Variables

x	population size, later vector of population sizes of different types
x'	population sizes at next timestep
t	time
x_t	population sizes at time t
w	growth rate of population per timestep
W	matrix of growth rates. Here W is usually diagonal, and $W_{i,i}$ is the growth rate of type i
s	fitness. $w = (1 + s)$. s is simply a more convenient way to represent fitness when it is close to 1
μ	mutation rate. Rate of switching away from a type per timestep
M	mutation matrix. Entry $M_{i,j}$ row i column j is the rate of switching from j to i per timestep
D	diagonal matrix
V	Matrix of eigenvectors of $W \times M$. We just find V and D so that $W \times M = V \times D \times V^{-1}$

Eigen example with long genome

L	length of genome. Could also be length of a part of the genome that we are considering.
i	index for type with i mutations vs. the optimal type.
x_i	population size for type i
ν	mutation rate per site. Chance of a single site to change in a single timestep. Assuming independence, global rate would be $1-(1-\nu)^L$, but I assume it is simply νL.
$p_{i\to i+1}$	Chance to mutate from i mutations to $i+1$, to add one additional mutation
$p_{i\to i-1}$	Chance to mutate from i mutations to $i-1$, to correct one mutation
M	mutation matrix. $M_{i,i+1}=p_{i\to i+1}$, $M_{i,i-1}=p_{i\to i-1}$
W	fitness matrix for genome. I assume that only the optimal type has fitness>1, so $W_{1,1}=(1+s)$

```
> library(expm);options(warn=-1)
```

Growth of single population

$$x' = w \cdot x$$

Therefore:

$$x_t = w^t x_0$$

Example

> $w = 0.9 \quad ;$
> $x0 = 1$

```
w=0.9;x0=1
```

> $T = 1...20 \quad ; x = \text{rep}(0, \text{length}(T));$
> $\text{for}(t \text{ in } T)\ x_t = w^t\, x0$

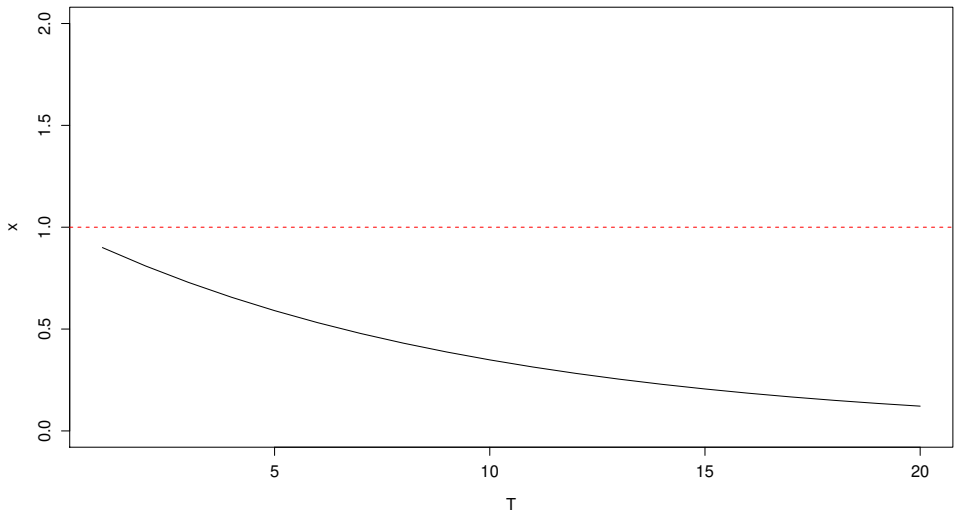
```
T=1:20;x=rep(0,length(T));for(t in T) x[t]=w^t*x0
```

> $y = x$

y=x

```
> plot_xt()
```

plot_xt()




```
>
```

We can represent two types as a vector

> $x0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

```
x0=(t(matrix(c(1, 1), 1, 2 )))
```

>

The growth rates can then be represented by a diagonal matrix

> $W = \begin{pmatrix} 1.1 & 0 \\ 0 & 0.95 \end{pmatrix}$

```
W=(t(matrix(c(1.1, 0, 0, 0.95), 2, 2 )))
```

>

$$x' = W x$$

```
> W %*% x0
```

```
> W %*% x0
```

W %*% x0

	[,1]
[1,]	1.1
[2,]	0.95

```
>W %*% W %*% x0
```

W %*% W %*% x0

	[,1]
[1,]	1.21
[2,]	0.9025

```
>
```

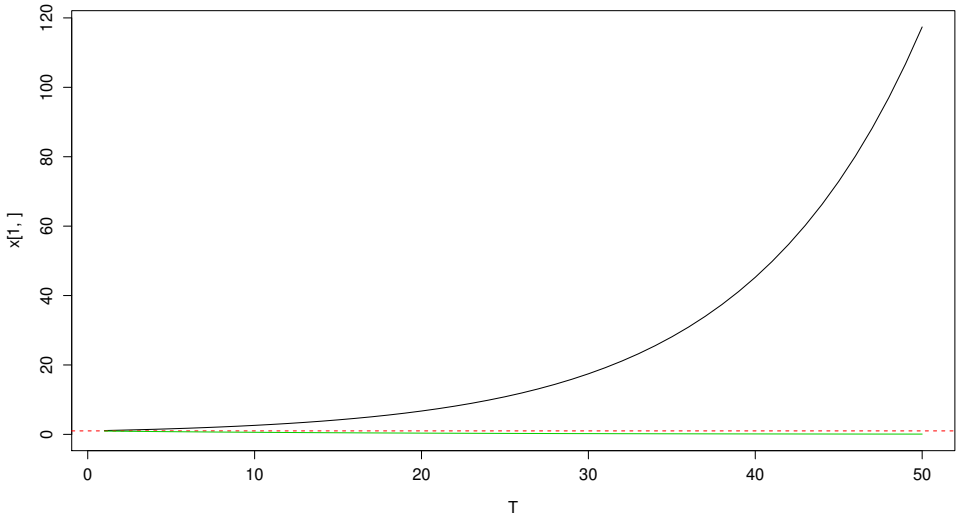
$$x_t = W^t x_0$$

>$T = 1...50 \quad ; x = \text{matrix}(0, 2, \text{length}(T));$
$\text{for}(t \text{ in } T)\, x_{,t} = (W \%\text{^}\%t) \%*\% x0$

```
T=1:50;x=matrix(0,2,length(T));for(t in T) x[,t]=(W%^%t) %*% x0
```

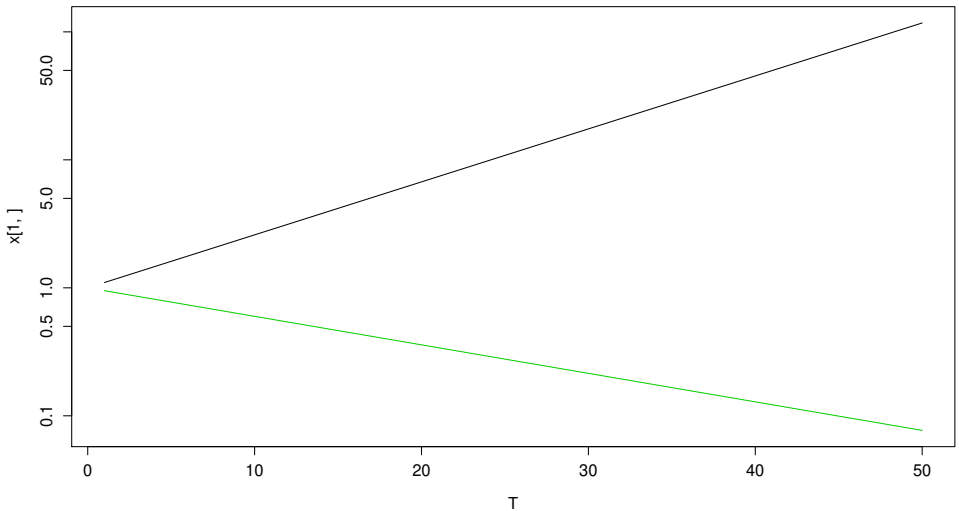
```
> plot_xt2()
```

```
plot_xt2()
```




```
> plot_log_xt2()
>
```

```
plot_log_xt2()
```




```
>
```

Mutations are represented by a stochastic matrix

> $\mu = 0.1;$
> $M = \begin{pmatrix} 1-\mu & 0 \\ \mu & 1 \end{pmatrix}$

```
mu=0.1;M=(t(matrix(c(1-mu, 0, mu, 1), 2, 2 )))
```

> $x0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}; x0$

```
x0=(t(matrix(c(1, 1), 1, 2 )));x0
```

	[,1]
[1,]	1
[2,]	1

```
>M %*% x0
```

```
M %*% x0
```

	[,1]
[1,]	0.9
[2,]	1.1

```
>
```

Now

$$x' = W \times M \times x$$

or

$$x_t = (W \times M)^t x_0$$

If we diagonalize $W \times M$:

$$W \times M = V \times D \times V^{-1}$$

with D diagonal, and V the eigenvectors of $W \times M$

$$(W \times M)^t = V D\, {}^tV^{-1}$$

D^t is dominated by the largest eigenvalue.

> $x0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}; \mu = 0.2; M = \begin{pmatrix} 1-\mu & 0 \\ \mu & 1 \end{pmatrix}; W = \begin{pmatrix} 1.1 & 0 \\ 0 & 0.95 \end{pmatrix}$

```
<, 2, 2 )));W=(t(matrix(c(1.1, 0, 0, 0.95), 2, 2 )))
```

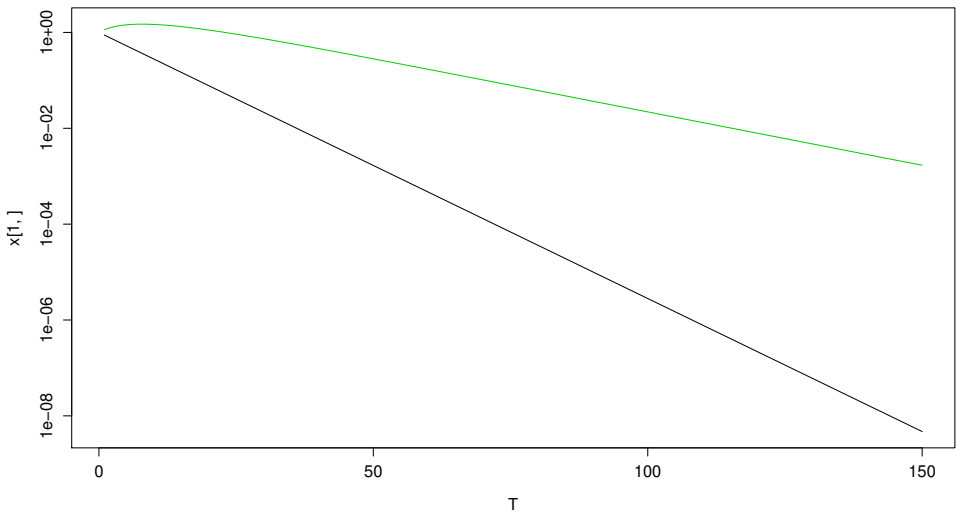
```
> T=1:150;x=matrix(0,2,length(T))
  for(t in T) x[,t]=((W%*%M) %^%t) %*% x0
```

```
T=1:150;x=matrix(0,2,length(T))
> for(t in T) x[,t]=((W%*%M) %^%t) %*% x0
```

```
T=1:150;x=matrix(0,2,length(T))
> for(t in T) x[,t]=((W%*%M) %^%t) %*% x0
```

```
> plot_log_xt2()
```

```
plot_log_xt2()
```




```
>
```

values of μ to try: 0.01, 0.08, 0.2

```
> eigen((W%*%M))
```

```
eigen((W%*%M))
eigen() decomposition
$values
[1] 0.95 0.88

$vectors
```

	[,1]	[,2]
[1,]	0	0.345705358827356
[2,]	1	-0.93834311681711

```
>
```

```
> z=sapply(seq(0,0.2,length=100),function(μ){M=matrix(c(1-μ,μ,0,1),2,2);W=
  matrix(c(1.1,0,0,0.95),2,2);eigen((W%*%M))$values})
```

```
<(1.1, 0, 0, 0.95), 2, 2 )));eigen((W%*%M))$values})
```

```
> plot(seq(0,0.2,length=100),z[1,], type="l",
      ylim=c(0.8,1.1),xlab="mu",ylab="Eigenvalues")
  lines(seq(0,0.2,length=100),z[2,]);v()
```

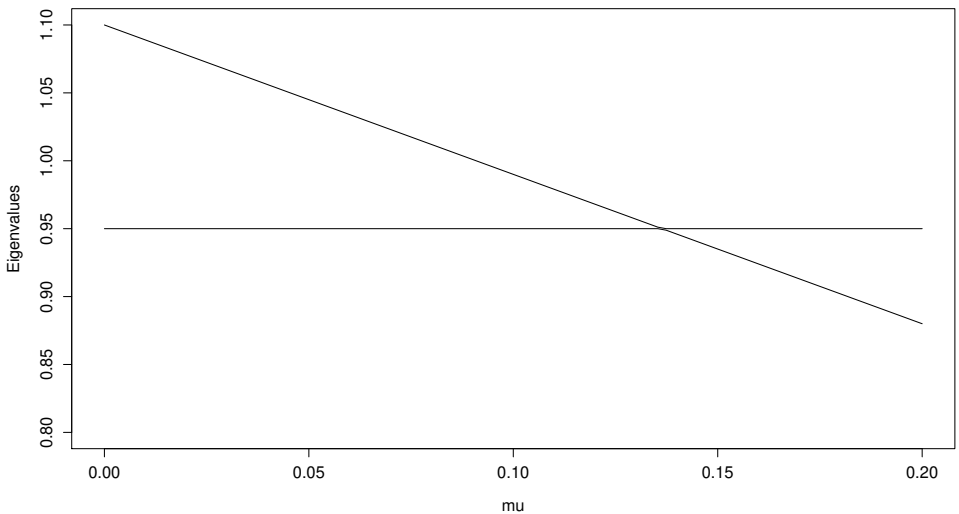
```
>
```

```
> z = sapply(seq(0, 0.2, length = 100), function(μ){M =
  matrix(c(1-μ,μ,μ/100,1-μ/100),2,2);W=matrix(c(1.1,0,0,0.95),2,2);eigen((W%*%M))$values})
```

```
> plot(seq(0,0.2,length=100),z[1,], type="l",
      ylim=c(0.8,1.1),xlab="mu",ylab="Eigenvalues")
  lines(seq(0,0.2,length=100),z[2,]);v()
```

```
> plot(seq(0,0.2,length=100),z[1,], type="l",
       ylim=c(0.8,1.1),xlab="mu",ylab="Eigenvalues")
  lines(seq(0,0.2,length=100),z[2,]);v()
```

```
plot(seq(0,0.2,length=100),z[1,], type="l",
+      ylim=c(0.8,1.1),xlab="mu",ylab="Eigenvalues")
> lines(seq(0,0.2,length=100),z[2,]);v()
```



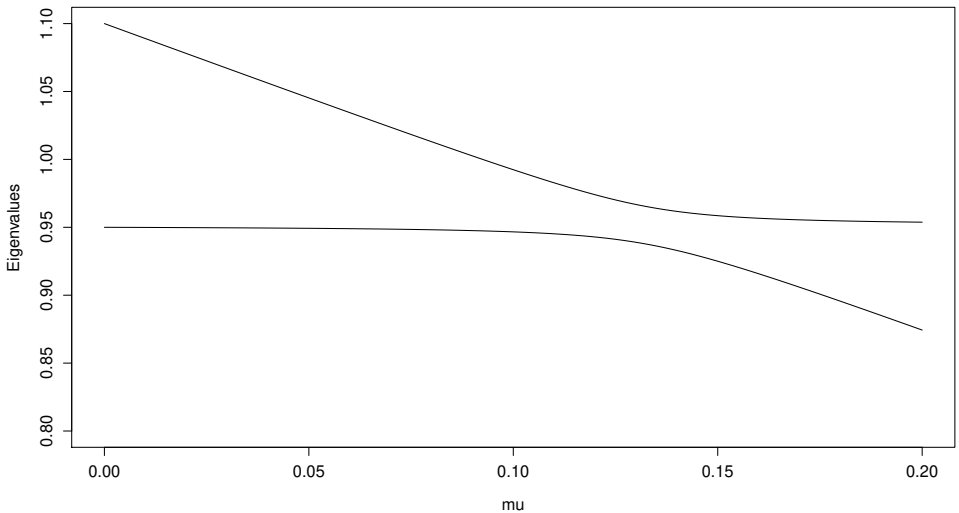

```
>
```

```
> z = sapply(seq(0, 0.2, length = 100), function(μ) {M =
  ( 1−μ   μ/100 ); W = ( 1.1  0 ); eigen((W%*%M))$values})
    μ   1−μ/100          0  0.95
```

```
<(1.1, 0, 0, 0.95), 2, 2 )));eigen((W%*%M))$values})
```

```
> plot(seq(0,0.2,length=100),z[1,], type="l",
       ylim=c(0.8,1.1),xlab="mu",ylab="Eigenvalues")
  lines(seq(0,0.2,length=100),z[2,]);v()
```

```
plot(seq(0,0.2,length=100),z[1,], type="l",
+      ylim=c(0.8,1.1),xlab="mu",ylab="Eigenvalues")
> lines(seq(0,0.2,length=100),z[2,]);v()
```




```
>
```

$$M = \begin{pmatrix} 1-\mu & 0 \\ \mu & 1 \end{pmatrix}; W = \begin{pmatrix} w & 0 \\ 0 & 1 \end{pmatrix}$$

$$W \times M = \begin{pmatrix} w & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1-\mu & 0 \\ \mu & 1 \end{pmatrix} = \begin{pmatrix} (1-\mu)w & 0 \\ \mu & 1 \end{pmatrix}$$

$$W \times M \times \begin{pmatrix} 1 \\ \alpha \end{pmatrix} = \begin{pmatrix} (1-\mu)w & 0 \\ \mu & 1 \end{pmatrix}\begin{pmatrix} 1 \\ \alpha \end{pmatrix} = \begin{pmatrix} (1-\mu)w \\ \mu+\alpha \end{pmatrix}$$

so to keep ratio

$$\frac{\mu+\alpha}{(1-\mu)w} = \frac{\alpha}{1}$$

$$\mu = \alpha(w(1-\mu)-1)$$

$$\alpha = \frac{\mu}{w(1-\mu)-1}$$

only positive if $w(1-\mu)>1$.

If we write $w \equiv (1+s)$ then $(1+s)(1-\mu)>1$

or $s \gtrsim \mu$

Genome of length L

Assume optimal sequence is

AAGCGGCTACTGCAAACGTC...

Manfred Eigen

Peter Schuster

Genome of length L

Assume optimal sequence is

AAGCGGCTACTGCAAACGTC...

We can work with a binary genome where

- 1 identical to optimum
- 0 different from optimum

ν is mutation rate per site, optimum to non-, or back.

1111111111111000000...

Assume only one mutation per timetep, rate $L\nu$.

i number of 0s.

0s	i
1s	$L-i$

$$p_{i\to i+1} = \frac{L-i}{L} L\nu = (L-i)\nu$$

$$p_{i\to i-1} = \frac{i}{L} L\nu = i\nu$$

> $\nu = 0.01; L = 10$

```
nu=0.01;L=10
```

>

$$p_{i \to i+1} = (L - i)\nu$$

$$p_{i \to i-1} = i\,\nu$$

```
> M=matrix(0,L+1,L+1);for(i in 1:(L+1))M[i,i]=1;
```

```
M=matrix(0,L+1,L+1);for(i in 1:(L+1))M[i,i]=1;
```

> $\text{for}(i \text{ in } 0 \ldots (L-1)\,)\,\{M_{i+2,i+1} = (L-i)\,\nu;\, M_{i+1,i+1} = M_{i+1,i+1} - (L-i)\,\nu\}$

```
for(i in 0:(L-1) ) {M[i+2,i+1]=(L-i)*nu;M[i+1,i+1]=M[i+1,i+1]-(L-i)*nu}
```

> $\text{for}(i \text{ in } 1 \ldots (L)\,)\,\{M_{i,i+1} = i\,\nu; \qquad M_{i+1,i+1} = M_{i+1,i+1} - i\,\nu\}$

```
for(i in 1:(L) ) {M[i,i+1]=i*nu;M[i+1,i+1]=M[i+1,i+1]-i*nu}
```

> M

M

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]
[1,]	0.9	0.01	0	0	0	0	0	0	0	0	0
[2,]	0.1	0.9	0.02	0	0	0	0	0	0	0	0
[3,]	0	0.09	0.9	0.03	0	0	0	0	0	0	0
[4,]	0	0	0.08	0.9	0.04	0	0	0	0	0	0
[5,]	0	0	0	0.07	0.9	0.05	0	0	0	0	0
[6,]	0	0	0	0	0.06	0.9	0.06	0	0	0	0
[7,]	0	0	0	0	0	0.05	0.9	0.07	0	0	0
[8,]	0	0	0	0	0	0	0.04	0.9	0.08	0	0
[9,]	0	0	0	0	0	0	0	0.03	0.9	0.09	0
[10,]	0	0	0	0	0	0	0	0	0.02	0.9	0.1
[11,]	0	0	0	0	0	0	0	0	0	0.01	0.9

>$s=0.08; W=\text{matrix}(0, L+1, L+1); \text{for}(i \text{ in } 0...L)\, W_{i+1,i+1}=1$

```
s=0.08;W=matrix(0,L+1,L+1);for(i in 0:L) W[i+1,i+1]=1
```

> $W_{1,1}=1+s; W$

```
W[1,1]=1+s;W
```

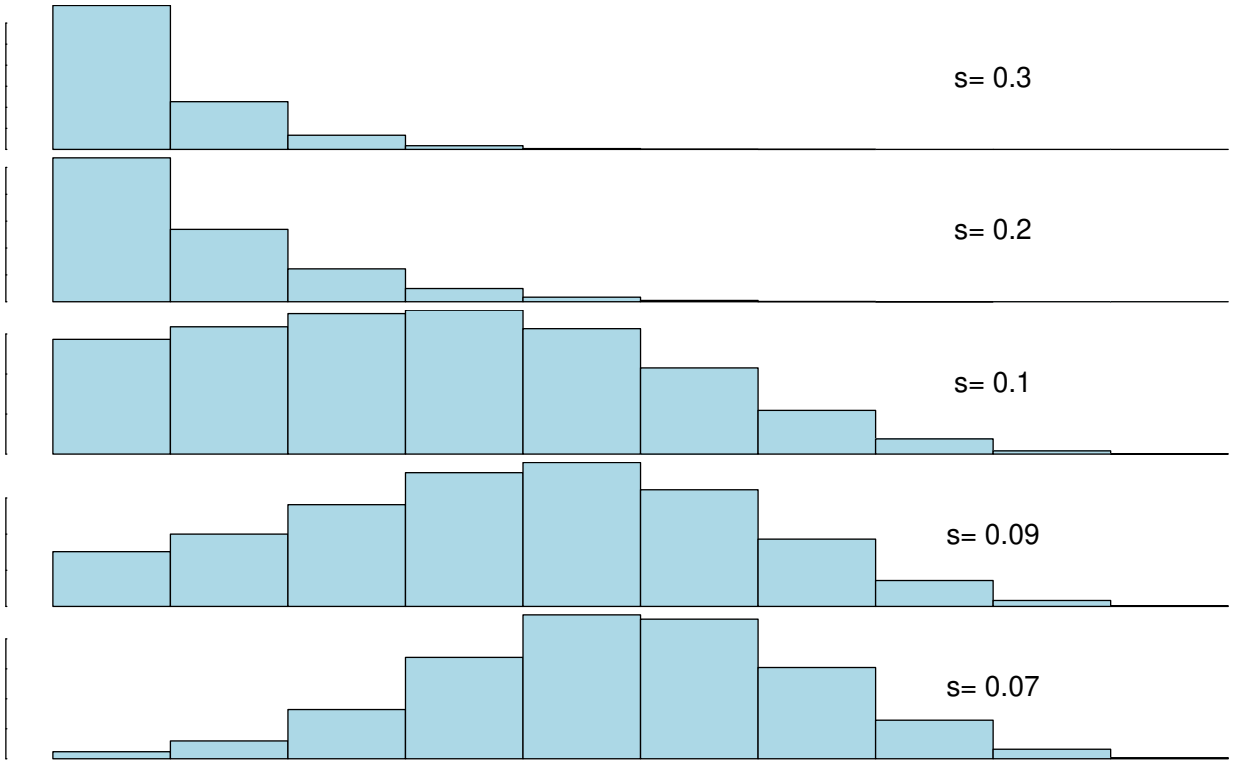
	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]
[1,]	1.08	0	0	0	0	0	0	0	0	0	0
[2,]	0	1	0	0	0	0	0	0	0	0	0
[3,]	0	0	1	0	0	0	0	0	0	0	0
[4,]	0	0	0	1	0	0	0	0	0	0	0
[5,]	0	0	0	0	1	0	0	0	0	0	0
[6,]	0	0	0	0	0	1	0	0	0	0	0
[7,]	0	0	0	0	0	0	1	0	0	0	0
[8,]	0	0	0	0	0	0	0	1	0	0	0
[9,]	0	0	0	0	0	0	0	0	1	0	0
[10,]	0	0	0	0	0	0	0	0	0	1	0
[11,]	0	0	0	0	0	0	0	0	0	0	1

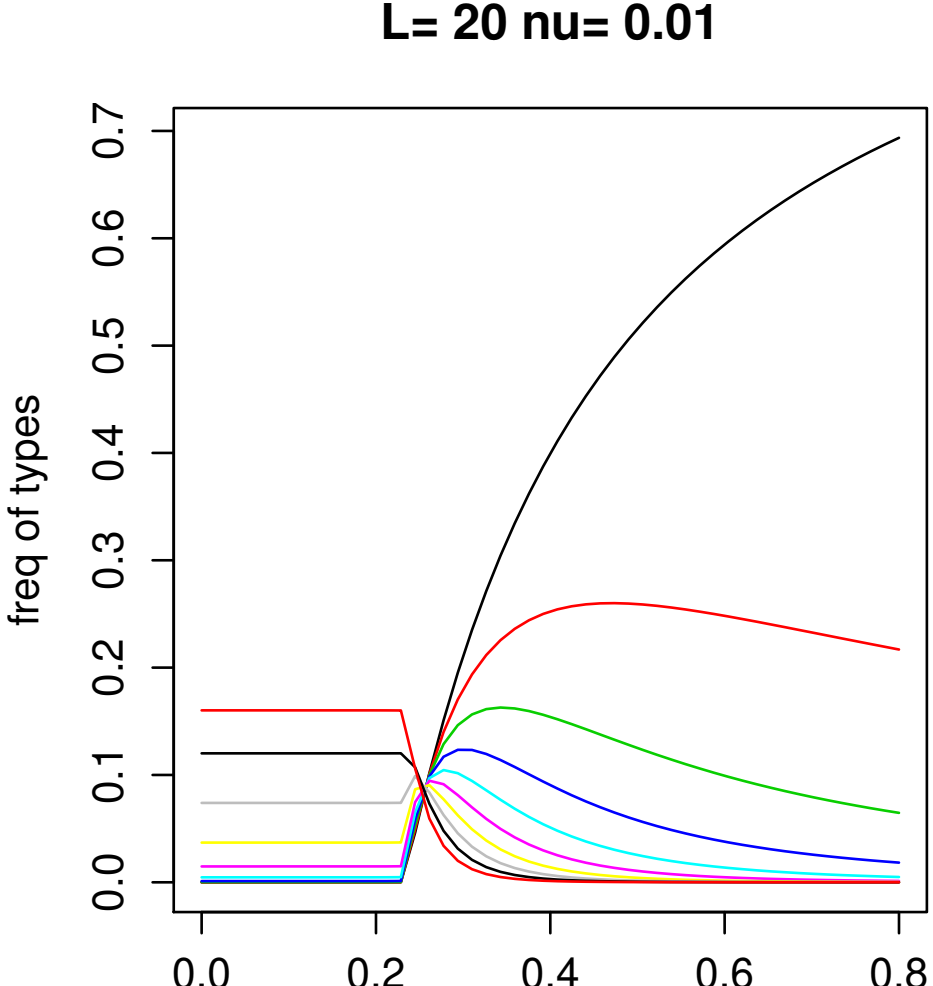
```
>round( max.eigen(W %*% M)$vector,2 )
```

```
round( max.eigen(W %*% M)$vector,2 )
```

	freq
i=0	0.01
i=1	0.02
i=2	0.05
i=3	0.12
i=4	0.21
i=5	0.24
i=6	0.2
i=7	0.11
i=8	0.04
i=9	0.01
i=10	0

```
>
```


s= 0.3
s= 0.2
s= 0.1
s= 0.09
s= 0.07


L= 20 nu= 0.01
freq of types
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
0.2
0.4
0.6
0.8

$$p_{i \to i+1} = (L - i)\nu$$

$$p_{i \to i-1} = i\nu$$

so:

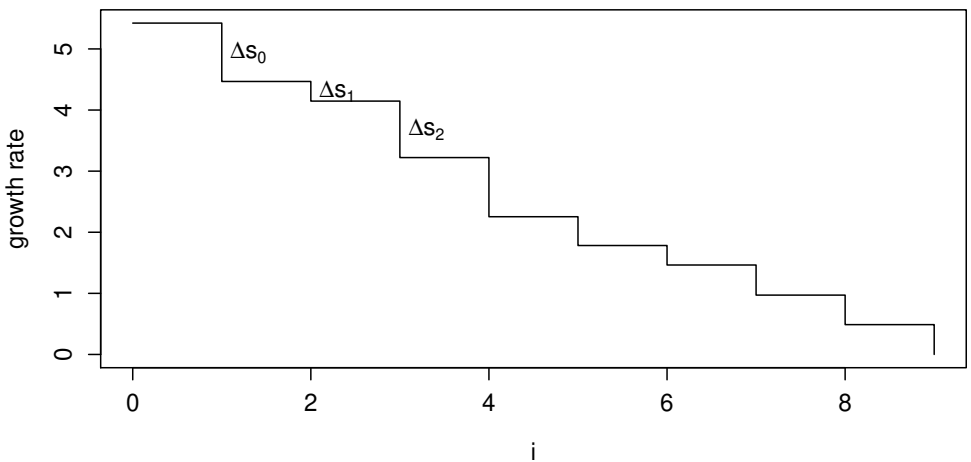
$$p_{0 \to 1} = L\nu$$

Where is the error catastrophe?

since $\mu = L\nu$

$$s \gtrsim L\nu$$

$$p_{i \to i+1} = (L - i)\nu$$

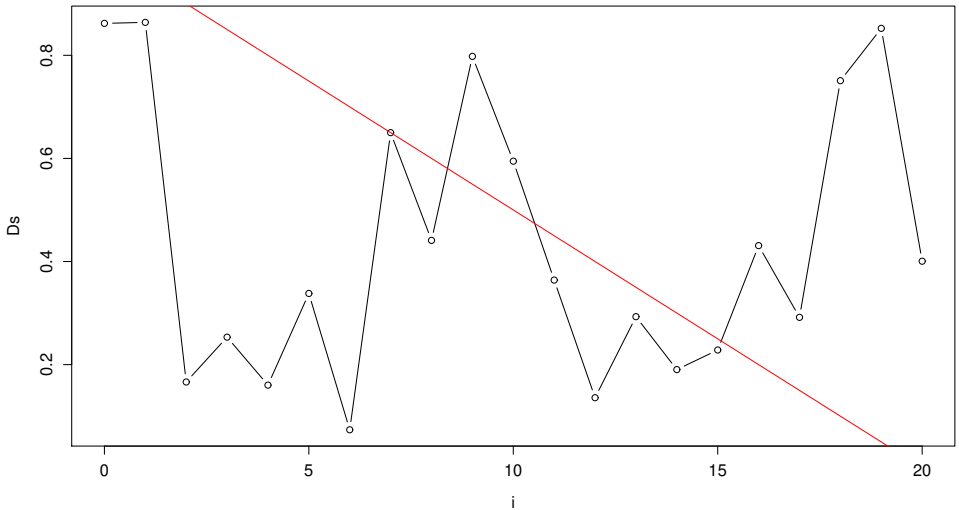


If change in growth from i to $i+1$ is Δs_i

Population will be where $\Delta s_i > (L - i)\nu$

```
> L=20; Ds=runif(L+1);  nu=0.05; i=0:L
  plot( i, Ds,type="b"); lines(i, (L-i)*nu,col=2);v()
```


```
L=20; Ds=runif(L+1);  nu=0.05; i=0:L
> plot( i, Ds,type="b"); lines(i, (L-i)*nu,col=2);v()
```


```
>
```