Lecture 3 – Applying MLE to forgetting

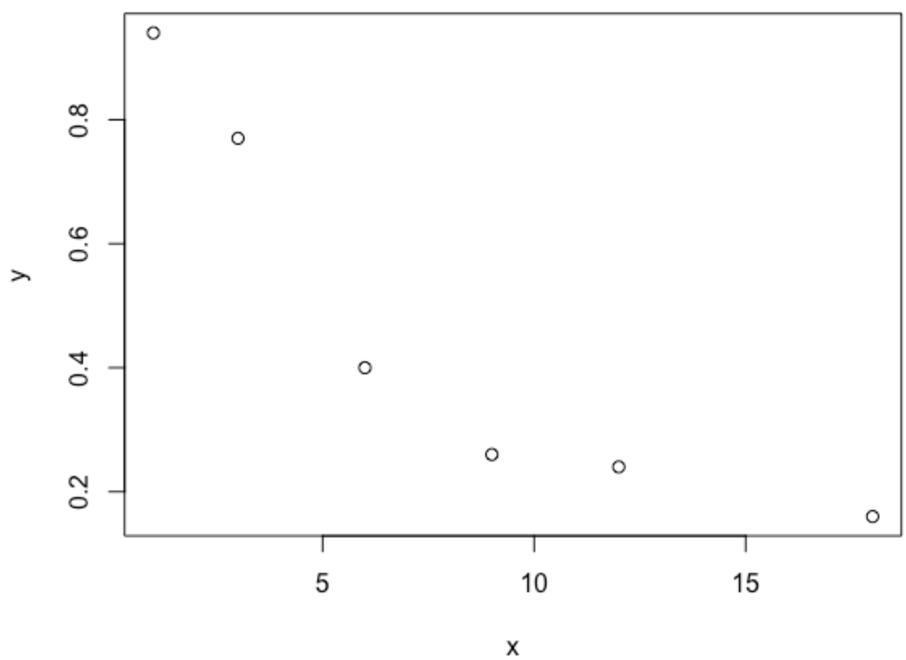
Example: Murdock (1961) – "forgetting curve"

* Ss were presented a "word", counted backward for a short duration, then were asked to recall the word.
  - manipulated the length of the retention interval

Retention interval (x)	1	3	6	9	12	18
Mean prop. recalled (Y)	.94	.77	.40	.26	.24	.16

```r
x = c(1,3,6,9,12,18) # retention intervals (in seconds)
y = c(0.94, 0.77, 0.40, 0.26, 0.24, 0.16) # proportion recalled

plot(y~x)
```



Let's recast our observed proportions as the number of successful recalls out of $N = 100$ trials.

Retention interval (x)	1	3	6	9	12	18
Mean # recalled (Y)	94	77	40	26	24	16

Each person completed 100 trials, and let's assume that each of these 100 trials is independent of the others. Then the binomial model can be used to predict the probability of correctly recalling x items:

$$P(x \mid w) = \binom{100}{x} w^x (1-w)^{100-x}$$

The critical parameter is w: the probability of successful recall on any one trial.

Question: w clearly depends on retention interval t.

Is it a power function?

$$w(t) = at^b$$

or an exponential function?

$$w(t) = ab^t$$

Let's do Maximum likelihood estimation to find out:

To begin, we'll assume a power model: $w = at^b$

First, we rewrite the probability function

$$P(x \mid w) = \binom{100}{x} w^x (1-w)^{100-x}$$

as $P(x \mid a,b) = \binom{100}{x} (at^b)^x (1-at^b)^{100-x}$

which we cast as a likelihood function

$$L(a,b \mid x) = \binom{100}{x} (at^b)^x (1-at^b)^{100-x}$$

Let's use the logarithm to convert products to sums:

$$\log L(a,b \mid x) = \log \binom{100}{x} + x \log(at^b) + (100-x) \log(1-at^b)$$

Now, this is for a single observation x.

We have 6 observations, $x = (x_1, x_2, x_3, x_4, x_5, x_6)$!

If we assume they are all independent, we can just mutiply the likelihoods:

$$L\left(a,b \,\middle|\, x = (x_1, x_2, x_3, x_4, x_5)\right) = \prod_{i=1}^{6} L\left(a,b \,\middle|\, x_i\right)$$

so the log-likelihood is just

$$\log L = \log\left(\prod_{i=1}^{6} L\left(a,b \,\middle|\, x_i\right)\right)$$

$$= \sum_{i=1}^{6} \log L\left(a,b \,\middle|\, x_i\right)$$

$$= \sum_{i=1}^{5}\left[\log\binom{100}{x_i} + x_i \log\left(at^b\right) + \left(100 - x_i\right)\log\left(1 - at^b\right)\right]$$

So, to find the parameters a, b that maximize the likelihood function, given data x, we use R to minimize the neg log-likelihood function $-\log L$, above:

```r
# Murdock (1961) data

times = c(1,3,6,9,12,18) # retention intervals (in seconds)
numRecall = c(94, 77, 40, 26, 24, 16) # number of words correctly recalled

# put these into ONE data "frame"
X = data.frame(times, numRecall)

# first, we'll fit the power model

# define negative log likelihood (the thing we're minimizing)
# note: pars is actually a vector with two numbers (a,b)
nll.power = function(data, pars){
  a = pars[1]
  b = pars[2]
  t = data$times
  x = data$numRecall
  tmp1 = log(choose(100,x))
  tmp2 = x * log(a*t^b)
  tmp3 = (100-x) * log(1-a*t^b)
  return(-1*sum(tmp1 + tmp2 + tmp3))
}
```

extract parameters (a,b) & data (x,t)

$$\log L = \sum_{i=1}^{6}\left[\overbrace{\log\binom{100}{x_i}}^{\text{tmp1}} + \overbrace{x_i \log\left(at^b\right)}^{\text{tmp2}} + \overbrace{\left(100-x_i\right)\log\left(1-at^b\right)}^{\text{tmp3}}\right]$$

Using optim:

```r
# find minimum of NLL
# first, need a guess for initial values of a,b
# let's just try random numbers

a_init = runif(1)
b_init = runif(1)
```

"runif" = random # between 0 & 1.

```r
initPar = c(a_init, b_init) # collect a,b into one parameter vector

optim(par = initPar,
      fn = nll.power,
      data = X)
```

initial guesses

objective function

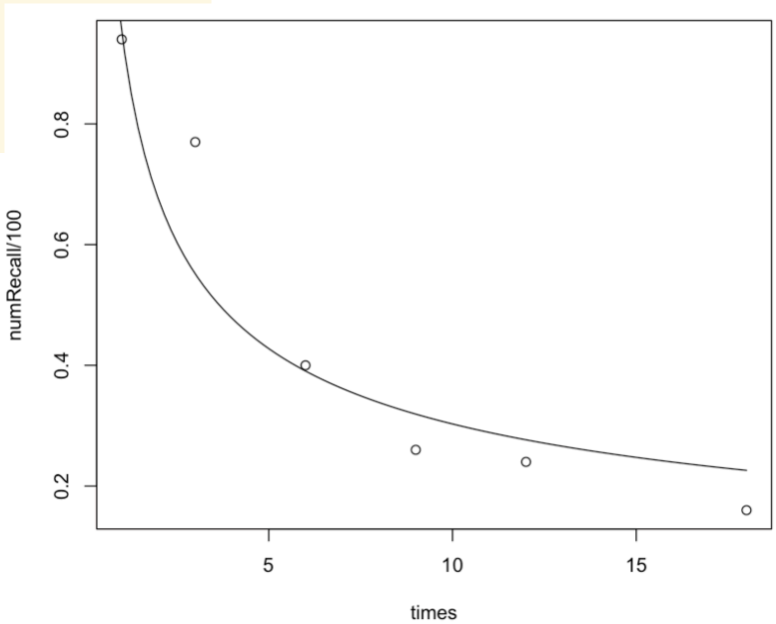
data

```r
# we'll want to extract the MLEs, so assign the optim into an object

model1 = optim(par = initPar,
               fn = nll.power,
               data = X)

# let's plot on top of original data
# first, extract parameters, then plot
a = model1$par[1]
b = model1$par[2]

plot(numRecall/100 ~ times)
curve(a*x^b,
      from=0, to=18,
      add=T)
```



Challenge: now do this with exponential model!