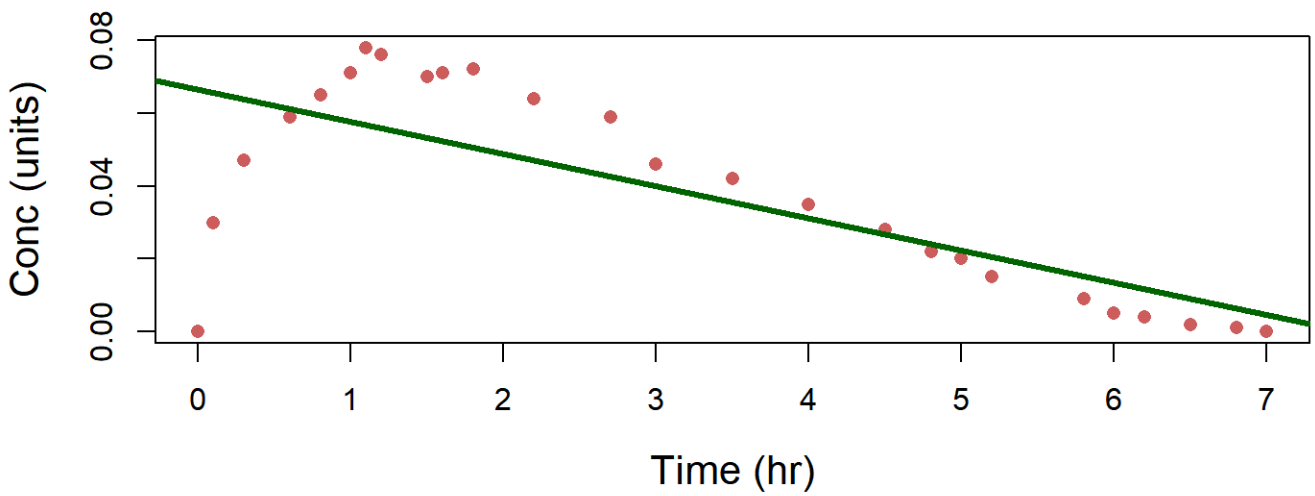
Disadvantage?

- Numerical solutions are often needed, but this is now straight-forward on a computer.

Fitting Linear Model to BAC

Model1: Simple Linear Model (straight line)

```
plot(Time, Conc, xlab = "Time (hr)", ylab = "Conc (units)", pch = 16, cex.lab = 1.3,
    col = "indianred")
abline(lm(Conc ~ Time), col = "darkgreen", lwd = 3)
```



What are the pros and cons of this fit? What does it describe well, and poorly?

```r
linear.model = lm(Conc ~ Time)
summary(linear.model)
```

```
##
## Call:
## lm(formula = Conc ~ Time)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.066294 -0.005978 -0.000313  0.015705  0.021508
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.066394   0.006545  10.144 3.71e-10 ***
## Time        -0.008835   0.001662  -5.315 1.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01945 on 24 degrees of freedom
## Multiple R-squared:  0.5406, Adjusted R-squared:  0.5215
## F-statistic: 28.25 on 1 and 24 DF,  p-value: 1.877e-05
```

What is the equation for this model? What is the correlation coefficient?

Model2: Quadratic Model

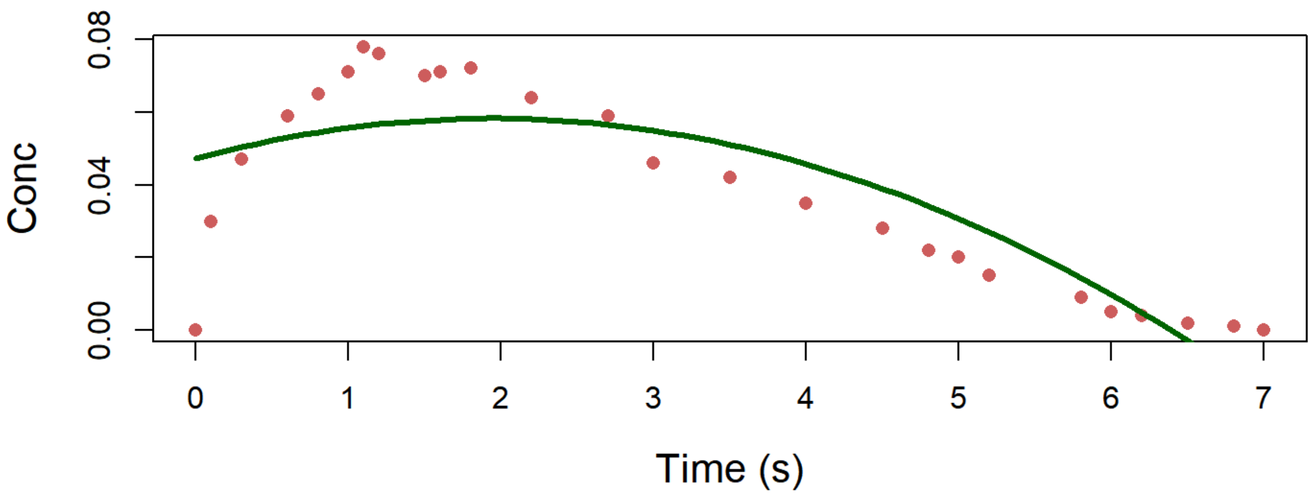
```r
# Fit Model
Time2 = Time^2  # Quadratic variable
quad.model = lm(Conc ~ Time + Time2)
summary(quad.model)
```

```
##
## Call:
## lm(formula = Conc ~ Time + Time2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.047174 -0.010294  0.000788  0.012266  0.021772
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0472736  0.0073866   6.400 1.57e-06 ***
## Time         0.0113721  0.0056123   2.026  0.05448 .
## Time2       -0.0029376  0.0007922  -3.708  0.00116 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01571 on 23 degrees of freedom
## Multiple R-squared:  0.7125, Adjusted R-squared:  0.6875
## F-statistic:  28.5 on 2 and 23 DF,  p-value: 5.943e-07
```

What is the equation for this model? What is the correlation coefficient?

```r
# Predicted Values
timevalues = seq(0, 7, 0.1)
predictedvalues = predict(quad.model, list(Time = timevalues, Time2 = timevalues^2))

# Plot of data with predicted values as curve
plot(Time, Conc, pch = 16, xlab = "Time (s)", ylab = "Conc", cex.lab = 1.3, col = "indianred")
lines(timevalues, predictedvalues, col = "darkgreen", lwd = 3)
```



What are the pros and cons of this fit?

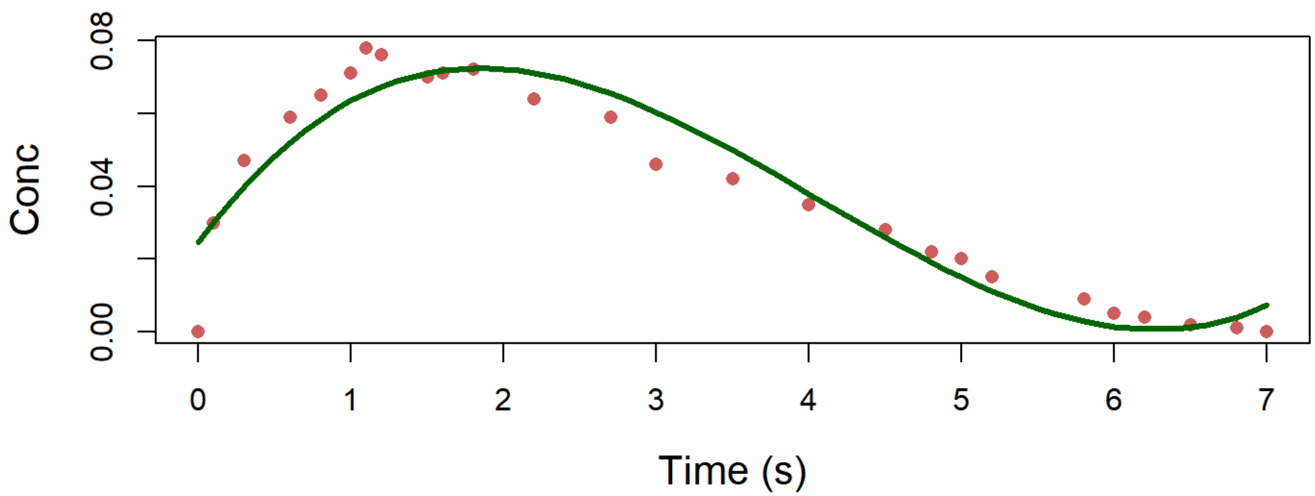
Model3: Cubic Model

```r
# Fit Model
Time3 <- Time^3  # Cubic variable
cubic.model <- lm(Conc ~ Time + Time2 + Time3)
summary(cubic.model)
```

```
##
## Call:
## lm(formula = Conc ~ Time + Time2 + Time3)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.024389 -0.003094  0.001368  0.005802  0.012343
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.024489   0.004956   4.941 6.08e-05 ***
## Time         0.057663   0.006771   8.516 2.06e-08 ***
## Time2       -0.020227   0.002306  -8.773 1.24e-08 ***
## Time3        0.001663   0.000218   7.629 1.29e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008415 on 22 degrees of freedom
## Multiple R-squared:  0.9211, Adjusted R-squared:  0.9104
## F-statistic: 85.66 on 3 and 22 DF,  p-value: 2.732e-12
```

```r
# Predicted Values
timevalues <- seq(0, 7, 0.1)
predictedvalues <- predict(cubic.model, list(Time = timevalues, Time2 = timevalues^2,
    Time3 = timevalues^3))

# Plot of data with predicted values as curve
plot(Time, Conc, pch = 16, xlab = "Time (s)", ylab = "Conc", cex.lab = 1.3, col = "indianred")
lines(timevalues, predictedvalues, col = "darkgreen", lwd = 3)
```



What are the pros and cons of this fit? What is the equation and correlation coefficient?

Fitting Non-Linear Models to BAC

Model4: Exponential Model

Step1: Find initial parameters, by fitting a linear model to $log(y)$

```
logConc <- log(Conc)
log.linear.model <- lm(logConc ~ Time)
summary(log.linear.model)
```

```
##
## Call:
## lm(formula = logConc ~ Time)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4861 -0.1718  0.6407  0.8976  1.0432
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.7243     0.5703  -4.777 7.32e-05 ***
## Time         -0.4178     0.1448  -2.885  0.00815 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.694 on 24 degrees of freedom
## Multiple R-squared:  0.2575, Adjusted R-squared:  0.2265
## F-statistic: 8.322 on 1 and 24 DF,  p-value: 0.008146
```

This gives the model: $log(y) = -2.7243 - 0.4178x$.

Rearranging: $y = e^{-2.7243-0.4178x} = e^{-2.7243}e^{-0.4178x}$.

This is an exponential $y = y_0e^{kx}$, with parameters

- $y_0 = e^{-2.7243} \approx 0.0655921$.
- $k = -0.4178$.

Step2: Fit exponential model

```
## Initial parameters
initial.mod <- c(Y0 = exp(-2.7243), k = -0.4178)
## Fit exponential model
fit.mod <- nls(Conc ~ Y0 * exp(k * Time), start = initial.mod, trace = T)
```

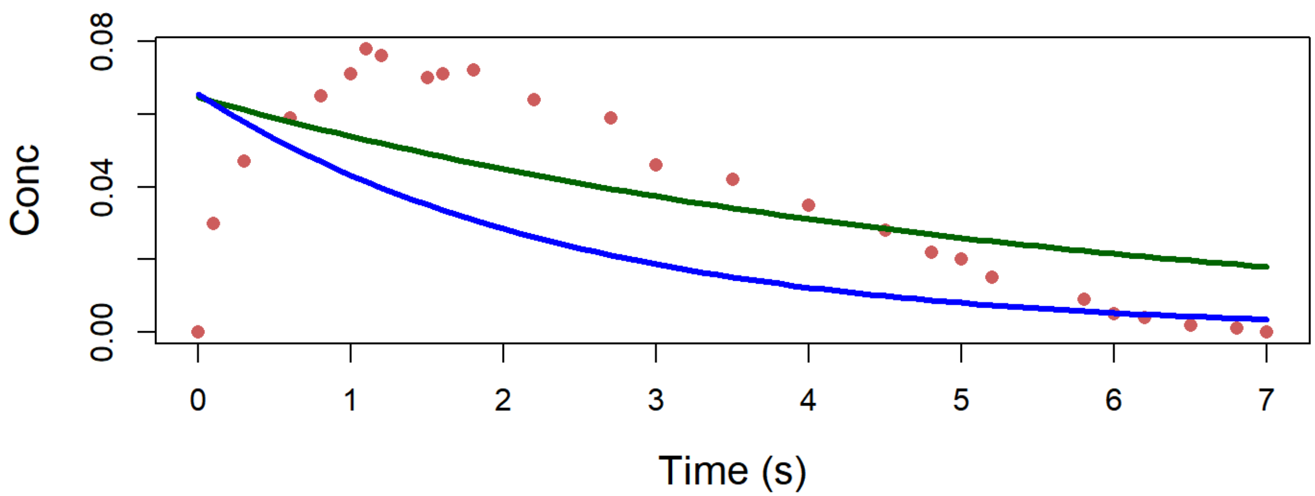
```
## 0.01917787 :   0.0655921 -0.4178000
## 0.01401737 :   0.05343120 -0.08024218
## 0.01159705 :   0.06748269 -0.20625319
## 0.01152147 :   0.06326921 -0.16917905
## 0.01149306 :   0.06529795 -0.18817310
## 0.01148797 :   0.06434151 -0.17933773
## 0.01148661 :   0.06480815 -0.18369714
## 0.01148631 :   0.06458263 -0.18159998
## 0.01148624 :   0.06469227 -0.18262207
## 0.01148622 :   0.0646391 -0.1821270
## 0.01148622 :   0.06466491 -0.18236752
## 0.01148622 :   0.06465239 -0.18225083
## 0.01148622 :   0.06465847 -0.18230748
## 0.01148622 :   0.06465552 -0.18227999
## 0.01148622 :   0.06465695 -0.18229333
## 0.01148622 :   0.06465625 -0.18228685
## 0.01148622 :   0.06465659 -0.18229000
```

```r
## Summarise model
summary(fit.mod)
```

```
##
## Formula: Conc ~ Y0 * exp(k * Time)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## Y0  0.06466    0.00919   7.036 2.83e-07 ***
## k  -0.18229    0.05966  -3.056  0.00544 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02188 on 24 degrees of freedom
##
## Number of iterations to convergence: 16
## Achieved convergence tolerance: 5.22e-06
```

```r
predictedvalues <- predict(fit.mod, list(Time = timevalues))
predictedinitial <- exp(-2.7243) * exp(timevalues * -0.4178)
```

```r
## Plot fitted model (darkgreen) and initial model(blue)
plot(Time, Conc, pch = 16, xlab = "Time (s)", ylab = "Conc", cex.lab = 1.3, col = "indianred")
lines(timevalues, predictedvalues, col = "darkgreen", lwd = 3)
lines(timevalues, predictedinitial, col = "blue", lwd = 3)
```



What are the pros and cons of this fit?

Model5: Von Richter

$$y = axe^{-bx}$$

```
## Initial parameters
non.mod <- c(a = 4, b = 0.2)

## Fit Von Richter model
nl.mod <- nls(Conc ~ a * Time * exp(-b * Time), start = non.mod, trace = T)
```

```
## 842.6317 :  4.0 0.2
## 0.2012686 :  0.07868176 0.20319545
## 0.02696781 :  0.07933915 0.36701788
## 0.002951188 :  0.1126975 0.5922925
## 0.0006955368 :  0.1487270 0.7430856
## 0.000627384 :  0.1583134 0.7649397
## 0.0006272734 :  0.1584495 0.7646349
## 0.0006272734 :  0.1584499 0.7646364
```

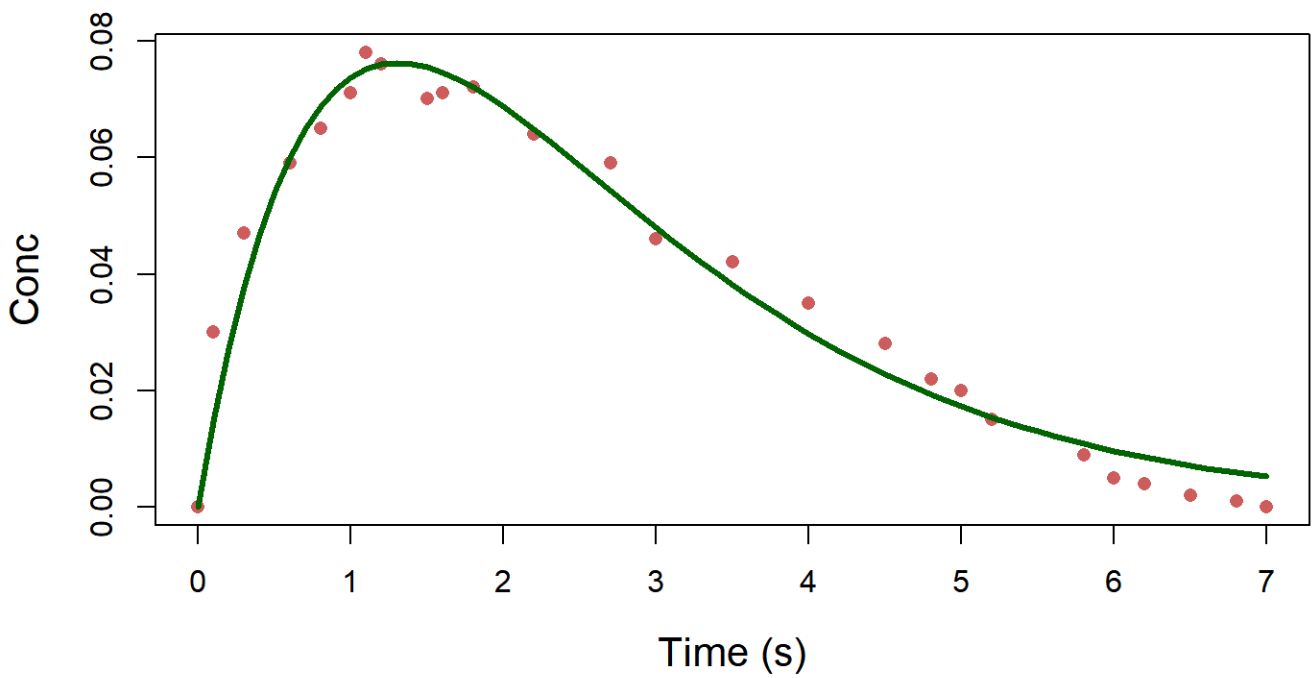
```
summary(nl.mod)
```

```
##
## Formula: Conc ~ a * Time * exp(-b * Time)
##
## Parameters:
##   Estimate Std. Error t value Pr(>|t|)
## a 0.158450   0.006539   24.23   <2e-16 ***
## b 0.764636   0.021386   35.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005112 on 24 degrees of freedom
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 4.635e-08
```

```r
timevalues <- seq(0, 7, 0.1)

predictedvalues <- predict(nl.mod, list(Time = timevalues))

plot(Time, Conc, pch = 16, xlab = "Time (s)", ylab = "Conc", cex.lab = 1.3, col = "indianred")
lines(timevalues, predictedvalues, col = "darkgreen", lwd = 3)
```



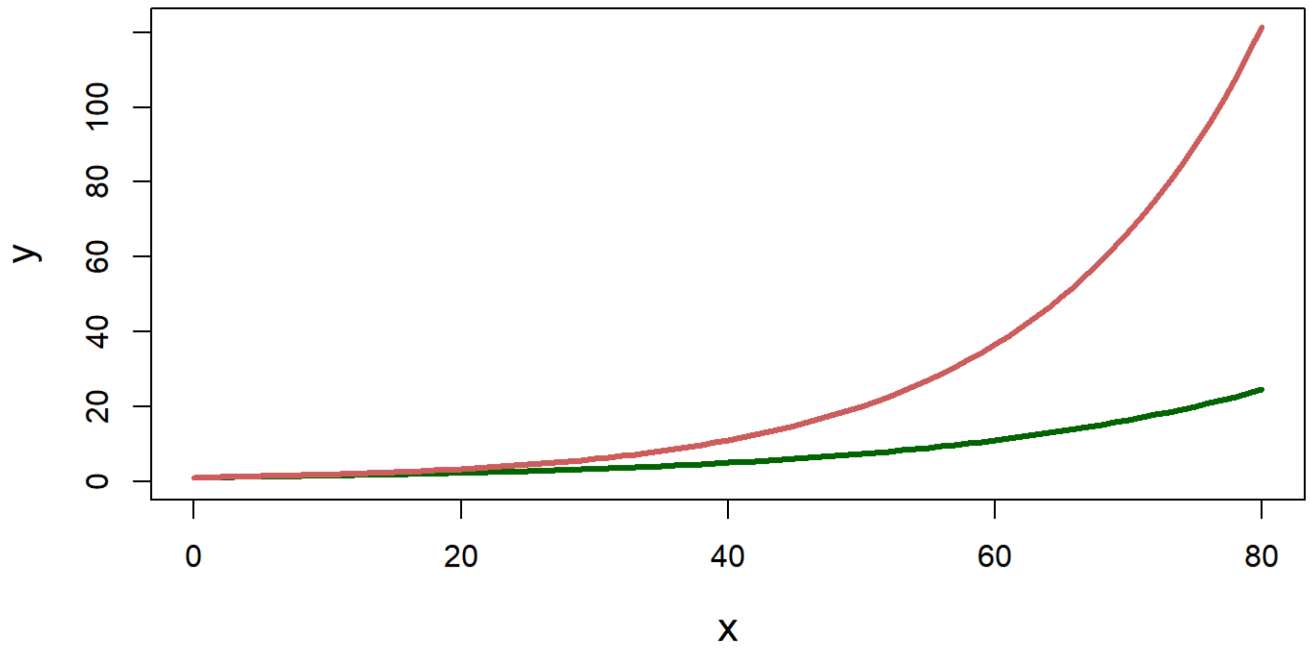
Other Models

Other Models

- There are many different types of models, and more being invented all the time for new types of data and contexts.
- What follows are some examples for modelling growth over time, and some of the usages.

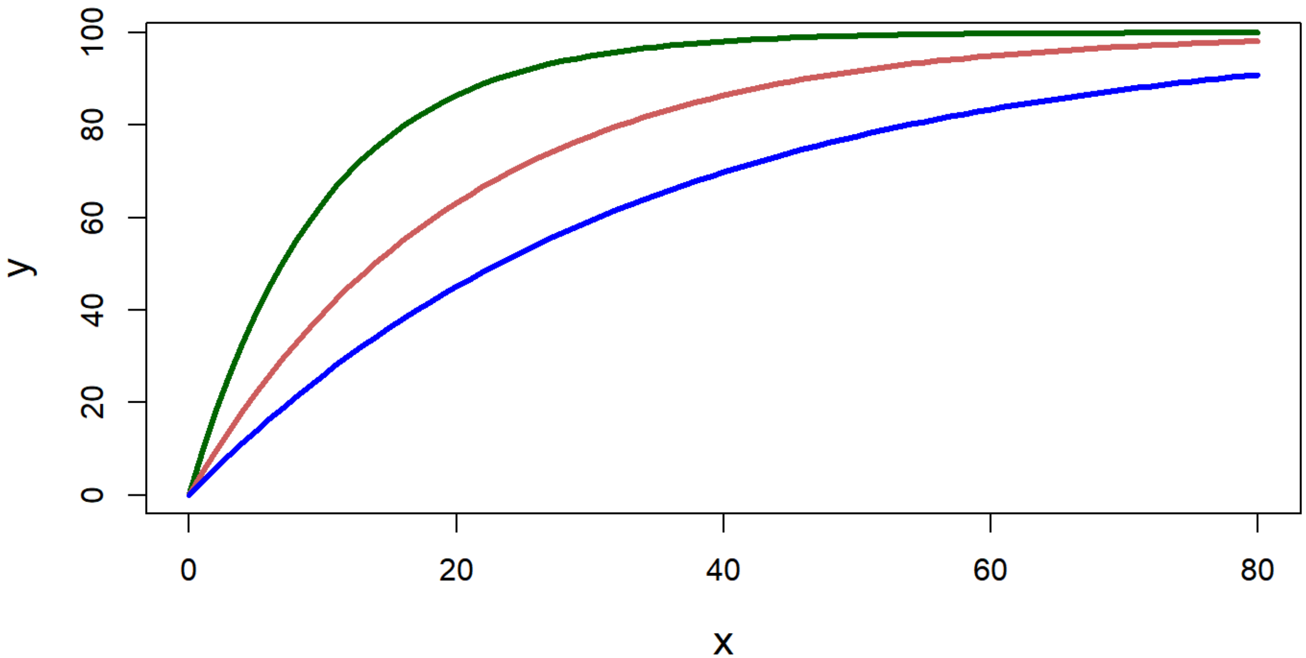
Exponential growth: $y = k_0 e^{k_1 x}$

- Uses: biology (population), physics (nuclear chain reaction), economics/finance, IT (processing power)



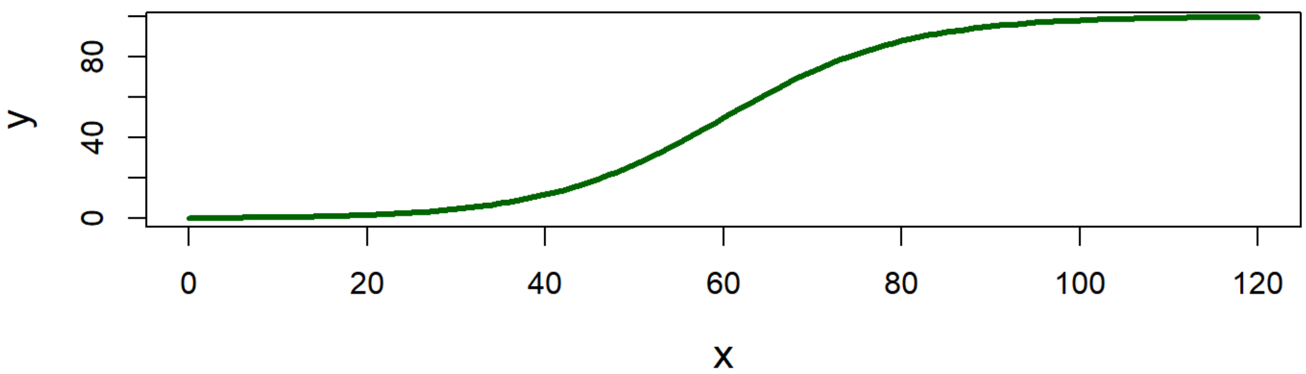
Exponential decay: $y = k_0(1 - e^{k_1 x})$

- Uses: beer froth Ig NobelPrize, fluid dynamics, luminescence, geophysics, vibrations



Logistic growth: $y = \frac{c}{1+e^{-k(x-m)}}$

- $k > 0$
- the maximum growth is $< c$, as there is a horizontal asymptote at $Y = c$
- the maximum growth rate occurs at $x = m$ (when $Y = \frac{c}{2}$).
- Uses: ecology (population), linguistics (language), agriculture (crops), chemistry (reactions), machine learning (neural networks)



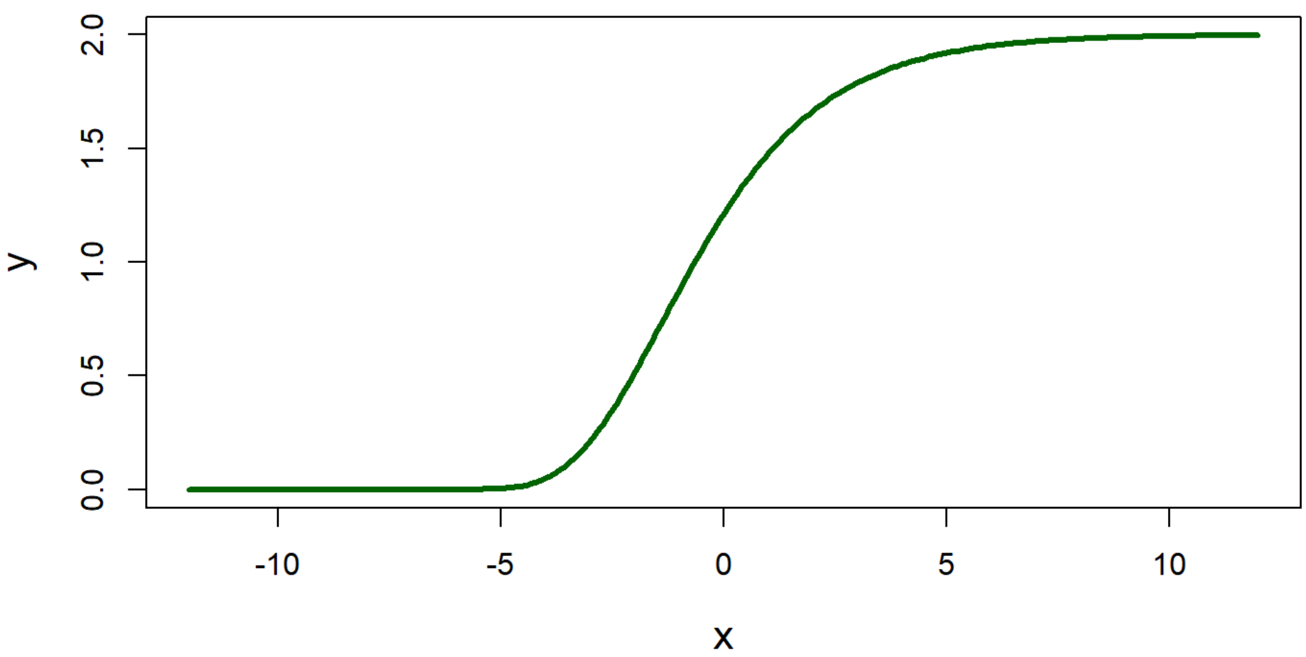
Gompertz: $y = ae^{-be^{-ct}}$

- horizontal asymptote at $y = a$
- b is the displacement along the x axis
- $c > 0$ sets the growth
- Uses: cancer tumours, bacterial growth

```r
a = 2
b = 0.5
c = 0.5

x <- seq(-12, 12, 0.1)
y <- a * exp(-b * exp(-c * x))

plot(x, y, pch = 16, ylim = c(0, max(y)), type = "l", lwd = 3, cex.lab = 1.3, col = "darkgreen")
```



Summary

Fitting models requires educated trial and error. No model fits perfectly, rather we are looking for a way to describe the main shape of the data. Linear and non-linear models have different advantages and disadvantages.

Key Words

linear, non-linear, parametes, power, degree, aymptotes, polynomial, quadratic, cubic, exponential