

# Continuous State Markov Chains

Jesse Perla, Thomas J. Sargent and John Stachurski

December 4, 2020

## 1 Contents

- Overview [2](#)
- The Density Case [3](#)
- Beyond Densities [4](#)
- Stability [5](#)
- Exercises [6](#)
- Solutions [7](#)
- Appendix [8](#)

## 2 Overview

In a [previous lecture](#) we learned about finite Markov chains, a relatively elementary class of stochastic dynamic models.

The present lecture extends this analysis to continuous (i.e., uncountable) state Markov chains.

Most stochastic dynamic models studied by economists either fit directly into this class or can be represented as continuous state Markov chains after minor modifications.

In this lecture, our focus will be on continuous Markov models that

- evolve in discrete time
- are often nonlinear

The fact that we accommodate nonlinear models here is significant, because linear stochastic models have their own highly developed tool set, as we'll see [later on](#).

The question that interests us most is: Given a particular stochastic dynamic model, how will the state of the system evolve over time?

In particular,

- What happens to the distribution of the state variables?
- Is there anything we can say about the “average behavior” of these variables?
- Is there a notion of “steady state” or “long run equilibrium” that’s applicable to the model?
  - If so, how can we compute it?

Answering these questions will lead us to revisit many of the topics that occupied us in the finite state case, such as simulation, distribution dynamics, stability, ergodicity, etc.

### Note

For some people, the term “Markov chain” always refers to a process with a finite or discrete state space. We follow the mainstream mathematical literature (e.g., [4]) in using the term to refer to any discrete **time** Markov process.

## 2.1 Setup

```
In [1]: using InstantiateFromURL
        # optionally add arguments to force installation: instantiate = true,
        ↪precompile = true
        github_project("QuantEcon/quantecon-notebooks-julia", version = "0.8.0")
```

```
In [2]: using LinearAlgebra, Statistics
        using KernelDensity, Distributions, Plots, QuantEcon, Random
```

## 3 The Density Case

You are probably aware that some distributions can be represented by densities and some cannot.

(For example, distributions on the real numbers  $\mathbb{R}$  that put positive probability on individual points have no density representation)

We are going to start our analysis by looking at Markov chains where the one step transition probabilities have density representations.

The benefit is that the density case offers a very direct parallel to the finite case in terms of notation and intuition.

Once we’ve built some intuition we’ll cover the general case.

### 3.1 Definitions and Basic Properties

In our [lecture on finite Markov chains](#), we studied discrete time Markov chains that evolve on a finite state space  $S$ .

In this setting, the dynamics of the model are described by a stochastic matrix — a nonnegative square matrix  $P = P[i, j]$  such that each row  $P[i, \cdot]$  sums to one.

The interpretation of  $P$  is that  $P[i, j]$  represents the probability of transitioning from state  $i$  to state  $j$  in one unit of time.

In symbols,

$$\mathbb{P}\{X_{t+1} = j \mid X_t = i\} = P[i, j]$$

Equivalently,

- $P$  can be thought of as a family of distributions  $P[i, \cdot]$ , one for each  $i \in S$

- $P[i, \cdot]$  is the distribution of  $X_{t+1}$  given  $X_t = i$

(As you probably recall, when using Julia arrays,  $P[i, \cdot]$  is expressed as  $\mathbf{P}[\mathbf{i}, :]$ )

In this section, we'll allow  $S$  to be a subset of  $\mathbb{R}$ , such as

- $\mathbb{R}$  itself
- the positive reals  $(0, \infty)$
- a bounded interval  $(a, b)$

The family of discrete distributions  $P[i, \cdot]$  will be replaced by a family of densities  $p(x, \cdot)$ , one for each  $x \in S$ .

Analogous to the finite state case,  $p(x, \cdot)$  is to be understood as the distribution (density) of  $X_{t+1}$  given  $X_t = x$ .

More formally, a *stochastic kernel on  $S$*  is a function  $p: S \times S \rightarrow \mathbb{R}$  with the property that

1.  $p(x, y) \geq 0$  for all  $x, y \in S$
2.  $\int p(x, y) dy = 1$  for all  $x \in S$

(Integrals are over the whole space unless otherwise specified)

For example, let  $S = \mathbb{R}$  and consider the particular stochastic kernel  $p_w$  defined by

$$p_w(x, y) := \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{(y-x)^2}{2} \right\} \quad (1)$$

What kind of model does  $p_w$  represent?

The answer is, the (normally distributed) random walk

$$X_{t+1} = X_t + \xi_{t+1} \quad \text{where} \quad \{\xi_t\} \stackrel{\text{i.i.d.}}{\sim} N(0, 1) \quad (2)$$

To see this, let's find the stochastic kernel  $p$  corresponding to (2).

Recall that  $p(x, \cdot)$  represents the distribution of  $X_{t+1}$  given  $X_t = x$ .

Letting  $X_t = x$  in (2) and considering the distribution of  $X_{t+1}$ , we see that  $p(x, \cdot) = N(x, 1)$ .

In other words,  $p$  is exactly  $p_w$ , as defined in (1).

### 3.2 Connection to Stochastic Difference Equations

In the previous section, we made the connection between stochastic difference equation (2) and stochastic kernel (1).

In economics and time series analysis we meet stochastic difference equations of all different shapes and sizes.

It will be useful for us if we have some systematic methods for converting stochastic difference equations into stochastic kernels.

To this end, consider the generic (scalar) stochastic difference equation given by

$$X_{t+1} = \mu(X_t) + \sigma(X_t) \xi_{t+1} \quad (3)$$

Here we assume that

- $\{\xi_t\} \stackrel{\text{IID}}{\sim} \phi$ , where  $\phi$  is a given density on  $\mathbb{R}$
- $\mu$  and  $\sigma$  are given functions on  $S$ , with  $\sigma(x) > 0$  for all  $x$

**Example 1:** The random walk (2) is a special case of (3), with  $\mu(x) = x$  and  $\sigma(x) = 1$ .

**Example 2:** Consider the [ARCH model](#)

$$X_{t+1} = \alpha X_t + \sigma_t \xi_{t+1}, \quad \sigma_t^2 = \beta + \gamma X_t^2, \quad \beta, \gamma > 0$$

Alternatively, we can write the model as

$$X_{t+1} = \alpha X_t + (\beta + \gamma X_t^2)^{1/2} \xi_{t+1} \quad (4)$$

This is a special case of (3) with  $\mu(x) = \alpha x$  and  $\sigma(x) = (\beta + \gamma x^2)^{1/2}$ .

**Example 3:** With stochastic production and a constant savings rate, the one-sector neoclassical growth model leads to a law of motion for capital per worker such as

$$k_{t+1} = s A_{t+1} f(k_t) + (1 - \delta) k_t \quad (5)$$

Here

- $s$  is the rate of savings
- $A_{t+1}$  is a production shock
  - The  $t + 1$  subscript indicates that  $A_{t+1}$  is not visible at time  $t$
- $\delta$  is a depreciation rate
- $f: \mathbb{R}_+ \rightarrow \mathbb{R}_+$  is a production function satisfying  $f(k) > 0$  whenever  $k > 0$

(The fixed savings rate can be rationalized as the optimal policy for a particular set of technologies and preferences (see [3], section 3.1.2), although we omit the details here)

Equation (5) is a special case of (3) with  $\mu(x) = (1 - \delta)x$  and  $\sigma(x) = s f(x)$ .

Now let's obtain the stochastic kernel corresponding to the generic model (3).

To find it, note first that if  $U$  is a random variable with density  $f_U$ , and  $V = a + bU$  for some constants  $a, b$  with  $b > 0$ , then the density of  $V$  is given by

$$f_V(v) = \frac{1}{b} f_U\left(\frac{v - a}{b}\right) \quad (6)$$

(The proof is [below](#). For a multidimensional version see [EDTC](#), theorem 8.1.3)

Taking (6) as given for the moment, we can obtain the stochastic kernel  $p$  for (3) by recalling that  $p(x, \cdot)$  is the conditional density of  $X_{t+1}$  given  $X_t = x$ .

In the present case, this is equivalent to stating that  $p(x, \cdot)$  is the density of  $Y := \mu(x) + \sigma(x) \xi_{t+1}$  when  $\xi_{t+1} \sim \phi$ .

Hence, by (6),

$$p(x, y) = \frac{1}{\sigma(x)} \phi\left(\frac{y - \mu(x)}{\sigma(x)}\right) \quad (7)$$

For example, the growth model in (5) has stochastic kernel

$$p(x, y) = \frac{1}{sf(x)} \phi \left( \frac{y - (1 - \delta)x}{sf(x)} \right) \quad (8)$$

where  $\phi$  is the density of  $A_{t+1}$ .

(Regarding the state space  $S$  for this model, a natural choice is  $(0, \infty)$  — in which case  $\sigma(x) = sf(x)$  is strictly positive for all  $s$  as required)

### 3.3 Distribution Dynamics

In [this section](#) of our lecture on **finite** Markov chains, we asked the following question: If

1.  $\{X_t\}$  is a Markov chain with stochastic matrix  $P$
2. the distribution of  $X_t$  is known to be  $\psi_t$

then what is the distribution of  $X_{t+1}$ ?

Letting  $\psi_{t+1}$  denote the distribution of  $X_{t+1}$ , the answer [we gave](#) was that

$$\psi_{t+1}[j] = \sum_{i \in S} P[i, j] \psi_t[i]$$

This intuitive equality states that the probability of being at  $j$  tomorrow is the probability of visiting  $i$  today and then going on to  $j$ , summed over all possible  $i$ .

In the density case, we just replace the sum with an integral and probability mass functions with densities, yielding

$$\psi_{t+1}(y) = \int p(x, y) \psi_t(x) dx, \quad \forall y \in S \quad (9)$$

It is convenient to think of this updating process in terms of an operator.

(An operator is just a function, but the term is usually reserved for a function that sends functions into functions)

Let  $\mathcal{D}$  be the set of all densities on  $S$ , and let  $P$  be the operator from  $\mathcal{D}$  to itself that takes density  $\psi$  and sends it into new density  $\psi P$ , where the latter is defined by

$$(\psi P)(y) = \int p(x, y) \psi(x) dx \quad (10)$$

This operator is usually called the *Markov operator* corresponding to  $p$

#### Note

Unlike most operators, we write  $P$  to the right of its argument, instead of to the left (i.e.,  $\psi P$  instead of  $P\psi$ ). This is a common convention, with the intention being to maintain the parallel with the finite case — see [here](#).

With this notation, we can write (9) more succinctly as  $\psi_{t+1}(y) = (\psi_t P)(y)$  for all  $y$ , or, dropping the  $y$  and letting “=” indicate equality of functions,

$$\psi_{t+1} = \psi_t P \tag{11}$$

Equation (11) tells us that if we specify a distribution for  $\psi_0$ , then the entire sequence of future distributions can be obtained by iterating with  $P$ .

It’s interesting to note that (11) is a deterministic difference equation.

Thus, by converting a stochastic difference equation such as (3) into a stochastic kernel  $p$  and hence an operator  $P$ , we convert a stochastic difference equation into a deterministic one (albeit in a much higher dimensional space).

### Note

Some people might be aware that discrete Markov chains are in fact a special case of the continuous Markov chains we have just described. The reason is that probability mass functions are densities with respect to the [counting measure](#).

## 3.4 Computation

To learn about the dynamics of a given process, it’s useful to compute and study the sequences of densities generated by the model.

One way to do this is to try to implement the iteration described by (10) and (11) using numerical integration.

However, to produce  $\psi P$  from  $\psi$  via (10), you would need to integrate at every  $y$ , and there is a continuum of such  $y$ .

Another possibility is to discretize the model, but this introduces errors of unknown size.

A nicer alternative in the present setting is to combine simulation with an elegant estimator called the *look ahead* estimator.

Let’s go over the ideas with reference to the growth model [discussed above](#), the dynamics of which we repeat here for convenience:

$$k_{t+1} = sA_{t+1}f(k_t) + (1 - \delta)k_t \tag{12}$$

Our aim is to compute the sequence  $\{\psi_t\}$  associated with this model and fixed initial condition  $\psi_0$ .

To approximate  $\psi_t$  by simulation, recall that, by definition,  $\psi_t$  is the density of  $k_t$  given  $k_0 \sim \psi_0$ .

If we wish to generate observations of this random variable, all we need to do is

1. draw  $k_0$  from the specified initial condition  $\psi_0$
2. draw the shocks  $A_1, \dots, A_t$  from their specified density  $\phi$
3. compute  $k_t$  iteratively via (12)

If we repeat this  $n$  times, we get  $n$  independent observations  $k_t^1, \dots, k_t^n$ .

With these draws in hand, the next step is to generate some kind of representation of their distribution  $\psi_t$ .

A naive approach would be to use a histogram, or perhaps a [smoothed histogram](#) using the `kde` function from [KernelDensity.jl](#).

However, in the present setting there is a much better way to do this, based on the look-ahead estimator.

With this estimator, to construct an estimate of  $\psi_t$ , we actually generate  $n$  observations of  $k_{t-1}$ , rather than  $k_t$ .

Now we take these  $n$  observations  $k_{t-1}^1, \dots, k_{t-1}^n$  and form the estimate

$$\psi_t^n(y) = \frac{1}{n} \sum_{i=1}^n p(k_{t-1}^i, y) \tag{13}$$

where  $p$  is the growth model stochastic kernel in (8).

What is the justification for this slightly surprising estimator?

The idea is that, by the strong [law of large numbers](#),

$$\frac{1}{n} \sum_{i=1}^n p(k_{t-1}^i, y) \rightarrow \mathbb{E}p(k_{t-1}^i, y) = \int p(x, y) \psi_{t-1}(x) dx = \psi_t(y)$$

with probability one as  $n \rightarrow \infty$ .

Here the first equality is by the definition of  $\psi_{t-1}$ , and the second is by (9).

We have just shown that our estimator  $\psi_t^n(y)$  in (13) converges almost surely to  $\psi_t(y)$ , which is just what we want to compute.

In fact much stronger convergence results are true (see, for example, this paper).

### 3.5 Implementation

A function which calls an `LAE` type for estimating densities by this technique can be found in [lae.jl](#).

This function returns the right-hand side of (13) using

- an object of type `LAE` that stores the stochastic kernel and the observations
- the value  $y$  as its second argument

The function is vectorized, in the sense that if `psi` is such an instance and  $\mathbf{y}$  is an array, then the call `psi(y)` acts elementwise.

(This is the reason that we reshaped  $\mathbf{X}$  and  $\mathbf{y}$  inside the type — to make vectorization work)

### 3.6 Example

The following code is example of usage for the stochastic growth model [described above](#)

```
In [3]: using Distributions, StatsPlots, Plots, QuantEcon, Random
gr(fmt = :png)
Random.seed!(42) # For deterministic results.

s = 0.2
δ = 0.1
a_σ = 0.4 # A = exp(B) where B ~ N(0, a_σ)
α = 0.4 # We set f(k) = k**α
ψ_0 = Beta(5.0, 5.0) # Initial distribution
φ = LogNormal(0.0, a_σ)

function p(x, y)
    # Stochastic kernel for the growth model with Cobb-Douglas production.
    # Both x and y must be strictly positive.

    d = s * x.^α

    pdf_arg = clamp((y .- (1-δ) .* x) ./ d, eps(), Inf)
    return pdf.(φ, pdf_arg) ./ d
end

n = 10000 # Number of observations at each date t
T = 30 # Compute density of k_t at 1, ..., T+1

# Generate matrix s.t. t-th column is n observations of k_t
k = zeros(n, T)
A = rand!(φ, zeros(n, T))

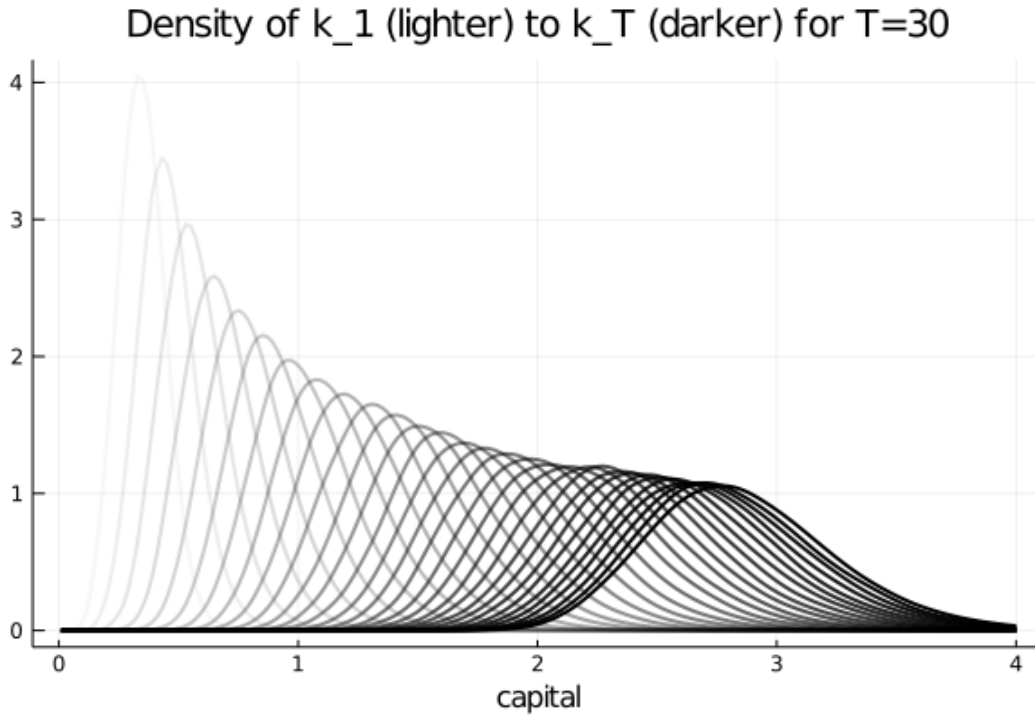
# Draw first column from initial distribution
k[:, 1] = rand(ψ_0, n) ./ 2 # divide by 2 to match scale = 0.5 in py version
for t in 1:T-1
    k[:, t+1] = s*A[:, t] .* k[:, t].^α + (1-δ) .* k[:, t]
end

# Generate T instances of LAE using this data, one for each date t
laes = [LAE(p, k[:, t]) for t in 1:T]

# Plot
ygrid = range(0.01, 4, length = 200)
laes_plot = []
colors = []
for i in 1:T
    ψ = laes[i]
    push!(laes_plot, lae_est(ψ, ygrid))
    push!(colors, RGBA(0, 0, 0, 1 - (i - 1)/T))
end
plot(ygrid, laes_plot, color = reshape(colors, 1, length(colors)), lw = 2,
      xlabel = "capital", legend = :none)
t = "Density of k_1 (lighter) to k_T (darker) for T=$T"
plot!(title = t)
```

Out[3]:





The figure shows part of the density sequence  $\{\psi_t\}$ , with each density computed via the look ahead estimator.

Notice that the sequence of densities shown in the figure seems to be converging — more on this in just a moment.

Another quick comment is that each of these distributions could be interpreted as a cross sectional distribution (recall [this discussion](#)).

## 4 Beyond Densities

Up until now, we have focused exclusively on continuous state Markov chains where all conditional distributions  $p(x, \cdot)$  are densities.

As discussed above, not all distributions can be represented as densities.

If the conditional distribution of  $X_{t+1}$  given  $X_t = x$  **cannot** be represented as a density for some  $x \in S$ , then we need a slightly different theory.

The ultimate option is to switch from densities to [probability measures](#), but not all readers will be familiar with measure theory.

We can, however, construct a fairly general theory using distribution functions.

### 4.1 Example and Definitions

To illustrate the issues, recall that Hopenhayn and Rogerson [1] study a model of firm dynamics where individual firm productivity follows the exogenous process

$$X_{t+1} = a + \rho X_t + \xi_{t+1}, \quad \text{where } \{\xi_t\} \stackrel{\text{iid}}{\sim} N(0, \sigma^2)$$

As is, this fits into the density case we treated above.

However, the authors wanted this process to take values in  $[0, 1]$ , so they added boundaries at the end points 0 and 1.

One way to write this is

$$X_{t+1} = h(a + \rho X_t + \xi_{t+1}) \quad \text{where} \quad h(x) := x \mathbf{1}\{0 \leq x \leq 1\} + \mathbf{1}\{x > 1\}$$

If you think about it, you will see that for any given  $x \in [0, 1]$ , the conditional distribution of  $X_{t+1}$  given  $X_t = x$  puts positive probability mass on 0 and 1.

Hence it cannot be represented as a density.

What we can do instead is use cumulative distribution functions (cdfs).

To this end, set

$$G(x, y) := \mathbb{P}\{h(a + \rho x + \xi_{t+1}) \leq y\} \quad (0 \leq x, y \leq 1)$$

This family of cdfs  $G(x, \cdot)$  plays a role analogous to the stochastic kernel in the density case.

The distribution dynamics in (9) are then replaced by

$$F_{t+1}(y) = \int G(x, y) F_t(dx) \tag{14}$$

Here  $F_t$  and  $F_{t+1}$  are cdfs representing the distribution of the current state and next period state.

The intuition behind (14) is essentially the same as for (9).

## 4.2 Computation

If you wish to compute these cdfs, you cannot use the look-ahead estimator as before.

Indeed, you should not use any density estimator, since the objects you are estimating/computing are not densities.

One good option is simulation as before, combined with the [empirical distribution function](#).

## 5 Stability

In our [lecture](#) on finite Markov chains we also studied stationarity, stability and ergodicity.

Here we will cover the same topics for the continuous case.

We will, however, treat only the density case (as in [this section](#)), where the stochastic kernel is a family of densities.

The general case is relatively similar — references are given below.

## 5.1 Theoretical Results

Analogous to [the finite case](#), given a stochastic kernel  $p$  and corresponding Markov operator as defined in [\(10\)](#), a density  $\psi^*$  on  $S$  is called *stationary* for  $P$  if it is a fixed point of the operator  $P$ .

In other words,

$$\psi^*(y) = \int p(x, y)\psi^*(x) dx, \quad \forall y \in S \quad (15)$$

As with the finite case, if  $\psi^*$  is stationary for  $P$ , and the distribution of  $X_0$  is  $\psi^*$ , then, in view of [\(11\)](#),  $X_t$  will have this same distribution for all  $t$ .

Hence  $\psi^*$  is the stochastic equivalent of a steady state.

In the finite case, we learned that at least one stationary distribution exists, although there may be many.

When the state space is infinite, the situation is more complicated.

Even existence can fail very easily.

For example, the random walk model has no stationary density (see, e.g., [EDTC](#), p. 210).

However, there are well-known conditions under which a stationary density  $\psi^*$  exists.

With additional conditions, we can also get a unique stationary density ( $\psi \in \mathcal{D}$  and  $\psi = \psi P \implies \psi = \psi^*$ ), and also global convergence in the sense that

$$\forall \psi \in \mathcal{D}, \quad \psi P^t \rightarrow \psi^* \quad \text{as } t \rightarrow \infty \quad (16)$$

This combination of existence, uniqueness and global convergence in the sense of [\(16\)](#) is often referred to as *global stability*.

Under very similar conditions, we get *ergodicity*, which means that

$$\frac{1}{n} \sum_{t=1}^n h(X_t) \rightarrow \int h(x)\psi^*(x)dx \quad \text{as } n \rightarrow \infty \quad (17)$$

for any ([measurable](#)) function  $h: S \rightarrow \mathbb{R}$  such that the right-hand side is finite.

Note that the convergence in [\(17\)](#) does not depend on the distribution (or value) of  $X_0$ .

This is actually very important for simulation — it means we can learn about  $\psi^*$  (i.e., approximate the right hand side of [\(17\)](#) via the left hand side) without requiring any special knowledge about what to do with  $X_0$ .

So what are these conditions we require to get global stability and ergodicity?

In essence, it must be the case that

1. Probability mass does not drift off to the “edges” of the state space
2. Sufficient “mixing” obtains

For one such set of conditions see theorem 8.2.14 of [EDTC](#).

In addition

- [\[5\]](#) contains a classic (but slightly outdated) treatment of these topics.
- From the mathematical literature, [\[2\]](#) and [\[4\]](#) give outstanding in depth treatments.
- Section 8.1.2 of [EDTC](#) provides detailed intuition, and section 8.3 gives additional references.
- [EDTC](#), section 11.3.4 provides a specific treatment for the growth model we considered in this lecture.

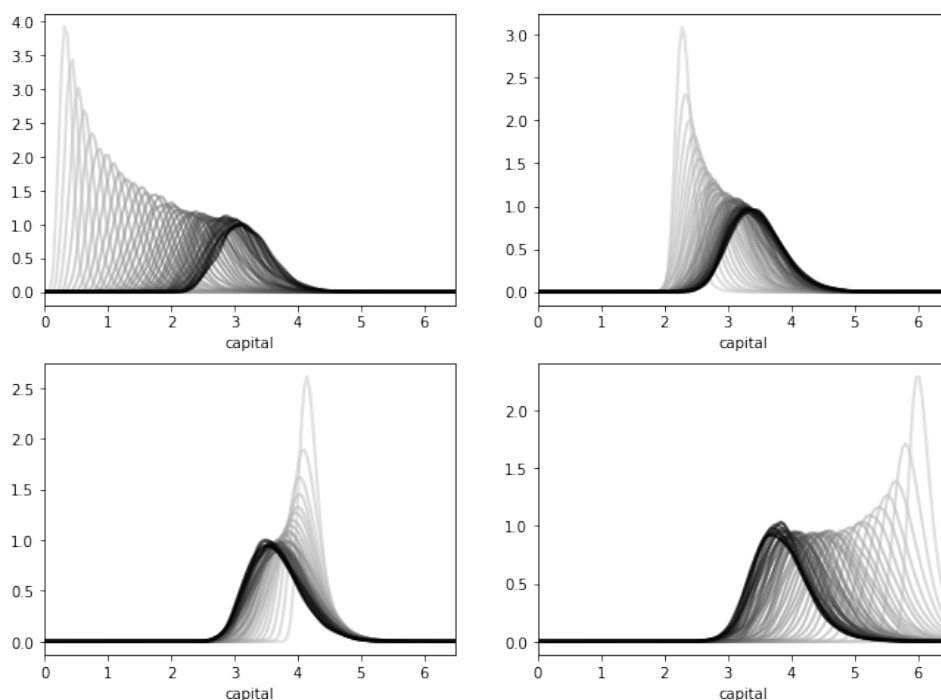
## 5.2 An Example of Stability

As stated above, the [growth model treated here](#) is stable under mild conditions on the primitives.

- See [EDTC](#), section 11.3.4 for more details.

We can see this stability in action — in particular, the convergence in [\(16\)](#) — by simulating the path of densities from various initial conditions.

Here is such a figure



All sequences are converging towards the same limit, regardless of their initial condition.

The details regarding initial conditions and so on are given in [this exercise](#), where you are asked to replicate the figure.

## 5.3 Computing Stationary Densities

In the preceding figure, each sequence of densities is converging towards the unique stationary density  $\psi^*$ .

Even from this figure we can get a fair idea what  $\psi^*$  looks like, and where its mass is located.

However, there is a much more direct way to estimate the stationary density, and it involves only a slight modification of the look ahead estimator.

Let's say that we have a model of the form (3) that is stable and ergodic.

Let  $p$  be the corresponding stochastic kernel, as given in (7).

To approximate the stationary density  $\psi^*$ , we can simply generate a long time series  $X_0, X_1, \dots, X_n$  and estimate  $\psi^*$  via

$$\psi_n^*(y) = \frac{1}{n} \sum_{t=1}^n p(X_t, y) \quad (18)$$

This is essentially the same as the look ahead estimator (13), except that now the observations we generate are a single time series, rather than a cross section.

The justification for (18) is that, with probability one as  $n \rightarrow \infty$ ,

$$\frac{1}{n} \sum_{t=1}^n p(X_t, y) \rightarrow \int p(x, y) \psi^*(x) dx = \psi^*(y)$$

where the convergence is by (17) and the equality on the right is by (15).

The right hand side is exactly what we want to compute.

On top of this asymptotic result, it turns out that the rate of convergence for the look ahead estimator is very good.

The first exercise helps illustrate this point.

## 6 Exercises

### 6.1 Exercise 1

Consider the simple threshold autoregressive model

$$X_{t+1} = \theta |X_t| + (1 - \theta^2)^{1/2} \xi_{t+1} \quad \text{where } \{\xi_t\} \stackrel{\text{i.i.d.}}{\sim} N(0, 1) \quad (19)$$

This is one of those rare nonlinear stochastic models where an analytical expression for the stationary density is available.

In particular, provided that  $|\theta| < 1$ , there is a unique stationary density  $\psi^*$  given by

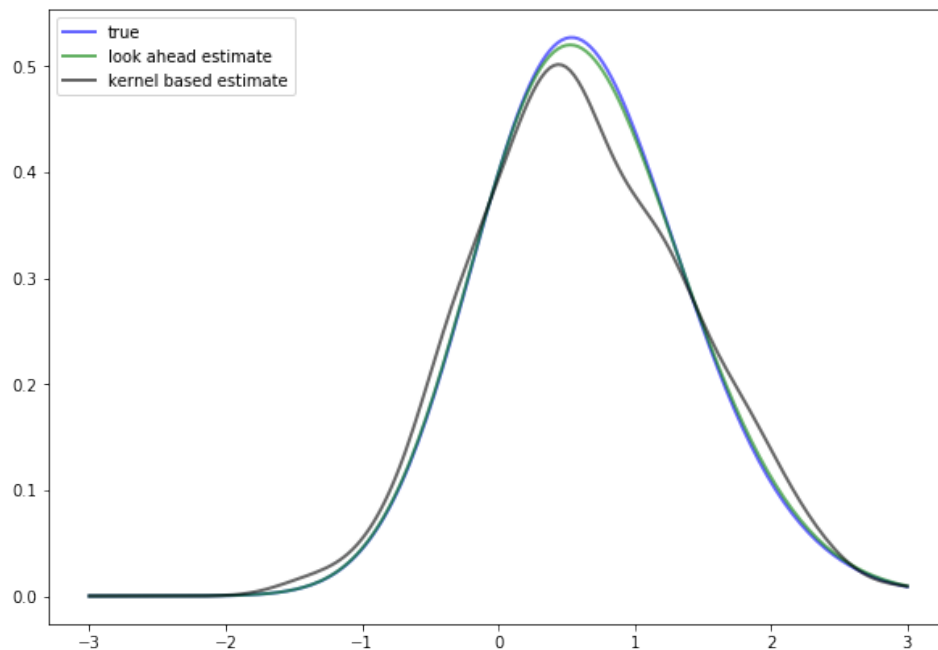
$$\psi^*(y) = 2 \phi(y) \Phi \left[ \frac{\theta y}{(1 - \theta^2)^{1/2}} \right] \quad (20)$$

Here  $\phi$  is the standard normal density and  $\Phi$  is the standard normal cdf.

As an exercise, compute the look ahead estimate of  $\psi^*$ , as defined in (18), and compare it with  $\psi^*$  in (20) to see whether they are indeed close for large  $n$ .

In doing so, set  $\theta = 0.8$  and  $n = 500$ .

The next figure shows the result of such a computation



The additional density (black line) is a [nonparametric kernel density estimate](#), added to the solution for illustration.

(You can try to replicate it before looking at the solution if you want to)

As you can see, the look ahead estimator is a much tighter fit than the kernel density estimator.

If you repeat the simulation you will see that this is consistently the case.

## 6.2 Exercise 2

Replicate the figure on global convergence [shown above](#).

The densities come from the stochastic growth model treated [at the start of the lecture](#).

Begin with the code found in [stochasticgrowth.py](#).

Use the same parameters.

For the four initial distributions, use the beta distribution and shift the random draws as shown below

```
 $\psi_0$  = Beta(5.0, 5.0) # Initial distribution
n = 1000
# .... more setup

for i in 1:4
    # .... some code
    rand_draws = (rand( $\psi_0$ , n) .+ 2.5i) ./ 2
```

### 6.3 Exercise 3

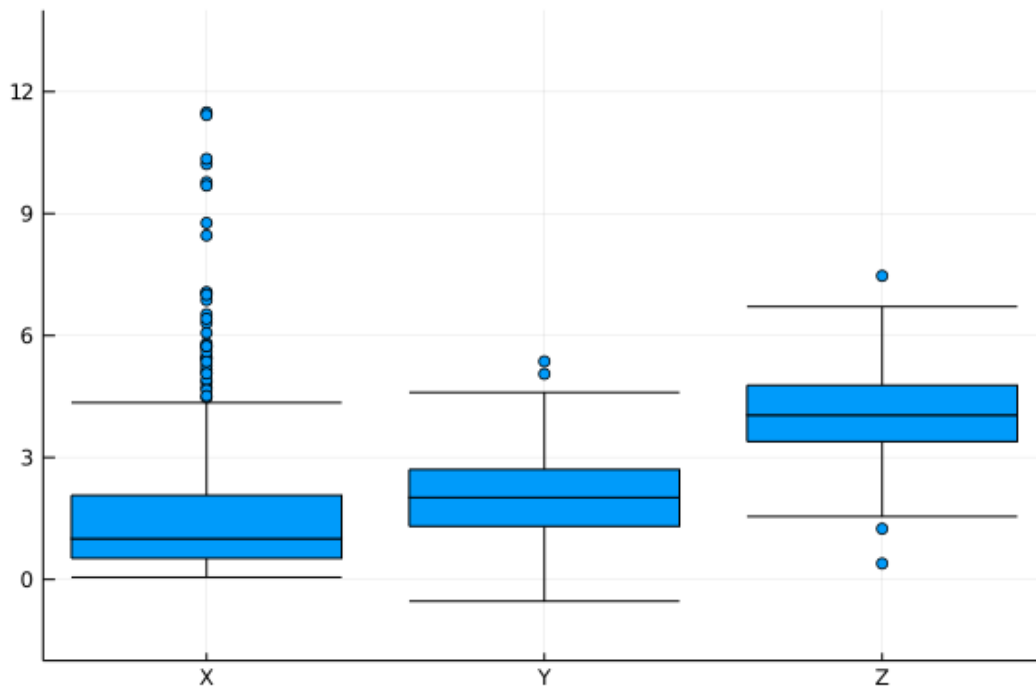
A common way to compare distributions visually is with [boxplots](#).

To illustrate, let's generate three artificial data sets and compare them with a boxplot

```
In [4]: n = 500
x = randn(n)          # N(0, 1)
x = exp.(x)          # Map x to lognormal
y = randn(n) .+ 2.0  # N(2, 1)
z = randn(n) .+ 4.0  # N(4, 1)
data = vcat(x, y, z)
l = ["X" "Y" "Z"]
xlabel = reshape(repeat(l, n), 3n, 1)

boxplot(xlabel, data, label = "", ylims = (-2, 14))
```

Out[4]:



The three data sets are

$$\{X_1, \dots, X_n\} \sim LN(0, 1), \quad \{Y_1, \dots, Y_n\} \sim N(2, 1), \quad \text{and} \quad \{Z_1, \dots, Z_n\} \sim N(4, 1),$$

The figure looks as follows.

Each data set is represented by a box, where the top and bottom of the box are the third and first quartiles of the data, and the red line in the center is the median.

The boxes give some indication as to

- the location of probability mass for each sample
- whether the distribution is right-skewed (as is the lognormal distribution), etc

Now let's put these ideas to use in a simulation.

Consider the threshold autoregressive model in (19).

We know that the distribution of  $X_t$  will converge to (20) whenever  $|\theta| < 1$ .

Let's observe this convergence from different initial conditions using boxplots.

In particular, the exercise is to generate  $J$  boxplot figures, one for each initial condition  $X_0$  in

```
initial_conditions = range(8, 0, length = J)
```

For each  $X_0$  in this set,

1. Generate  $k$  time series of length  $n$ , each starting at  $X_0$  and obeying (19).
2. Create a boxplot representing  $n$  distributions, where the  $t$ -th distribution shows the  $k$  observations of  $X_t$ .

Use  $\theta = 0.9, n = 20, k = 5000, J = 8$ .

## 7 Solutions

In [5]: **using** KernelDensity

### 7.1 Exercise 1

Look ahead estimation of a TAR stationary density, where the TAR model is

$$X_{t+1} = \theta|X_t| + (1 - \theta^2)^{1/2}\xi_{t+1}$$

and  $\xi_t \sim N(0, 1)$ . Try running at  $n = 10, 100, 1000, 10000$  to get an idea of the speed of convergence.

```
In [6]:  $\phi$  = Normal()
n = 500
 $\theta$  = 0.8
d = sqrt(1.0 -  $\theta^2$ )
 $\delta$  =  $\theta$  / d

# true density of TAR model
 $\phi\_star(y)$  = 2 .* pdf.( $\phi$ , y) .* cdf.( $\phi$ ,  $\delta$  * y)

# Stochastic kernel for the TAR model.
p_TAR(x, y) = pdf.( $\phi$ , (y -  $\theta$  .* abs.(x)) ./ d) ./ d

Z = rand( $\phi$ , n)
X = zeros(n)
for t in 1:n-1
    X[t+1] =  $\theta$  * abs(X[t]) + d * Z[t]
end
```



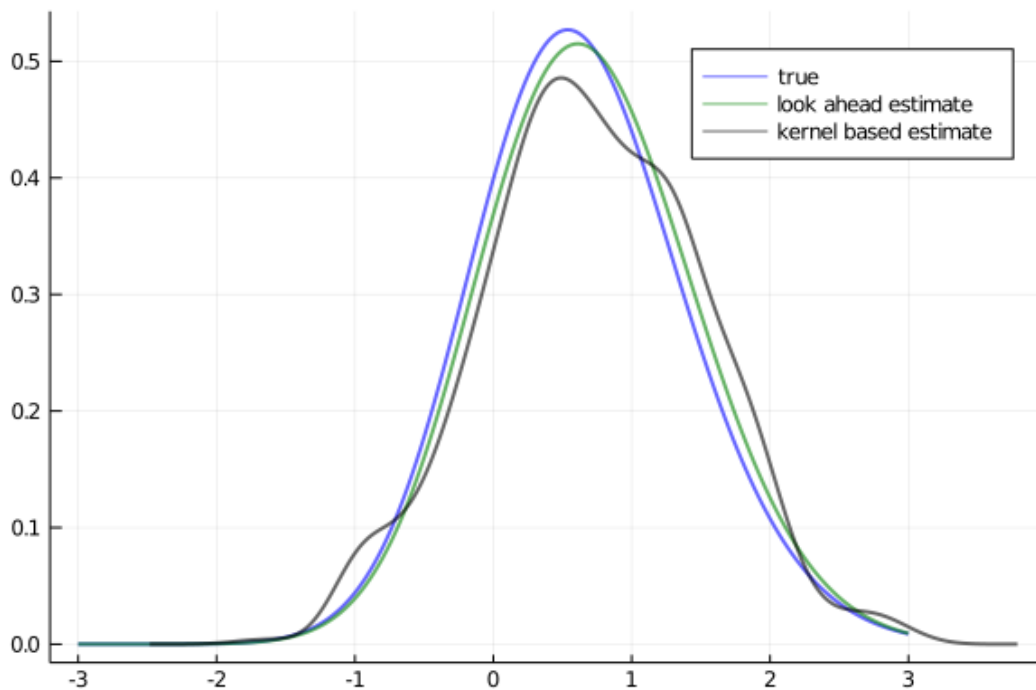
```

ψ_est(a) = lae_est(LAE(p_TAR, X), a)
k_est = kde(X)

ys = range(-3, 3, length = 200)
plot(ys, ψ_star(ys), color=:blue, lw = 2, alpha = 0.6, label = "true")
plot!(ys, ψ_est(ys), color=:green, lw = 2, alpha = 0.6, label = "look
↪ahead estimate")
plot!(k_est.x, k_est.density, color = :black, lw = 2, alpha = 0.6,
      label = "kernel based estimate")

```

Out[6]:



## 7.2 Exercise 2

Here's one program that does the job.

```

In [7]: s = 0.2
        δ = 0.1
        a_σ = 0.4 # A = exp(B) where B ~ N(θ, a_σ)
        α = 0.4   # We set f(k) = k**α
        ψ_0 = Beta(5.0, 5.0) # Initial distribution
        φ = LogNormal(0.0, a_σ)

function p_growth(x, y)
    # Stochastic kernel for the growth model with Cobb-Douglas production.
    # Both x and y must be strictly positive.

    d = s * x.^α

    pdf_arg = clamp.((y .- (1-δ) .* x) ./ d, eps(), Inf)
    return pdf.(φ, pdf_arg) ./ d

```

```

end

n = 1000 # Number of observations at each date t
T = 40   # Compute density of  $k_t$  at  $1, \dots, T+1$ 

xmax = 6.5
ygrid = range(0.01, xmax, length = 150)
laes_plot = zeros(length(ygrid), 4T)
colors = []
for i in 1:4
    k = zeros(n, T)
    A = rand!( $\phi$ , zeros(n, T))

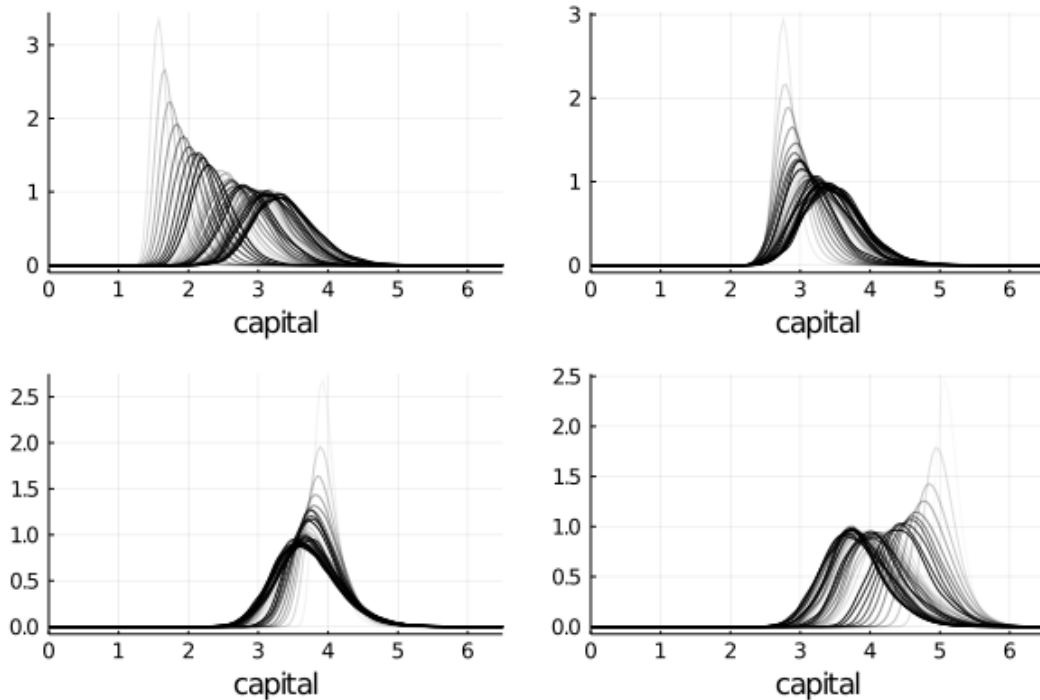
    # Draw first column from initial distribution
    # match scale = 0.5 and loc = 2i in julia version
    k[:, 1] = (rand( $\psi_0$ , n) .+ 2.5i) ./ 2
    for t in 1:T-1
        k[:, t+1] = s * A[:, t] .* k[:, t]. $\alpha$  + (1 -  $\delta$ ) .* k[:, t]
    end

    # Generate T instances of LAE using this data, one for each date t
    laes = [LAE(p_growth, k[:, t]) for t in T:-1:1]
    ind = i
    for j in 1:T
         $\psi$  = laes[j]
        laes_plot[:, ind] = lae_est( $\psi$ , ygrid)
        ind = ind + 4
        push!(colors, RGBA(0, 0, 0, 1 - (j - 1) / T))
    end
end

#colors = reshape(reshape(colors, T, 4)', 4*T, 1)
colors = reshape(colors, 1, length(colors))
plot(ygrid, laes_plot, layout = (2,2), color = colors,
      legend = :none, xlabel = "capital", xlims = (0, xmax))

```

Out[7]:



### 7.3 Exercise 3

Here's a possible solution.

Note the way we use vectorized code to simulate the  $k$  time series for one boxplot all at once.

```
In [8]: n = 20
k = 5000
J = 6

θ = 0.9
d = sqrt(1 - θ^2)
δ = θ / d

initial_conditions = range(8, θ, length = J)

Z = randn(k, n, J)
titles = []
data = []
x_labels = []
for j in 1:J
    title = "time series from t = $(initial_conditions[j])"
    push!(titles, title)

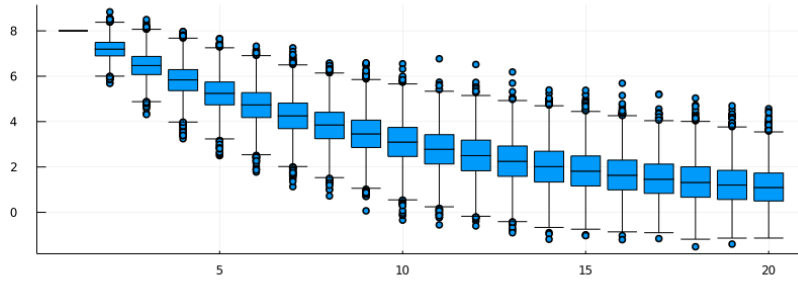
    X = zeros(k, n)
    X[:, 1] .= initial_conditions[j]
    labels = []
    labels = vcat(labels, ones(k, 1))
    for t in 2:n
        X[:, t] = θ .* abs.(X[:, t-1]) .+ d .* Z[:, t, j]
        labels = vcat(labels, t*ones(k, 1))
    end
    X = reshape(X, n*k, 1)
```

```
    push!(data, X)
    push!(x_labels, labels)
end
```

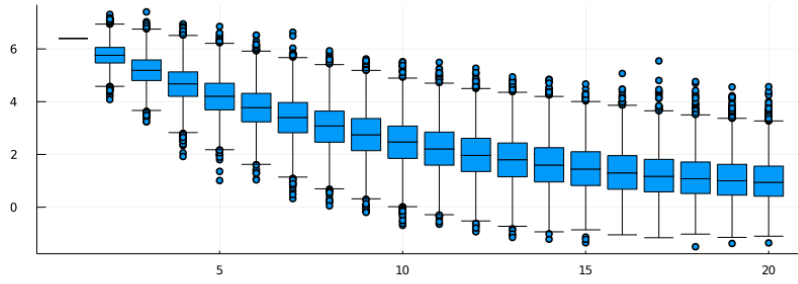
```
In [9]: plots = []
        for i in 1:J
            push!(plots, boxplot(vec(x_labels[i]), vec(data[i]), title = titles[i]))
        end
        plot(plots..., layout = (J, 1), legend = :none, size = (800, 2000))
```

Out[9]:

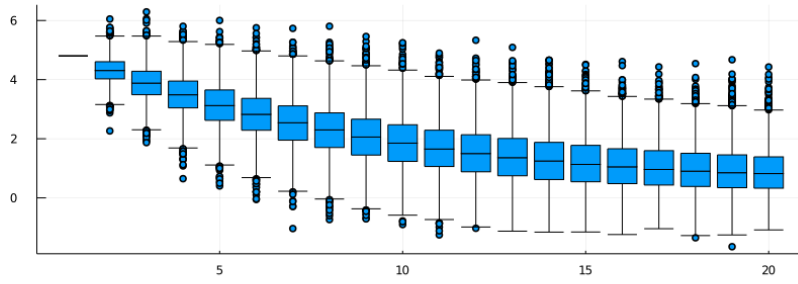
time series from t = 8.0



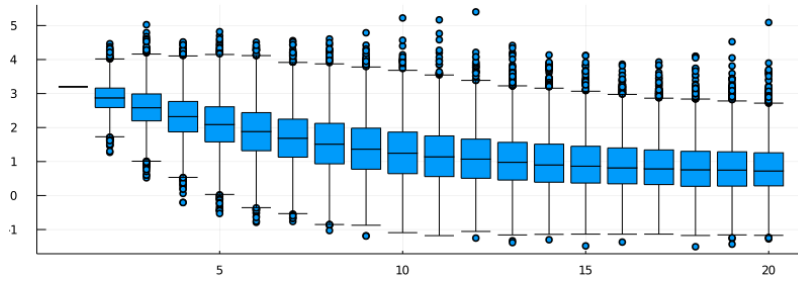
time series from t = 6.4



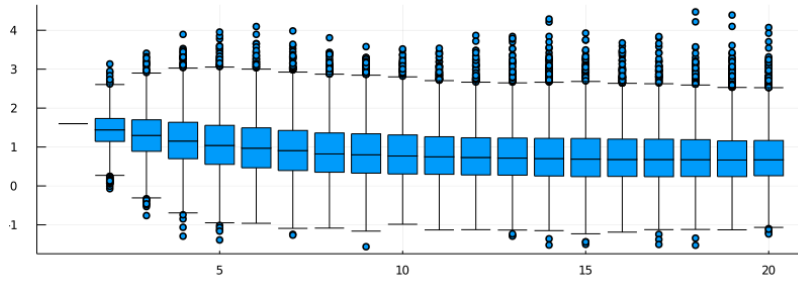
time series from t = 4.8



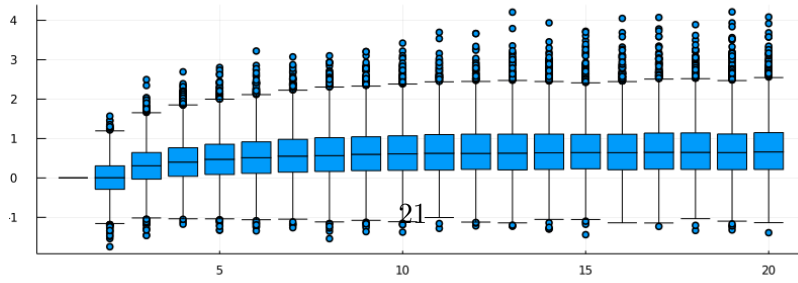
time series from t = 3.2



time series from t = 1.6



time series from t = 0.0



## 8 Appendix

Here's the proof of (6).

Let  $F_U$  and  $F_V$  be the cumulative distributions of  $U$  and  $V$  respectively.

By the definition of  $V$ , we have  $F_V(v) = \mathbb{P}\{a + bU \leq v\} = \mathbb{P}\{U \leq (v - a)/b\}$ .

In other words,  $F_V(v) = F_U((v - a)/b)$ .

Differentiating with respect to  $v$  yields (6).

## References

- [1] Hugo A Hopenhayn and Richard Rogerson. Job Turnover and Policy Evaluation: A General Equilibrium Analysis. *Journal of Political Economy*, 101(5):915–938, 1993.
- [2] A Lasota and M C MacKey. *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*. Applied Mathematical Sciences. Springer-Verlag, 1994.
- [3] L Ljungqvist and T J Sargent. *Recursive Macroeconomic Theory*. MIT Press, 4 edition, 2018.
- [4] S P Meyn and R L Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, 2009.
- [5] N L Stokey, R E Lucas, and E C Prescott. *Recursive Methods in Economic Dynamics*. Harvard University Press, 1989.