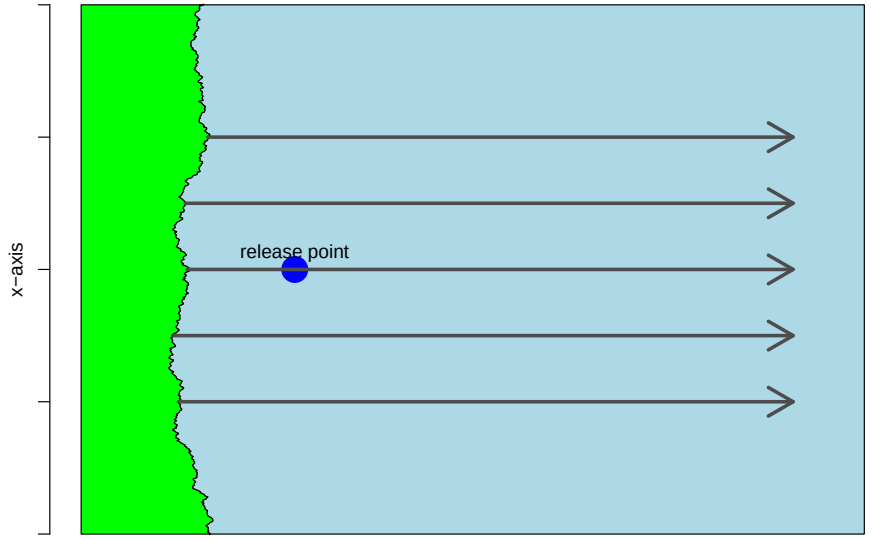
Woods Hole

Negative binomial as overdispersed Poisson

Anders Nielsen

Example: Diffusion of fish from central release

- Part of a larger study shown here:
- Consider the following experiment:
  - Day 0 release of N=3529 fish at a central location (position=0)
  - Day 1–9 a number of fish are caught at different locations
  - Observations are the number caught in each trawl
- The table show the catch at different positions on day 2



	day	position	catch
1	2	80	14
2	2	40	31
3	2	-40	8
4	2	-80	4
5	2	-1	16
6	2	20	20
7	2	120	4
8	2	-120	4

- Assume all fish are independent and following:

$$dx_t^{(i)} = \alpha dt + \sigma dB_t^{(i)}, \quad \& \quad x_0^{(i)} = 0$$

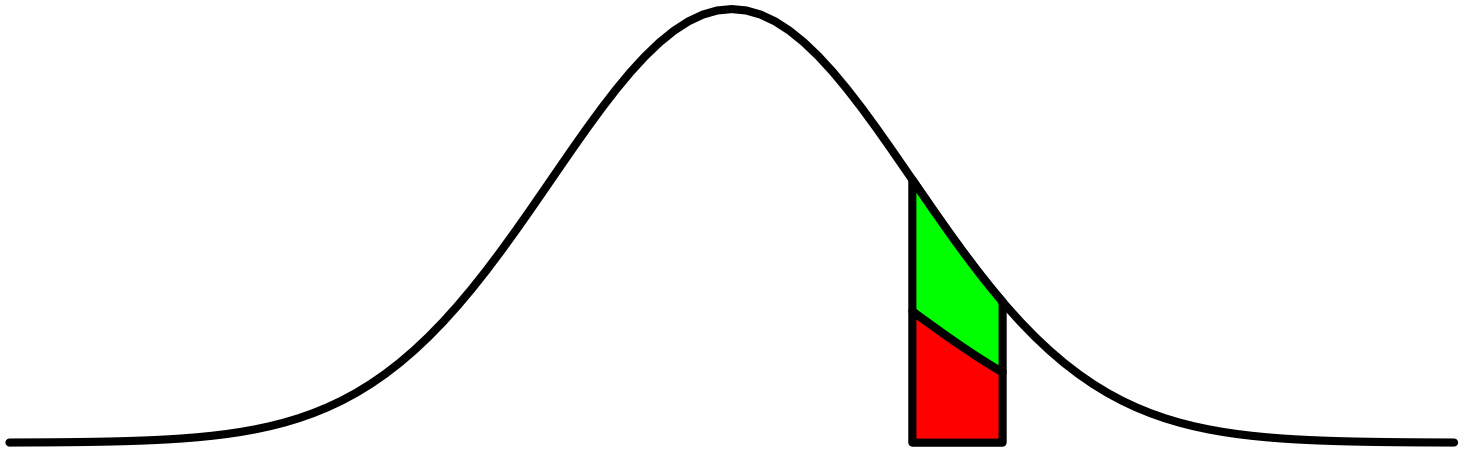
- The population at time t will follow a $\mathcal{N}(\alpha t, \sigma^2 t)$
- The expected number below a trawl with width w, at position p at time t will be:

$$\overline{N}_{w,p,t} = N_{0,0}\left(\Phi\left(\frac{p + w/2 - \alpha t}{\sqrt{\sigma^2 t}}\right) - \Phi\left(\frac{p - w/2 - \alpha t}{\sqrt{\sigma^2 t}}\right)\right)$$

- From this number a fraction q is caught. q reflects the efficiency of the trawl, the expected catch is:

$$\overline{C}_{w,p,t} = q\overline{N}_{w,p,t}$$

- If $w << \sqrt{\sigma^2 t}$ it is a fair to assume: $C_{w,p,t} \sim Pois(q\overline{N}_{w,p,t})$



- Now we can implement the likelihood:

```
for(int i=0; i<nobs; ++i){
  Nbar(i)=pnorm((pos(i) + W/2.0 - alpha * day(i))/sigma/sqrt(day(i)))-
          pnorm((pos(i) - W/2.0 - alpha * day(i))/sigma/sqrt(day(i)));
}
Nbar *= N00;
vector<Type> Cbar=q*Nbar;

Type nll = -sum(dpois(cat,Cbar,true));
```

- The entire TMB program is:

```
#include <TMB.hpp>
template<class Type>
Type objective_function<Type>::operator() ()
{
  DATA_VECTOR(cat);
  DATA_VECTOR(pos);
  DATA_VECTOR(day);
  DATA_SCALAR(N00);
  DATA_SCALAR(W);

  PARAMETER(alpha);
  PARAMETER(logsigma);
  PARAMETER(q);

  Type sigma = exp(logsigma);
  int nobs=cat.size();
  vector<Type> Nbar(nobs), Cbar(nobs);

  for(int i=0; i<nobs; ++i){
    Nbar(i)=pnorm((pos(i) + W/2.0 - alpha * day(i))/sigma/sqrt(day(i)))-
            pnorm((pos(i) - W/2.0 - alpha * day(i))/sigma/sqrt(day(i)));
  }
  Nbar *= N00;
  Cbar=q*Nbar;
  Type nll = -sum(dpois(cat,Cbar,true));
  ADREPORT(sigma);
  return nll;
}
```

- We run it from R with:

```
> library(TMB)
> compile("turbotpois.cpp")
Note: Using Makevars in /home/andni/.R/Makevars
[1] 0
> dyn.load(dynlib("turbotpois"))
> dat<-read.table("turbotpois.dat", head=TRUE)
> data <- list()
> data$cat <- dat$cat
> data$pos <- dat$pos
> data$day <- dat$day
> data$N00 <- 3529
> data$W <- 4.5
> param <- list()
> param$alpha <- 0
> param$logsigma <- log(1000)
> param$q <- 0.5
> obj <- MakeADFun(data, param, DLL="turbotpois", silent=TRUE)
> opt <- nlminb(obj$par, obj$fn, obj$gr, lower=c(-Inf,-Inf,0), upper=c(Inf,Inf,1))
> summary(sdreport(obj))
           Estimate Std. Error
alpha    12.5061258 3.92320213
logsigma  3.7825462 0.10373799
q         0.2083292 0.02217766
sigma    43.9277480 4.55697614
> opt$objective
[1] 22.28878
```

Exercise 1: In the 1D diffusion fish model we assumed:

$$C_{w,p,t} \sim Pois(\overline{C}_{w,p,t})$$

In the Poisson distribution the variance is equal to the mean, which is an assumption that is not always valid. In this exercise we will expand the model and test the assumption.

- Consider the model:

$$C \sim \text{Pois}(\lambda), \quad \text{where} \quad \lambda \sim \Gamma\left(n, \frac{1-\phi}{\phi}\right) \quad 0 < \phi < 1$$

- It can be shown that:

$$C \sim \text{Nbinom}(n, \phi)$$

- This negative binomial has mean $E(C) = n\frac{1-\phi}{\phi}$ and the variance is $E(C)/\phi$ (so greater than the mean)
- To extend the 1D diffusion fish model we use:

$$C_{w,p,t} \sim \text{Nbinom}\left(\overline{C}_{w,p,t} \cdot \left(\frac{\phi}{1-\phi}\right), \phi\right) \quad (\overline{C} \text{ is calculated as before.})$$

Modify the TMB program to estimate the four model parameters α, σ, q, and ϕ and compare the two models

Solution: The following program fits the model:

```
#include <TMB.hpp>
template<class Type>
Type objective_function<Type>::operator() ()
{
  DATA_VECTOR(cat);
  DATA_VECTOR(pos);
  DATA_VECTOR(day);
  DATA_SCALAR(N00);
  DATA_SCALAR(W);

  PARAMETER(alpha);
  PARAMETER(logsigma);
  PARAMETER(q);
  PARAMETER(phi);

  Type sigma = exp(logsigma);
  int nobs=cat.size();
  vector<Type> Nbar(nobs), Cbar(nobs);

  for(int i=0; i<nobs; ++i){
    Nbar(i)=pnorm((pos(i) + W/2.0 - alpha * day(i))/sigma/sqrt(day(i)))-
            pnorm((pos(i) - W/2.0 - alpha * day(i))/sigma/sqrt(day(i)));
  }
  Nbar *= N00;
  Cbar = q*Nbar*phi/(Type(1)-phi);
  Type nll = -sum(dnbinom(cat,Cbar,phi,true));
  ADREPORT(sigma);
  return nll;
}
```

We run it from R with:

```
> library(TMB)
> compile("turbotnegbin.cpp")
Note: Using Makevars in /home/andni/.R/Makevars
[1] 0
> dyn.load(dynlib("turbotnegbin"))
> dat<-read.table("turbotnegbin.dat", head=TRUE)
> data <- list()
> data$cat <- dat$cat
> data$pos <- dat$pos
> data$day <- dat$day
> data$N00 <- 3529
> data$W <- 4.5
> param <- list()
> param$alpha <- 0
> param$logsigma <- log(1000)
> param$q <- 0.5
> param$phi <- 0.5
> obj <- MakeADFun(data, param, DLL="turbotnegbin", silent=TRUE)
> opt <- nlminb(obj$par, obj$fn, obj$gr, lower=c(-Inf,-Inf,0,0), upper=c(Inf,Inf,1,1))
> summary(sdreport(obj))
           Estimate Std. Error
alpha    12.5464457 4.55070341
logsigma  3.7806342 0.12982962
q         0.2082136 0.02652893
phi       0.7014972 0.32476279
sigma    43.8438400 5.69222930
> opt$objective
[1] 21.93549
```

Model	ℓ	dim	G	P-value
Poisson	22.2888	3	0.7066	40%
Neg. binom.	21.9355	4		