Project I: Random Processes

Random variables, stationarity & ergodicity

Simon Andreas Bjørn

February 7, 2023

» 1. Random variables

Importing some statistic and plotting modules

```julia
using Printf
using Statistics
using NaNStatistics
using Plots
```

» 1.1 Random variables - uniform PDF

We start by sampling the uniform destribution on $[0, 1)$, with $N = 10000$ samples.

```julia
N = 10_000
x = rand(N)
```

We then find the first order momentum $m_x = E\{x\}$ and the second order central moment $\sigma_x^2 = E\{(x - m_x)^2\}$

```julia
# Evaluate E{x} and E{(x-mₓ)²} by definition
mₓ   = sum(x*1/N)
# -> 0.49863643598
σₓ² = sum((x.-mₓ).^2 * 1/N)
# -> 0.08423047222

# Evaluate E{x} and E{(x-mₓ)²} by builtin functions
mₓ_builtin = mean(x)
σₓ²_builtin = std(x)^2
```

» Calculation differences

Calculating the difference between the builtin functions and our direct evaluation, we see that the the two methods are similar.

```julia
@printf "mₓ_err: %.4f" abs(mₓ-mₓ_builtin)
# -> mₓ_err: 0.0000
@printf "σₓ²_err: %.4f" abs(σₓ²-σₓ²_builtin)
# -> σₓ²_err: 0.0000
```

» Calculating the PDF

We then calculate the pdf using `histcounts`, and normalize the PDF.

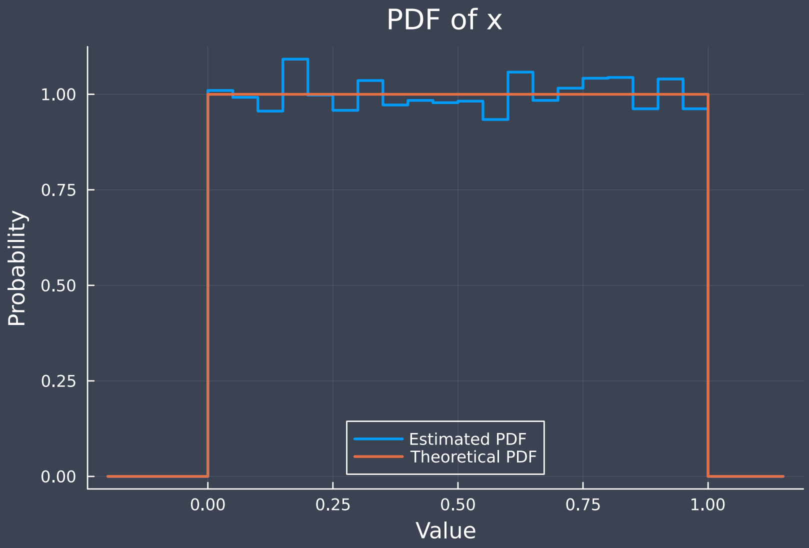
```
step_size = 0.05
edges = -0.2:step_size:1.2
bins = histcounts(x, edges)

# Normalize bins to get PDF
pdf = bins/sum(bins*step_size)

# Theoretical PDF
theoretical_pdf = (x) -> Float64(0 <= x < 1)
```

» Code for plotting the PDF

And we plot the result

```julia
plot(
    edges[1:end-1],
    pdf,
    seriestype=:steppost,
    label="Estimated PDF"
)
    plot!(
    edges[1:end-1],
    theoretical_pdf.(edges[1:end-1]),
    seriestype=:steppost, label="Theoretical PDF"
)
    plot!(
    legend=:bottom,
    background_color=:transparent,
    foreground_color=:white
)
title!("PDF of x");xlabel!("Value");
ylabel!("Probability");savefig("uniform_pdf.svg")
```

» Plot of the PDF



» Ensuring properties

We want to verify if our PDF fulfill the 2 following properties:

1. $\forall\alpha,\ f_x(\alpha) \geq 0$

   ```
   reduce(&, pdf .>= 0) # -> true
   ```

2. $\int_{-\infty}^{\infty} f_x(\alpha)\, \mathrm{d}\alpha = 1$

   ```
   abs(sum(pdf*step_size)-1.0) < 1e-12 # -> true
   ```

» Deriving the theoretical mean [m_x]

We start at the definition of the mean value which is given by

$$m_x = E\{x\} = \int_{-\infty}^{\infty} \alpha f_x(\alpha) \mathsf{d}\alpha \tag{1}$$

We then use the theoretical probability density function of a uniform distribution which is given by

$$f_x(\alpha) = \begin{cases} 0, & \alpha < a \\ \frac{1}{b-a}, & a \leq \alpha \leq b \\ 0, & \alpha > b \end{cases}$$

» Deriving the theoretical mean [m_x]

With $a = 0, b = 1$, we get the theoretical PDF

$$f_x(\alpha) = \begin{cases} 0, & \alpha < 0 \\ 1, & 0 \leq \alpha \leq 1 \\ 0, & \alpha > 1 \end{cases}$$

We simplify eq.1, and get the integral

$$m_x = E\{x\} = \int_0^1 \alpha \; \mathsf{d}\alpha$$

We simply solve this and get the mean value to be

$$m_x = \frac{1}{2}\left[\alpha^2\right]_0^1 = \frac{1}{2}$$

» Deriving the theoretical variance [σ_x^2]

Again, we start at the definition

$$\sigma_x^2 = E\left\{(x - m_x)^2\right\} = \int_{-\infty}^{\infty} (\alpha - m_x) f_x(\alpha) \mathsf{d}\alpha \tag{2}$$

We do the same simplification as with the mean value, and and fill in for m_x. This gives the variance

$$\begin{aligned} \sigma_x^2 &= \int_0^1 \left(\alpha - \frac{1}{2}\right)^2 \ \mathsf{d}\alpha \\ &= \int_0^1 \alpha^2 - \alpha + \frac{1}{4} \ \mathsf{d}\alpha \\ &= \left[\frac{1}{3}\alpha^3 - \frac{1}{2}\alpha^2 + \frac{1}{4}\alpha\right]_0^1 \\ &= \frac{1}{12} \approx 0.08334 \end{aligned}$$

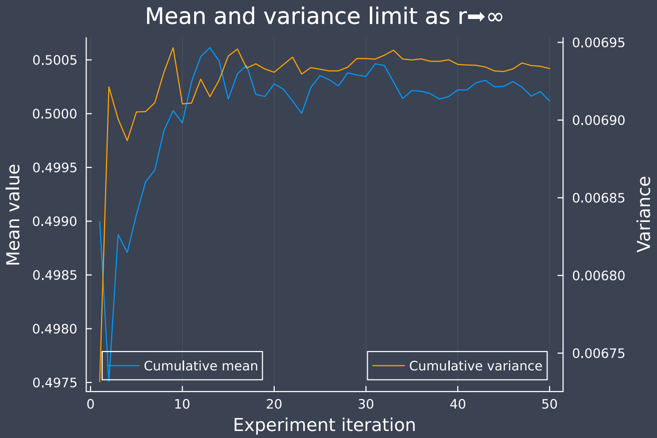
» Repeating the uniform experiment

We repeat the experiment, find the mean and variance of each experiment and accumulate the results.

```julia
r = 50 # Repeat r times
xs = rand(Float64, (N,r))
ms = mean(xs, dims=1)
σs = std(xs, dims=1).^2
# Accumulate the results
i_mean = [mean(ms[1:i]) for i in 1:length(ms)]
i_std = [mean(σs[1:i])^2 for i in 1:length(s)]
# Plot the results
plot(1:r, i_mean, label="Cumulative mean",
    legend=:bottomleft, xlabel="Experiment iteration",
    ylabel="Mean value")
plot!( twinx(), i_std, label="Variance",
    legend=:bottomright, color=:orange, ylabel="Variance")
plot!(foreground_color=:white,
    background_color=:transparent)
savefig!("repeated_experiment_uniform.svg")
```

» Repeating the uniform experiment

We can see that the mean and variance approach some values as we perform more experiments



» 1.2 Gaussian PDF

We perform the same experiment, but sample using `randn`.

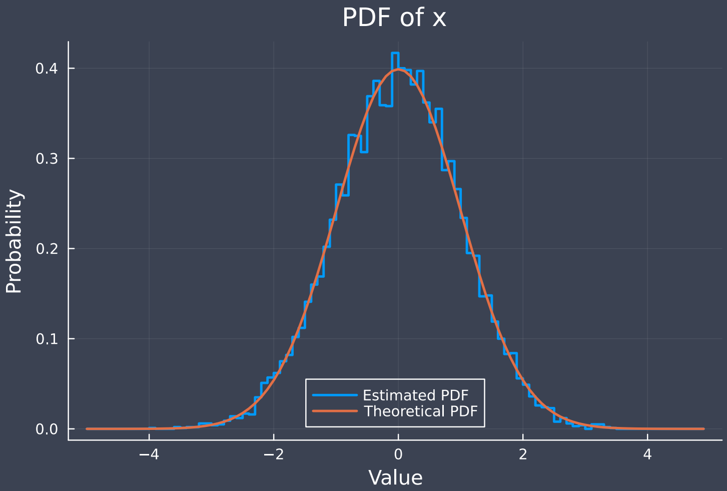
```julia
N = 10_000
x = randn(N)
mₓ  = sum(x*1/N)
σₓ² = sum((x.-mₓ).^2 * 1/N)

step_size = 0.1
edges = -5.0:step_size:5.0
bins = histcounts(x, edges)
# Normalize PDF
pdf = bins/sum(bins*step_size)
theoretical_pdf = (x) -> 1.0/√(2π)*exp(-x.^2 ./ 2)
```

» 1.2 Plotting code

We then plot the results in a similar manner as 1.1.

```julia
plot( edges[1:end-1], pdf, seriestype=:steppost,
    label="Estimated PDF", linewidth=2)

plot!(edges[1:end-1], theoretical_pdf.(edges[1:end-1]),
    label="Theoretical PDF", linewidth=2)

plot!(legend=:bottom, background_color=:transparent,
    foreground_color=:white)

title!("PDF of x");xlabel!("Value");
ylabel!("Probability");
```

» 1.2 Gaussian PDF plot



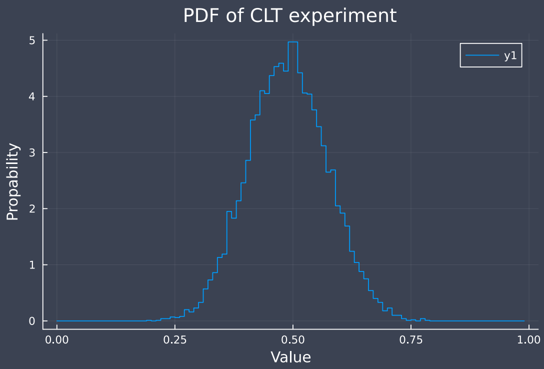
» 1.3 Central Limit Theorem

Again, we sample `rand` as according to the task. We have have the following code.

```julia
N = 10_000
x = rand(Float64, (12, N))
ms = mean(x, dims=1)

step_size = 0.01
edges = 0.0:step_size:1.0
bins = histcounts(ms, edges)

pdf = bins/sum(bins*step_size)
sum(pdf*step_size)

mₓ = mean(ms)
σₓ = std(ms)^2

@printf "mₓ: %.3f" mₓ  # -> mₓ: 0.499
@printf "σₓ²: %.3f" σₓ # -> σₓ²: 0.007
```

We plot the estimated PDF with the following code

```julia
plot(edges[1:end-1], pdf, seriestype=:steppre)
plot!(background_color=:transparent,foreground_color=:white)
xlabel!("Value"); ylabel!("Propability")
title!("PDF of CLT experiment")
savefig("clt_pdf.svg")
```



» Theoretical mean of CTF experiment [m_z]

We start by noting the random variable for our experiment can be defined as $Z = \frac{1}{12}\sum_{i=1}^{12} X_i$.

Then

$$E\left\{Z\right\} = E\left\{\frac{1}{12}\sum_{i=1}^{12} X_i\right\} = \frac{1}{12}E\left\{\sum_{i=1}^{12} X_i\right\} = \frac{1}{12}\sum_{i=1}^{12} E\left\{X_i\right\}$$

As we know that $E\left\{X_i\right\} = \frac{1}{2}$, we have

$$m_z = E\left\{Z\right\} = \frac{1}{12}\sum_{i=1}^{12}\frac{1}{2} = \frac{1}{2}$$

» Theoretical variance of CTF experiment [σ_z^2]

We want to find the variance $\sigma_x^2 = E\left\{(Z-\bar{Z})^2\right\}$, and using the definition from the previous slide, we have

$$E\left\{(Z-E\{Z\})^2\right\} = E\left\{\left(\frac{1}{12}\sum_{i=1}^{12} X_i - E\left\{\frac{1}{12}\sum_{i=1}^{12} X_i\right\}\right)^2\right\}$$

We simplify, using

$$E\{X+Y\} = E\{X\} + E\{Y\} \tag{3}$$

and get the following

$$E\left\{(Z-E\{Z\})^2\right\} = E\left\{\left(\frac{1}{12}\sum_{i=1}^{12} X_i - \frac{1}{12}\sum_{i=1}^{12} E\{X_i\}\right)^2\right\}$$

» Theoretical variance of CTF experiment [σ_z^2]

We factorize and combine our sums to get

$$E\left\{(Z - E\{Z\})^2\right\} = \left(\frac{1}{12}\right)^2 E\left\{\left(\sum_{i=1}^{12} X_i - E\{X_i\}\right)^2\right\}$$

Expanding the binomial, we get

$$\begin{aligned} &= \frac{1}{12^2}\sum_{i=1}^{12} E\left\{(X_i - E\{X_i\})^2\right\} \\ &+ 2\sum_{\substack{i=1\\ j=2\\ i<j}}^{j\to 12} E\left\{(X_i - E\{X_i\})(X_j - E\{X_j\})\right\} \end{aligned}$$

Which almost looks good, but there is an ugly term from the binomial expansion which ruins the fun.

» Theoretical variance of CTF experiment [σ_z^2]

There is luckely an easy clean-up for this. We know that our separate observations X_i are all independent. This means that each $(X_i - E\{X_i\})$ for $i = 1, 2, \ldots, 12$ are independent. We can then use the multiplicative property

Multiplicative Property

For two independent variables X and Y, the expectation of their product is equal to the product of the expectation of each variable.

$$E\{XY\} = E\{X\} E\{Y\}$$

This allows us to rewrite the additional binomial term as

$$E\{X_i - E\{X_i\}\} \cdot E\{X_j - E\{X_j\}\}$$

» Theoretical variance of CTF experiment [σ_z^2]

We then utilize e.q.(3) on each factor of the binomial term, and we see that

$$E\{X_i - E\{X_i\}\} = E\{X_i\} - E\{E\{X_i\}\} = 0$$

$$\Downarrow$$

$$2\sum_{\substack{i=1\\ j=2\\ i<j}}^{j\to 12} E\{X_i - E\{X_i\}\}\,E\left\{X_j - E\left\{X_j\right\}\right\} = 0$$

And thus the big bad evil term dissapered.

» Theoretical variance of CTF experiment [σ_z^2]

Now that we know the additional term equates to 0 and we know $E\left\{(X_i - E\{X_i\})^2\right\} = \frac{1}{12}$, we insert into our equation and get values into our equation

$$E\left\{(Z - E\{Z\})^2\right\} = \frac{1}{12^2}\sum_{i=1}^{12} E\left\{(X_i - E\{X_i\})^2\right\}$$

$$\sigma_z^2 = \frac{1}{12^2}\sum_{i=1}^{12}\frac{1}{12} = \frac{1}{144} \approx 0.0069$$

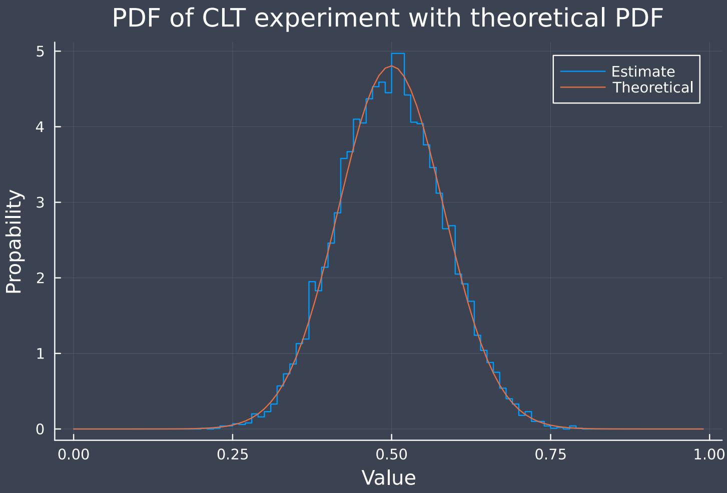
» Comparing the theoretical and estimated values

We got our theoretical values to be $m_z = 0.5,\ \sigma_z^2 = 0.0069$. Comparing this to our result from earlier, we see that our estimated values of $m_z = 0.499,\ \sigma_z^2 = 0.007$ is pretty much spot on.

We then plot a gaussian curve over our previous plot with these values using the following code

```julia
theoretical_pdf = (x) -> 1.0/√(2π*σₓ²)*exp(-(x-mₓ).^2 ./ (2σₓ²))
plot(edges[1:end-1], pdf, seriestype=:steppost, label="Estimate")
plot!(edges[1:end-1], theoretical_pdf.(edges[1:end-1]), label="Theoretical")
plot!(background_color=:transparent,foreground_color=:white)
xlabel!("Value"); ylabel!("Propability")
title!("PDF of CLT experiment")
savefig("clt_pdf_theoretical.svg")
```

» Plot of theoretical and estimated PDF



» 2 - Stationarity and ergodicity

We start by defining the functions as described in the task. The Julia version of the code looks pretty similar to that of MATLAB.

```julia
function rp1(M,N)
    a = 0.02; b = 5;
    Mc = ones(M,1)*b*sin.((1:N)*π/N)'
    Ac = a*ones(M,1)*vec(1:N)';
    return (rand(M,N).-0.5).*Mc + Ac
end
function rp2(M,N)
    Ar = rand(M,1)*ones(1,N);
    Mr = rand(M,1)*ones(1,N);
    return ( rand(M,N) .- 0.5 ).*Mr + Ar;
end
function rp3(M,N)
    a = 0.5; m = 3;
    return ( rand(M,N) .- 0.5 ) .* m .+ a;
end
```

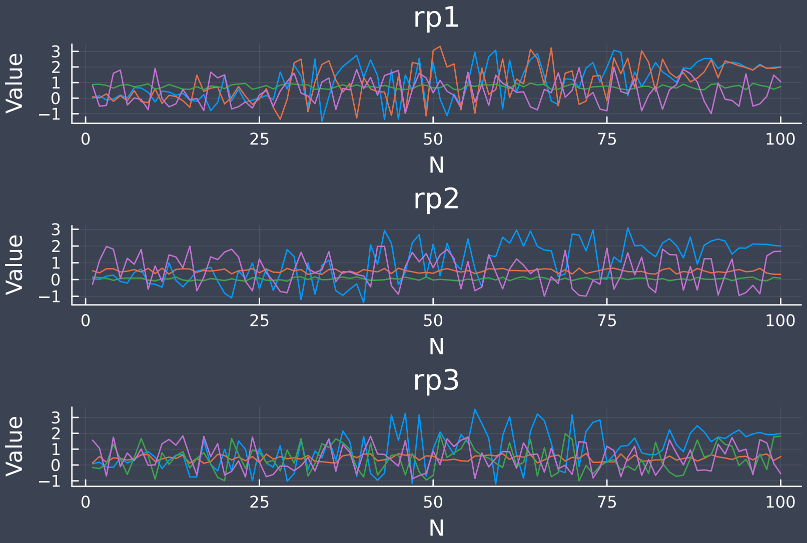
» 2a - Stationarity or ergodic?

We then create the M=4 realizations of each process with N=100 points in time.

```julia
M = 4; N = 100
r1, r2, r3 = rp1(M,N), rp2(M,N), rp3(M,N)
plot(
    [r1', r2', r3'], layout = (3,1),
    title=["rp1" "rp2" "rp3"], legend=false,
    background_color=:transparent,
    foreground_color=:white
)
xlabel!("N"); ylabel!("Value")
```

» 2a - Stationarity or ergodic?

We get the following plot



» 2a - Stationarity or ergodic?

By inspection, we see that

- rp1 tends to increase in value → not stationary
- rp2 looks pretty similar over time → stationary
- rp3 stays pretty much the same → stationary

We then note that

- rp1 is and for early N, will not equal time average → not ergodic
- rp2 seems to have different time average on green and blue realization which would not match ensemble average → not ergodic
- rp3 seems to be quite similar over time and realizations → seems ergodic

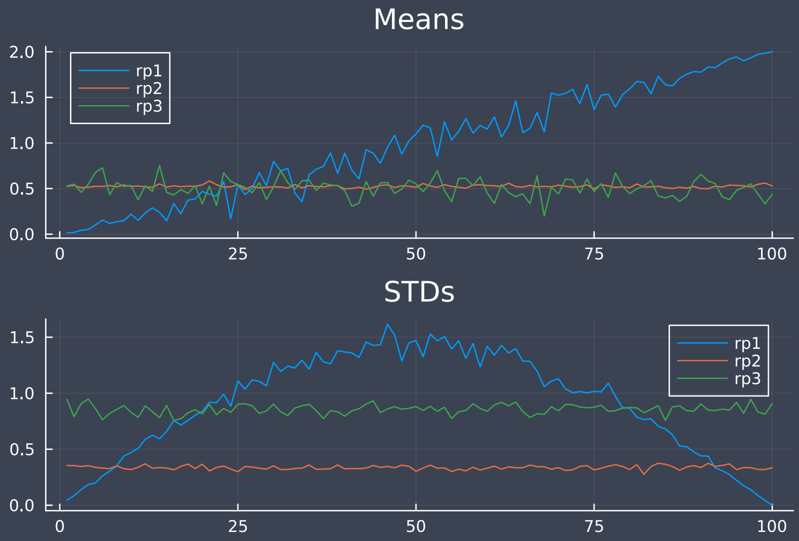
» 2b - Ensemble averages

We compute the ensemble mean and std of each process and plot them with the following code

```julia
M = 80; N = 100
r1, r2, r3 = rp1(M,N), rp2(M,N), rp3(M,N)
l = @layout [means; stds]
p1 = plot(mean(r1, dims=1)', label="rp1")
plot!(mean(r2, dims=1)', label="rp2")
plot!(mean(r3, dims=1)', label="rp3")
p2 = plot(std(r1, dims=1)', label="rp1")
plot!(std(r2, dims=1)', label="rp2")
plot!(std(r3, dims=1)', label="rp3")
plot(p1, p2, layout=l,
    background_color=:transparent,
    foreground_color=:white,
    title=["Means" "STDs"]
)
```

» 2b - Ensemble averages

We then get the following plot



» 2c - Time averages

We estimate the time averages of each process by

```
M = 4; N = 1000
r1, r2, r3 = rp1(M,N), rp2(M,N), rp3(M,N)

@printf "rp1 | Time averages: [%5.2f, %5.2f, %5.2f, %5.2f]"
    mean(r1,dims=2)...
@printf "rp2 | Time averages: [%5.2f, %5.2f, %5.2f, %5.2f]"
    mean(r2,dims=2)...
@printf "rp3 | Time averages: [%5.2f, %5.2f, %5.2f, %5.2f]"
    mean(r3,dims=2)...
```

» 2c - Time averages

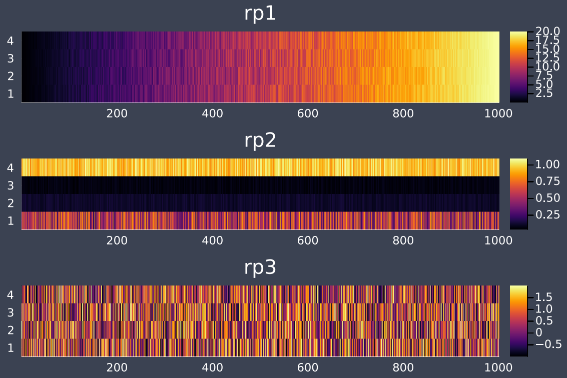
We get the following output

```
# -> rp1 | Time averages: [10.09, 10.04, 10.00, 10.04]
# -> rp2 | Time averages: [ 0.55,  0.11,  0.07,  0.95]
# -> rp3 | Time averages: [ 0.55,  0.52,  0.57,  0.46]
```

We gather from this that rp3 is ergodic, while rp1 and rp2 are not. Displaying each process as images is pretty streight forward aswell with

```
l = @layout [rp1 ; rp2 ; rp3]
h1 = heatmap(r1)
h2 = heatmap(r2)
h3 = heatmap(r3)
plot(
    h1, h2, h3, layout = l,
    title=["rp1" "rp2" "rp3"]
)
```

This gives us the following images



Here we can clearly see that rp1 and rp2 are not ergodic as no individual random variable are able to represent the enitre domain of each process. rp3 however, satisfy this criteria, and match with our previous findings about its ergodicity.