Stochastic systems

Modelling Biological Systems, BIOS13

Biology Dept., Lund University
Mikael Pontarp

Stochastic systems

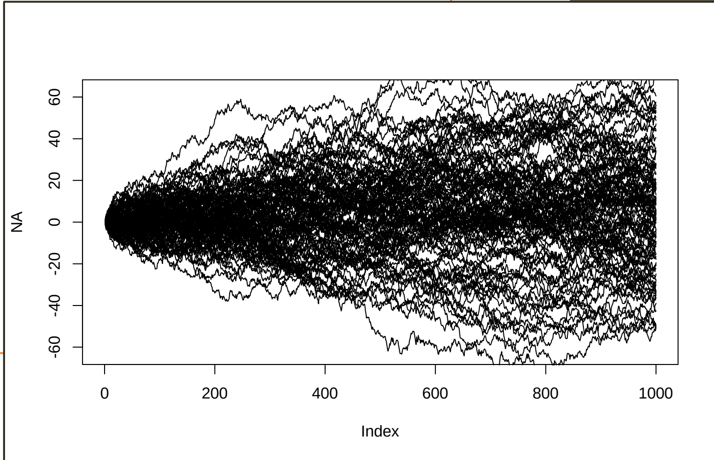
- A stochastic system has an uncertain outcome, it can not be perfectly predicted beforehand.
- Examples: weather, population dynamics, genetic inheritance, football, the throw of a die, etc. Actually most things.
- A single 'run' of a stochastic system is called a *realization*. Last week's weather is a single realization of the weather system. Next year at the same dates, a new realization will occur.
- Although a single realization is hard to predict, one can often say something about the distribution of *all possible* realizations. Such distributions are often calculated from many realizations. Example: weather statistics.

Stochastic models

- A stochastic model is a model of a stochastic system (usually).
- A stochastic model has one or several steps or parts that are drawn from some random distribution, such that the model outcome is more or less unpredictable.
- Example 1: A random walk.

```
n <- 1000 # the length of the walk
x <- rep(0, n) # start with all zeros

# get an empty plot to start with:
plot(NA, type = 'n', xlim=c(0,n), ylim=c(-2*sqrt(n),2*sqrt(n)))
for (iter in 1:100) { # repeat 100 times
     # walk randomly:
     for (i in 1:(n-1)) {
          x[i+1] <- x[i] + rnorm(1)
     }
     # plot:
     lines(1:n, x, col = "black")
}
```



Exercise: Random walk

1. Write a function `randomWalk` that simulates `N` number of random walks for `tmax` time-steps, each walk starting at zero.

2. Plot all random walks in the same plot.

3. Plot the endpoint of all random walks as a histogram. How does `tmax` affect the distribution?

Useful functions in R

- runif(1) uniform distribution
- runif(k) a vector of k random numbers
- runif(1, min=0, max=100) uniform in a specified interval
- rnorm(1) normal distribution
- rnorm(1, mean=1, sd=2) with specified mean and SD

- sample(v, k) samples k elements from vector v
- sample.int(n, k) samples k numbers (integers) from 1:n
- sample(v, k, prob=weights) weighted sampling
- sample.int(n, k, prob=weights) weighted sampling

Common implementations

- Draw a random number from some distribution and ad it to existing variable
  - E.g. random walk
  - E.g. phenotypic change due to mutation
- Sample elements in a vector randomly with some probability
  - E.g. Choose individuals that reproduce or die in a population
  - E.g. Chose mutant individuals from a offspring population
- Draw a random number from a uniform distribution and compare to some value that represent a probability
  - E.g. Evaluate if a reproduction event should happen or not
  - E.g. Evaluate if a mutation event should happen or not

Exercise: A queue

- A model of a standard queue (post office, lunch restaurant, …)
- Every minute, a new customer is added to the queue with probability Pin.
- Also every minute, a customer is finished with probability Pout and leaves the queue.
- Write a function 'runQ.R' that takes three input variables:

  ```
  runQ <- function(Pin, Pout, T_queue) {
  …
  }
  ```

  and

  1) simulates a queue for T_queue minutes

  2) plots the length of the queue over time

Example: The Moran process

- The Moran process is a model in population genetics
- It is a model of the spread of an allele in a population
- A population has N individuals of type A or B.
- Each time-step, one random individual is chosen to die. It is replaced by the copy of another randomly chosen individual.
- The process continues until A or B is fixed.

N=6

A	A	A	A	A	A	A
B	B	B	B	B	B	B
B	~~B~~		B	B	B	B
B	B	B	B	B	B	B
B	B	B	B	~~B~~		A
B	B	B	B	B	B	B

Exercise: The Moran process

1. Write a function `runMoran` that simulates the Moran process `tmax` (input variable) time-steps, starting with a single copy of A and returns the final number of A:s.

2. Write another function `Pfixation` that uses `runMoran` to iterate the Moran process `repeats` times and returns the probability of fixation of A.

3. Test the theoretical result that P(fixation) = 1/N

4. Introduce selection, such that A:s are more likely to reproduce than B:s. The weights are (1+s) and 1, respectively.

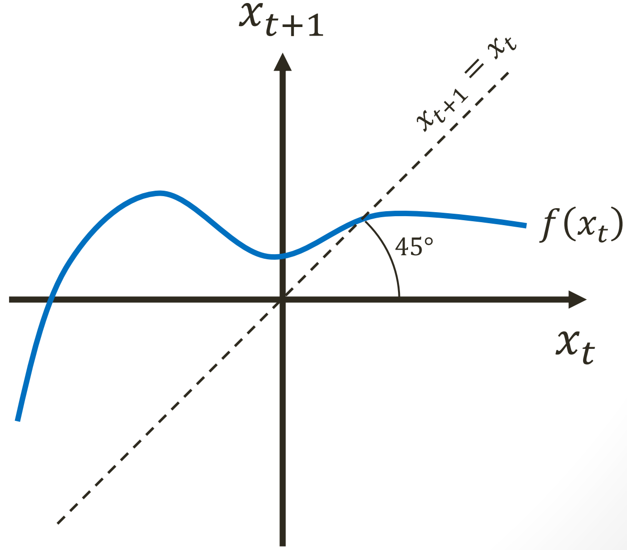
Some theory on discrete time dynamics

A discrete time dynamic system can be described by a *difference equation*, instead of a differential equation:

$$x_{t+1} = f(x_t),$$

where x_t is a state variable. The function $f(x_t)$ is sometimes called a *renewal function.*

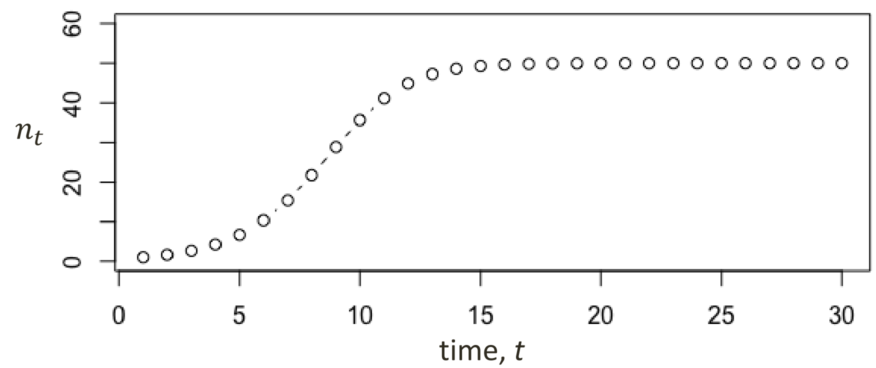
Equilibrium: $x_{t+1} = x_t \Rightarrow x^* = f(x^*)$



An example, the Ricker equation

$$n_{t+1} = f(n_t) = n_t e^{r_0\left(1-\frac{n_t}{K}\right)}$$

n_t is population size at time t
r_0 is an intrinsic growth rate
K is a carrying capacity



<u>Where is the equilibrium?</u>

$$n^* = n^* e^{r_0\left(1-\frac{n^*}{K}\right)}$$

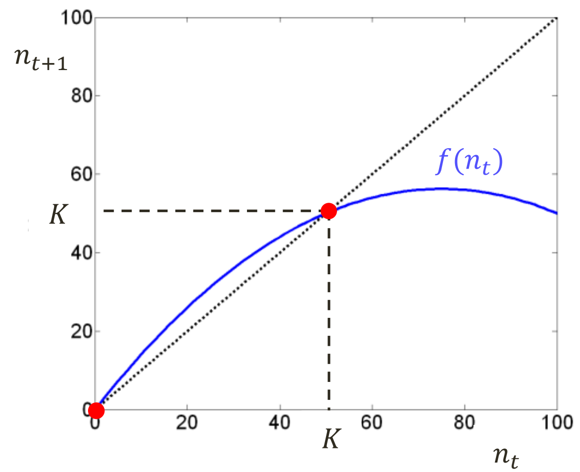
Trivial solution: $n^* = 0$

Non-trivial solution:

$$e^{r_0\left(1-\frac{n^*}{K}\right)} = 1$$

$$r_0\left(1-\frac{n^*}{K}\right) = 0$$

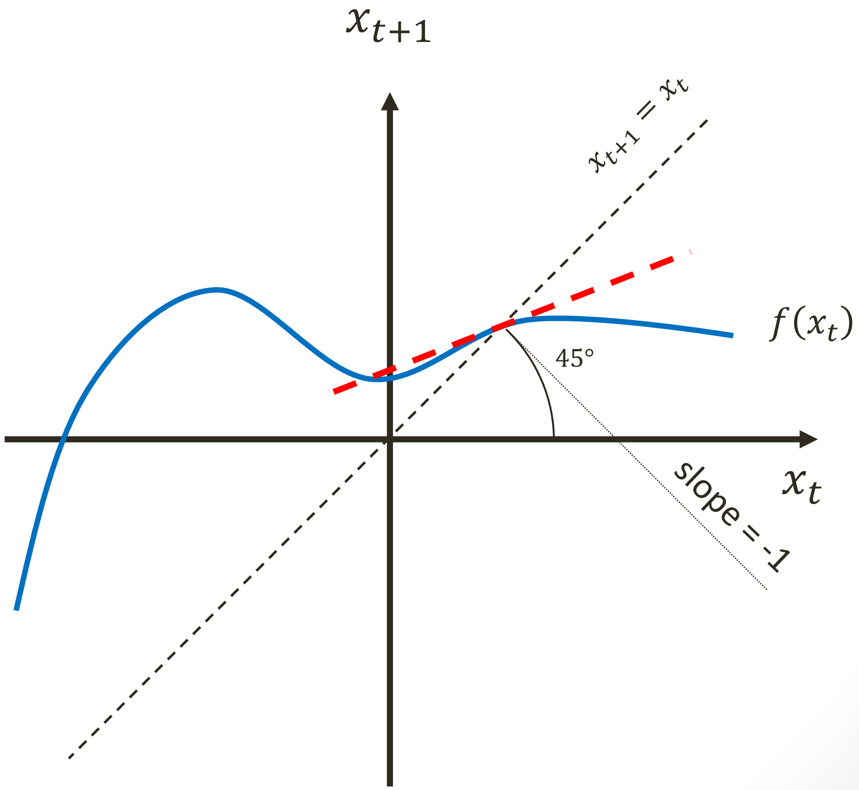
$$n^* = K$$



Stability in discrete time

An equilibrium of a discrete time system depends on the slope of the renewal function at the equilibrium.

It is stable if: $|f'(x^*)| < 1$



Stability of the Ricker equation

For the Ricker equation, the slope at equilibrium (*K*) depends on the intrinsic growth rate, r_0.

$$n_{t+1} = f(n_t) = n_t e^{r_0\left(1-\frac{n_t}{K}\right)}$$

$$n^* = K$$

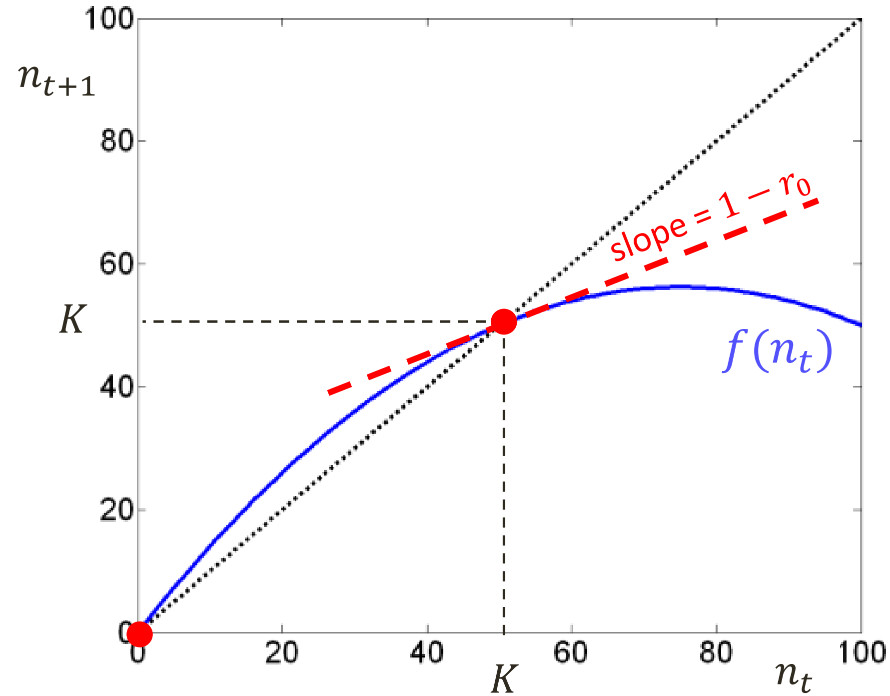
Product rule

$$f'(n_t) = e^{r_0\left(1-\frac{n_t}{K}\right)} + n_t e^{r_0\left(1-\frac{n_t}{K}\right)}\left(-\frac{r_0}{K}\right)$$

$$f'(n^*) = f'(K) =$$
$$= e^0 + Ke^0\left(-\frac{r_0}{K}\right) = 1 - r_0$$

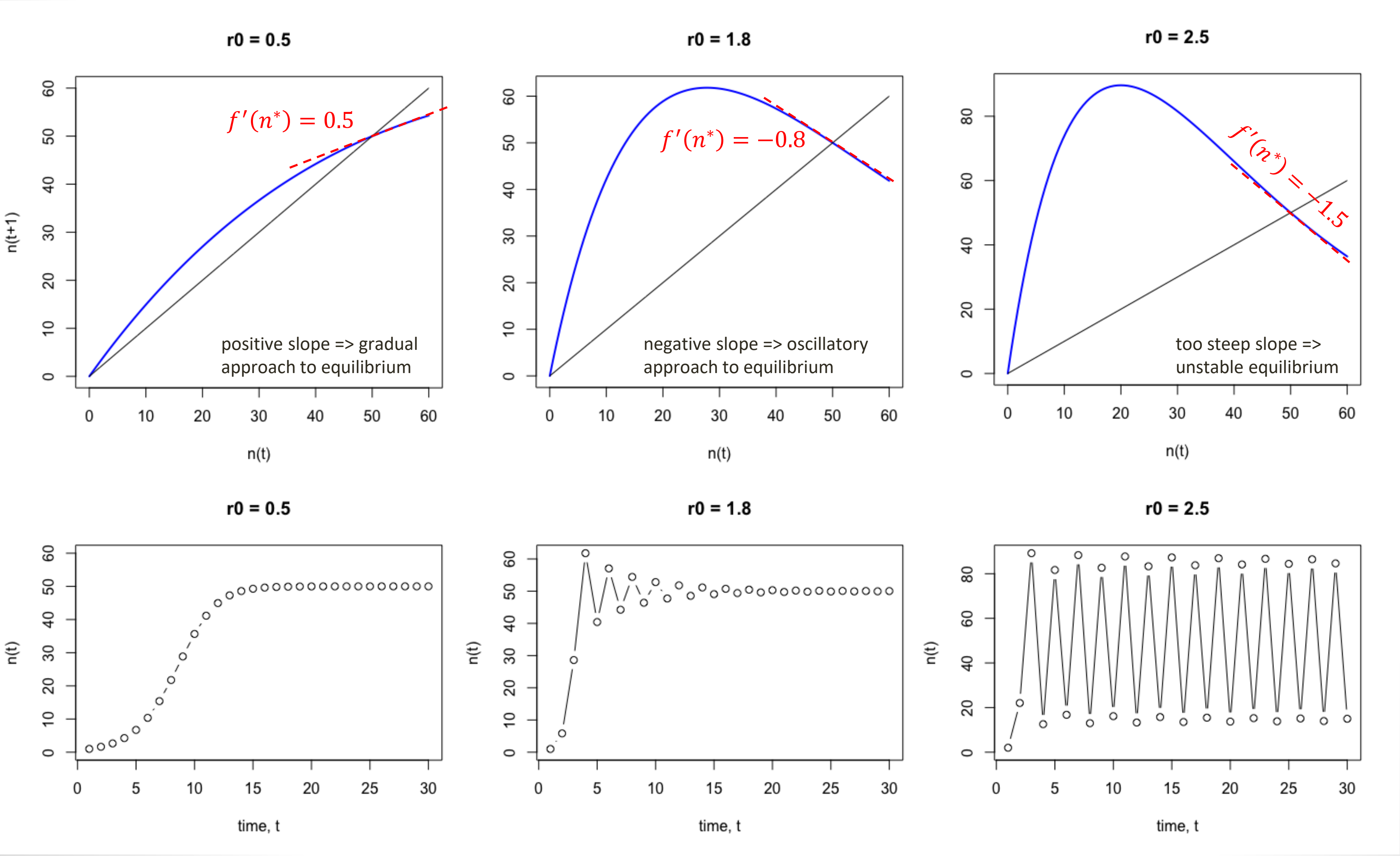
$$|f'(n^*)| < 1 \;\Leftrightarrow\; |1 - r_0| < 1 \Leftrightarrow$$

$$\Leftrightarrow 0 < r_0 < 2$$



Stability of the Ricker equation

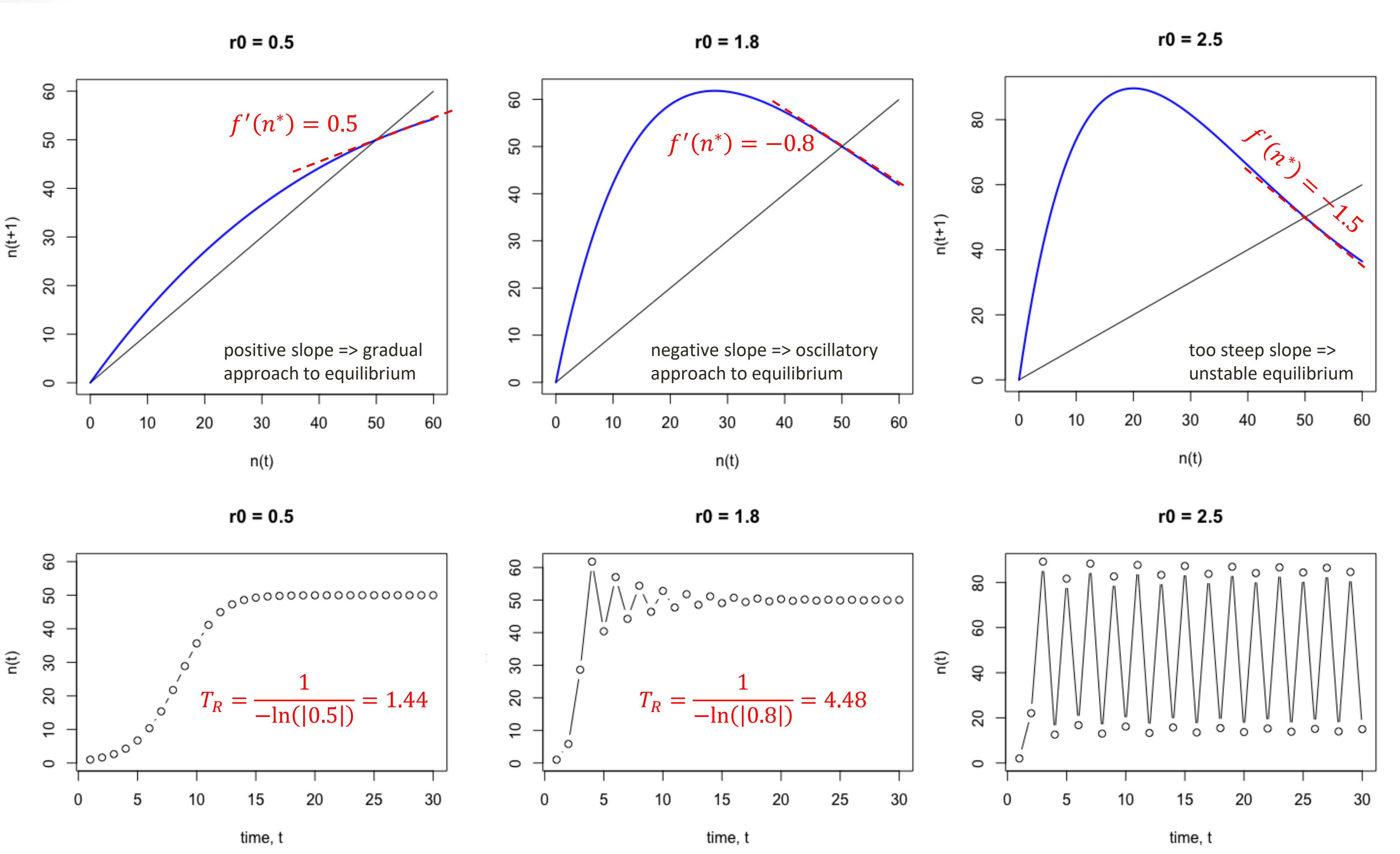
For the Ricker equation, the slope at equilibrium (*K*) depends on the intrinsic growth rate, r_0. $f'(n^*) = 1 - r_0$



Dynamics close to an equilibrium

Generally, the slope of the renewal function at an equilibrium determines the character of the dynamics close to that equilbrium.

The *return time* of a stable equilibrium is given by $T_R = \dfrac{1}{-\ln(|f'(x^*)|)}$



Stochastic discrete time dynamics

A *stochastic* discrete time system can be described by a stochastic difference equation, in the general case something like:

$$x_{t+1} = f(x_t, \varepsilon_t),$$

where ε_t is a stochastic variable, drawn from some distribution. It can represent variable weather conditions or other fluctuating environmental factors.

It can be assumed that $E(\varepsilon_t) = mean(\varepsilon_t) = 0$.

The dynamics at the mean environment,

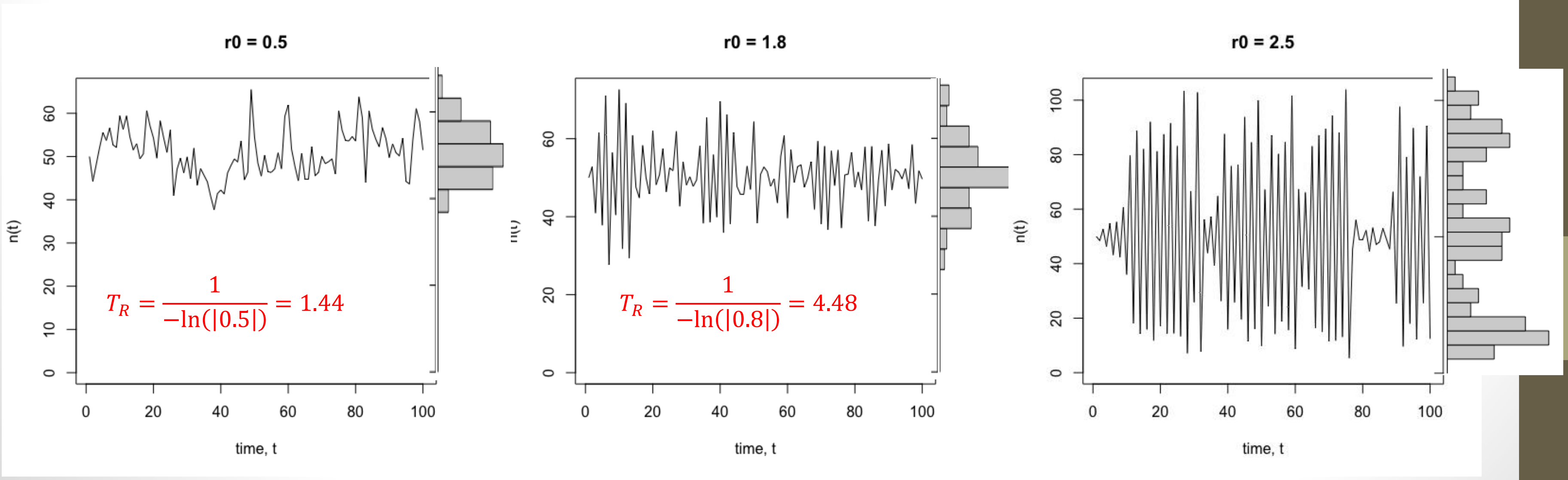
$$x_{t+1} = f(x_t, 0),$$

is called the *deterministic skeleton*, and can be treated as any deterministic difference equation. One can calculate equilibrium states, analyse their stability, and so on.

Stochastic discrete time dynamics

- There is a direct correspondence between the dynamics of the deterministic skeleton and the dynamics of the full, stochastic model.
- If the deterministic skeleton has a stable equilibrium, it can be assumed that the stochastic dynamics will remain in the vicinity of that equilibrium.
- The longer the return time, the larger will the stochastic fluctuations be (each disturbance takes longer time to die out).

Model: $n_{t+1} = n_t e^{r_0\left(1-\frac{n_t}{K}\right)+\varepsilon_t},\ \varepsilon_t \in \mathrm{N}(0,\sigma)$



Stochastic discrete time dynamics

- The slope of the renewal function at the equilibrium determines the character of the stochastic dynamics, too.
- A positive slope implies 'sluggish' dynamics with *positive autocorrelation*, which means that the next value is close to the previous.
- A negative slope implies 'boom-and-bust' dynamics with *negative autocorrelation*, which means that consequitive values are often on opposite sides of the equilibrium.