MTHM506/COMM511 - Statistical Data Modelling

Topic 1 - Beyond Linear Modelling

Preliminaries

In this session, we will start Topic 1 and begin extending the traditional linear modelling framework. These notes refer to Topics 1.1-1.5 (and 1.10) from the lecture notes. All practical aspects in this session will be done using the `fish` dataframe. This can be found in `datasets.RData` on the course ELE page and can be loaded into `R` using the `load()` function.

```r
# Loading datasets required
load('datasets.RData')
```

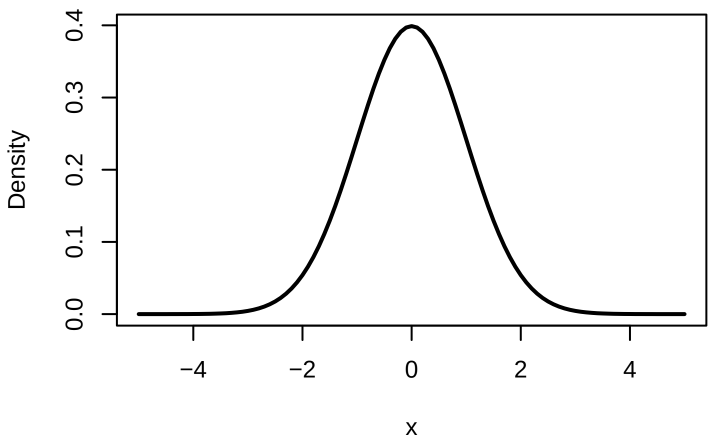
Introduction

In order to perform "Statistical Data Modelling", we need a framework in which to start modelling data. This is where Linear Modelling (LM) comes in, as it is the first framework that you will have seen or considered. Formally, the way we write a model down is through a random variable Y_i. We need to decide what the distribution is, and this is a choice that we as modellers have. It will need to reflect and respect the nature of the data. LM does not always allow for this as it forces a Normal distribution i.e.

$$Y_i \sim N(\mu_i, \sigma^2), \quad Y_i \text{ indep.}$$

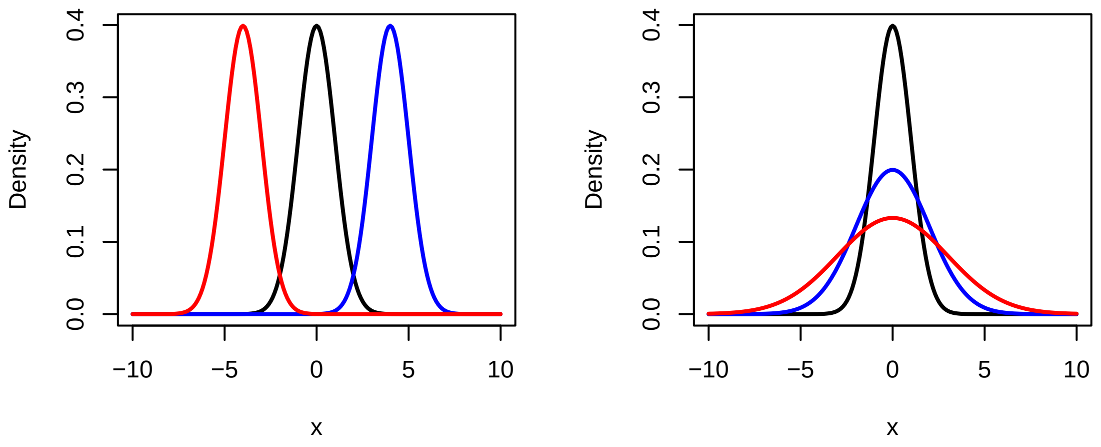
The normal distribution is specified through the mean μ_i and standard deviation σ.

```r
# x values
x <- seq(-5, 5, by = 0.1)

# Getting probability density function
y <- dnorm(x, mean = 0, sd = 1)

# Plotting normal distrbution
plot(x, y, type = 'l', xlab = 'x', ylab = 'Density', lwd = 2)
```



The μ tells us where the peak is and the σ tells us the spread of the data.

```r
# 1x2 plot
par(mfrow=c(1,2))

# x values
x <- seq(-10, 10, by = 0.1)

# Getting probability density function
y1 <- dnorm(x, mean = 0, sd = 1)
y2 <- dnorm(x, mean = 4, sd = 1)
y3 <- dnorm(x, mean = -4, sd = 1)

# Plotting normal distrbution
plot(x, y1, type = 'l', xlab = 'x', ylab = 'Density', lwd = 2)
lines(x, y2, col = 'blue', lwd = 2)
lines(x, y3, col = 'red', lwd = 2)

# Getting probability density function
y1 <- dnorm(x, mean = 0, sd = 1)
y2 <- dnorm(x, mean = 0, sd = 2)
y3 <- dnorm(x, mean = 0, sd = 3)

# Plotting normal distrbution
plot(x, y1, type = 'l', xlab = 'x', ylab = 'Density', lwd = 2)
lines(x, y2, col = 'blue', lwd = 2)
lines(x, y3, col = 'red', lwd = 2)
```



LM assumes constant variance, and the mean is a function of some covariates:

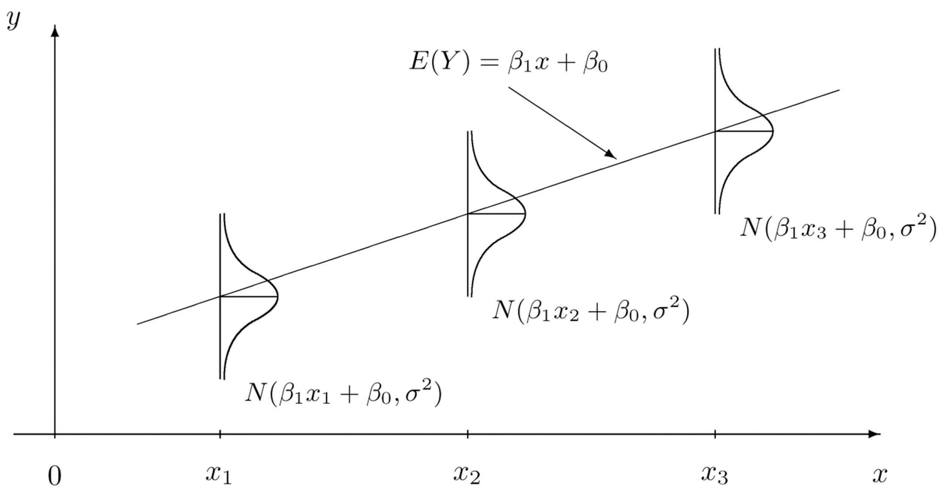
$$Y_i \sim N(\mu_i, \sigma^2), \quad Y_i \text{ indep.}$$

$$\mu_i = \beta_0 + \beta_1 x_i$$

The mean is then assumes a linear function (in beta). Consider an example when y_i is temperature and x_i is continuous, say wind speed, then the above is a model that says temperature is normally distributed with a mean that depends (in a linear way) on wind speed. You would end up with a straight line, which signifies the mean.

The for each value of x we have a distribution that is centered on the mean but has variability around that mean. For example if $x = 25$, then $y = 25 - 0.3 \times 25 = 17.5$ so

$$Y \sim N(17.5, 2) \text{ when } x = 25$$

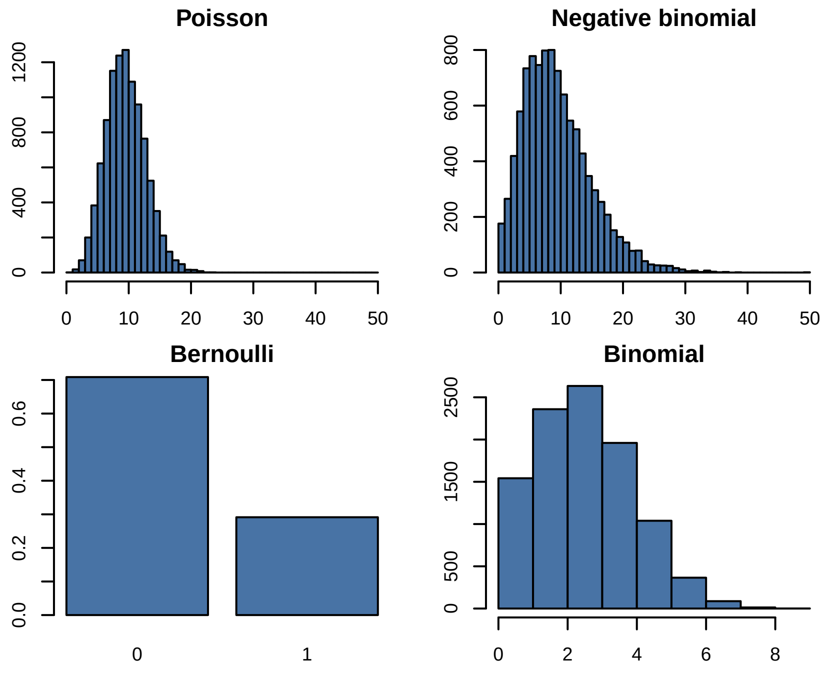


y
$E(Y) = \beta_1 x + \beta_0$
$N(\beta_1 x_3 + \beta_0, \sigma^2)$
$N(\beta_1 x_2 + \beta_0, \sigma^2)$
$N(\beta_1 x_1 + \beta_0, \sigma^2)$
0
x_1
x_2
x_3
x

We can then use this to generate confidence/prediction intervals.

This is the first modelling framework that you will have used seen, but we could argue that is too constraining on the basis that, you can't describe everything with a straight line. You can add more covariates to describe the mean and you are no longer describing the mean using a line (multiple regression). Generally, it's not the best idea to think about straight lines particularly if x is categorical like season or gender.

There are also other considerations here why is it necessarily true that the relationship between y and x should be linear? Why is the variance constant? Why is the distribution normal? What is the normal distribution? The values y can take is $(-\infty, \infty)$ and is continuous. For example, what happens when the data is positive only? Like wind speed? What happens when the data is discrete (count/binary)?



Beyond Linear Modelling

There are a few things to consider when building statistical models. What we want to explore here in Topic 1 is an example where everything you can do with LM fails and therefore why we need to consider extending the LM framework to fit a wider set of data using different distributions.

Lets look at the fish example from the lectures. There is threshold on fish length that authorities say fishermen can catch so we need to design a net that lets fish below that threshold escape, but catches bigger fish for the fishermen to sell. The dataframe `fish` pertains from an experiment testing a specific type of net that was designed to to do this. When tested they knew the number of fish of varying length (in cm) available to capture and recorded the number of fish that escaped through the net. Let's look at the data

```
# N rows
nrow(fish)
[1] 42

# Summary
summary(fish)
     length          escape          total       
 Min.   :21.00   Min.   :0.000   Min.   : 1.000  
 1st Qu.:33.25   1st Qu.:0.000   1st Qu.: 1.000  
 Median :43.50   Median :0.000   Median : 3.500  
 Mean   :46.02   Mean   :1.524   Mean   : 6.143  
 3rd Qu.:54.75   3rd Qu.:1.750   3rd Qu.: 9.750  
 Max.   :87.00   Max.   :9.000   Max.   :23.000  

# Top 12 rows
head(fish, 12)
   length escape total
1      21      1     1
2      23      1     1
3      25      6     6
4      26      6     6
5      27      9     9
6      28      7     8
7      29      3     3
8      30      8    11
9      31      6     7
10     32      2     6
11     33      6    21
12     34      5    23
```

The `fish` dataset contains 42 observations. The data tells use that there was one fish available to capture of length 21cm, which escaped. Then there were also one fish of length 23cm, which also escaped. Then as the fish length goes up we see that there were 21 fish of length 33cm, six of those escaped and so on. We can generally see from the data that the larger the fish, the less likely they are to escape as the net will not allow it.

What is the question here? What is the probability/propensity that fish escape the net as a function of length. In this example, some fish will get caught but we need to see a high probability of escaping at smaller lengths. So we need to do is ask ourselves what are we modelling. Naively, we could say that `escape` here is our Y_i.

In a LM framework we would model it like this:

$$Y_i \sim N(\mu_i, \sigma^2), \quad Y_i \text{ indep.}$$

$$\mu_i = \beta_0 + \beta_1 x_i$$

where Y_i is the number of fish escaping and x_i is the length (in cm)

Whats wrong with this formulation? If we look at the data:

```
# Top 12 rows
head(fish, 12)
   length escape total
1      21      1     1
2      23      1     1
3      25      6     6
4      26      6     6
5      27      9     9
6      28      7     8
7      29      3     3
8      30      8    11
9      31      6     7
10     32      2     6
11     33      6    21
12     34      5    23
```

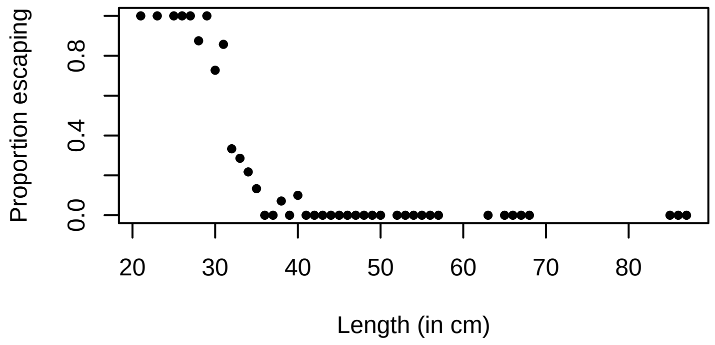
At length 25 we have 6 fish escaping, and at length 33 we have 6 fish escaping. However these are NOT the same as we have 6 out of 6 escaping for length 25 and 6 out of 21 escaping for length 31. So our escape variable is almost meaningless without context that there were known numbers to be caught. Therefore this model is not suitable. It tells use nothing about the chances of escaping because it tells us nothing about the total number of fish. Furthermore, y_i is discrete and bounded by zero so violates assumptions of the Normal distribution.

A more sensible thing which is to consider the proportion of the fish that escape

$$z_i = \frac{y_i}{N_i}$$

Now we can see the probability of escape is bounded between zero and one because it is a proportion. And plotting that as a function of fish length you will see that you know the typical "S" shape/relationship you can find in proportion data. This shape arises because the bounding above by one and below by zero.

```
# Getting proportions
fish$prop <- fish$escape/fish$total

# Plotting proportion as a function of length
plot(fish$length, fish$prop, pch = 20, xlab = 'Length (in cm)', ylab = 'Proportion escaping')
```



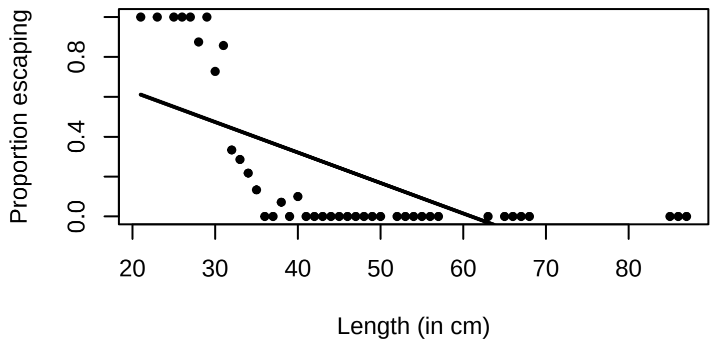
So how should we model this? Should we model this as a LM?

$$Z_i \sim N(\mu_i, \sigma^2), \quad Z_i \text{ indep.}$$

$$\mu_i = \beta_0 + \beta_1 x_i$$

where z_i is the proportion of fish escaping and x_i is the length (in cm)

```r
# Modelling the proportion
mod <- lm(prop ~ length, data = fish)

# Plotting proportion as a function of length
plot(fish$length, fish$prop, pch = 20, xlab = 'Length (in cm)', ylab = 'Proportion escaping')

# Plotting the fitted line
lines(fish$length, mod$fitted.values, lwd = 2)
```



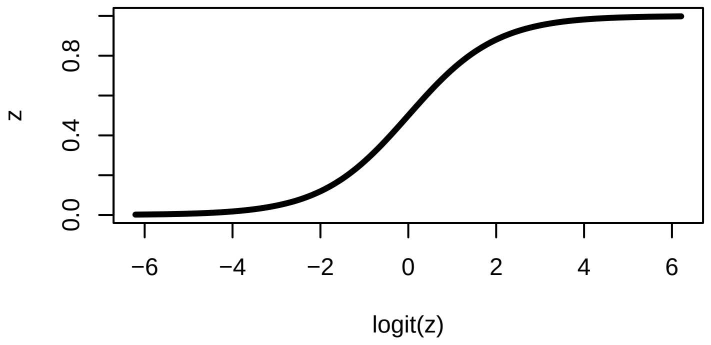
Is this OK? Does it respect the nature of the data? No - the Normal distribution is unbounded but z is between zero and one.

If we want to go down the LM route we could consider a transformation of the response variable to convert the bounding between zero and one to something that is between $-\infty$ and ∞. Furthermore, transformation might actually help in making relationships a bit more linear. How do we convert a range of zero and one to $-\infty$ and ∞? Luckily, we can use the logit function:

$$logit(z) = \log\left(\frac{z}{1-z}\right)$$

so for $z \in (0,1)$ we see that $logit(z) \in (-\infty, \infty)$:

```r
# Simulate 1000 values between 0 and 1 uniformly
z <- seq(0,1,length=500)

# Define the logit as an R function
logit <- function(z){ log(z/(1-z)) }

# Plot the logit
plot(logit(z),z,type="l",lwd=3)
```



So that's great, we can use the logit function to transform our z's and then retry linear regression. However, there remains a problem here in that we cannot compute the logit when z is equal to zero or one.

We can get around this by using a "trick" of calculating an adjusted proportion z_i' instead of the proportion z_i,

$$z_i' = \frac{y_i + 0.5}{N_i + 1}$$

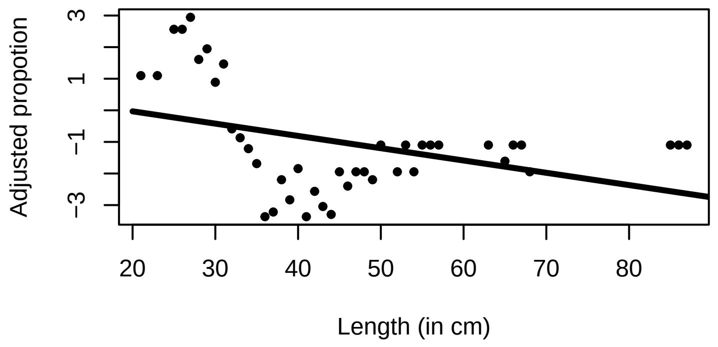
and then modelling $logit(z_i')$ using

$$W_i = logit(Z_i') \sim N(\mu_i, \sigma^2), \quad Z_i \text{ indep.}$$

$$\mu_i = \beta_0 + \beta_1 x_i$$

where Z_i' is the adjusted proportion of fish escaping and x_i is the length (in cm). Lets fit this model in R

```
# Transform the proportions using logit (with adjustment for 0's and 1's)
fish$logitZ <- logit((fish$escape+0.5)/(fish$total+1))

# Fit simple linear model and plot the mean
fish.lm <- lm(logitZ ~ length, data = fish)

# Plotting the data
plot(fish$length,fish$logitZ, pch  =20,
     ylab = "Adjusted propotion",
     xlab = "Length (in cm)")

# Adding the fitted model
xx <- seq(20, 90, len=100)
lines(xx, predict(fish.lm, newdata = data.frame(length = xx)),lwd = 3)
```



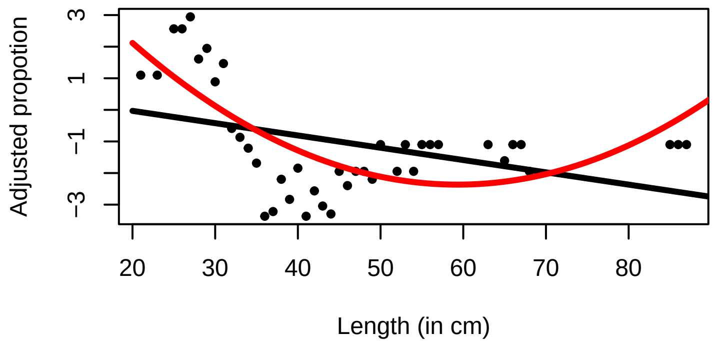
If you ignore the fact that there is uncertainty, the model still fits pretty poorly. The line does not represent what is going on with the data, it is very non-linear. However, LM is not just about straight lines, the linearity in LM is is not because of the x's but in the β's. We can extend this model to a non-linear relationship

$$W_i = logit(Z_i') \sim N(\mu_i, \sigma^2), \quad Z_i \text{ indep.}$$

$$\mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

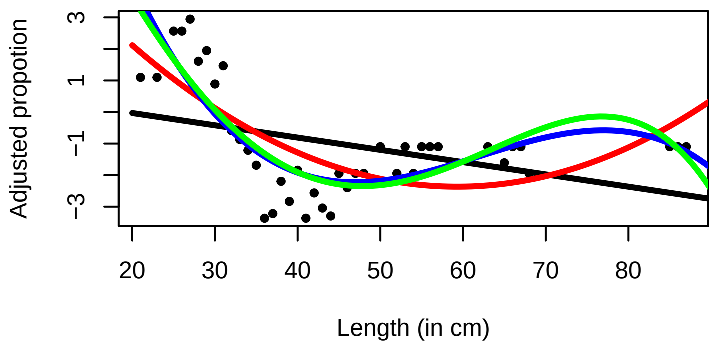
i.e. we're creating a new covariate x_i^2 and giving that its own coefficient which then will give me a quadratic relationship between the fish length and the probability they will escape. So you can write down complicated non-linear relationships in the covariates *as long* as they can be written down as linear in the parameters. Lets fit this in R:

```
# Fit linear model with quadratic relationship
fish$length2 <- fish$length^2
fish.lm2 <- lm(logitZ~length+length2,data=fish)

# Plotting the data
plot(fish$length,fish$logitZ, pch  =20,
     ylab = "Adjusted propotion",
     xlab = "Length (in cm)")

# Adding the fitted model
xx <- seq(20, 90, len=100)
lines(xx, predict(fish.lm, newdata = data.frame(length = xx)),lwd = 3)
lines(xx,predict(fish.lm2, newdata = data.frame(length = xx, length2 = xx^2)),
      lwd = 3, col = "red")
```



Better, but will adding more powers make it better?

```
# Fit linear model with cubic relationship
fish$length3 <- fish$length^3
fish.lm3 <- lm(logitZ~length+length2+length3,data=fish)

# Fit linear model with quartic relationship
fish$length4 <- fish$length^4
fish.lm4 <- lm(logitZ~length+length2+length3+length4,data=fish)

# Plotting the data
plot(fish$length,fish$logitZ, pch = 20,
     ylab = "Adjusted propotion",
     xlab = "Length (in cm)")

# Adding the fitted model
xx <- seq(20, 90, len=100)
lines(xx, predict(fish.lm, newdata = data.frame(length = xx)),lwd = 3)
lines(xx,predict(fish.lm2, newdata = data.frame(length = xx, length2 = xx^2)),
      lwd = 3, col = "red")
lines(xx, predict(fish.lm3, newdata = data.frame(length = xx, length2 = xx^2, length3 = xx^3)),
      lwd = 3, col = "blue")
lines(xx, predict(fish.lm4, newdata = data.frame(length = xx,length2 = xx^2,
                                                 length3 = xx^3, length4 = xx^4)),
      lwd = 3,col = "green")
```



Is this any good? We'll discuss in more detail at a later date when we formally discuss goodness-of-fit. But the blue or green line seem an acceptable fit, certainly more than the black line. The data and model seem to be suggesting similar things, up to a point. Looking at the longer fish lengths, the data is flattening out but the model is not. The variability of the point is very different depending on where you are. This is again an artifact of LM which assumes a constant/fixed variance. In many cases this might not be realistic and is certainly not in the case in the data.

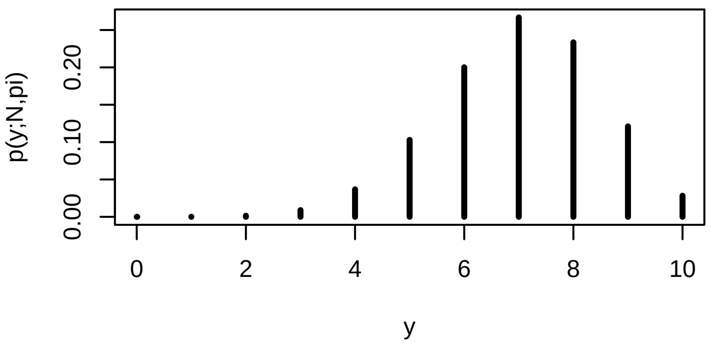
LM is a good and flexible framework, but consider the steps we took here to get to a reasonably fitting model. It would be better to leave things as "untransformed" as possible and model it that way. This way we specify the model that respects the nature of the data.

For example, the fish data is non-negative and count data so is there a better distribution to model this? Yes, the binomial distribution.

$$Y_i \sim Bin(N_i, \pi_i)$$

where π_i is the proportion of y out of N. This has the correct properties for the data we have collected. It is a discrete probability distribution, with one parameter, the probability π_i which is between zero and one. Here's some code to illustrate the binomial distribution when $N = 10$ and $\pi = 0.7$:

```r
N <- 10
pi <- 0.7
plot(0:10,dbinom(0:10,N,pi),type="h",lwd=3,xlab="y",ylab="p(y;N,pi)")
```



We end up with a distribution with a mean of 7, because remember $E(Y) = \mu = N\pi = 10 \times 0.7 = 7$ for the Binomial distribution. It also has variance $Var(Y) = \sigma^2 = N\pi(1 - \pi)$ which is NOT constant as it depends on the mean. We'll talk about this model more in the next session.

There are a lot of probability distributions that can accommodate many different data types. This is what this course will help you do, give you a framework to work with many types of data in a similar way to the Normal distribution and LM.