

# Metropolis-Hastings (MH) para simular amostras da distribuição Gama ( $k=5, \theta=2$ )

Adriana Eva Fernandes da Silva

# Descrição do Algoritmo MH

O Metropolis-Hastings é um algoritmo que funciona gerando amostras da distribuição de interesse que pode ser complicada de simular, usando uma distribuição proposta que é mais fácil de simular. O algoritmo funciona gerando uma cadeia de Markov, em que cada estado da cadeia representa uma possível amostra da distribuição de interesse. A cada passo da cadeia, o algoritmo propõe uma nova amostra aleatória, que é aceita ou rejeitada de acordo com uma regra de aceitação probabilística baseada na relação entre as distribuições de probabilidade da amostra proposta e da amostra atual.

# Distribuição de interesse

A distribuição *Gama* é comumente utilizada para modelar tempos de espera, taxas de falha, e outras grandezas positivas. A *Gama* com parâmetros de forma  $k = 5$  e de escala  $\theta = 2$ , indica que a distribuição tem forma um pouco mais assimétrica e achatada. A sua função densidade é dada por:

- $\pi(x) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{\frac{-x}{\theta}}$ , com  $k > 0$ ,  $\theta > 0$  e suporte  $x \in (0, \infty)$

Note que a constante normalizadora  $\frac{1}{\Gamma(k)\theta^k}$  não depende de  $x$ , portanto, com o objetivo de otimizar o tempo de execução do algoritmo, a nossa distribuição de interesse pode ser escrita de maneira mais simples, somente utilizando o núcleo da *f.d.p* da *Gama*, como:

- $\pi(x) = x^{k-1} e^{\frac{-x}{\theta}}$ , com  $k > 0$ ,  $\theta > 0$  e suporte  $x \in (0, \infty)$

Onde o valor esperado e a variância são, respectivamente,

- $E(X) = k \cdot \theta = 10$

- $VAR(X) = k \cdot \theta^2 = 20$

# Distribuição proposta

A escolha da função de densidade proposta impacta significativamente na eficiência do algoritmo MH. Dessa forma, buscaremos a função de transição de duas formas: utilizando as propostas do **MH independente** e **MH passeio aleatório**. Ao fim, faremos uma comparação entre elas.

# MH - Cadeia independente

Uma forma de simular valores para propor a transição da cadeia de markov é fazê-lo de forma independente em relação ao valor atual da cadeia. Isso pode ser feito assumindo que  $q(X_{prop}|X) = g(X_{prop})$  em que  $g(.)$  é a densidade de uma distribuição que não depende de  $x$ . Além disso, a escolha da melhor distribuição proposta leva em consideração aquela que possui formato parecido com a distribuição de interesse, média parecida, mesmo suporte e é fácil de gerar amostras. Nesse sentido vamos escolher como distribuição proposta a distribuição *Exponencial*(0.1). A sua relação com a distribuição *Gama*(5, 2) se dá pelo parâmetro da exponencial  $\lambda = \frac{1}{k\theta}$  onde  $k\theta$  é a média da gama. Sendo assim, utilizaremos como distribuição proposta a *Exponencial* com parâmetro  $\lambda = \frac{1}{k\theta} = \frac{1}{2 \cdot 5} = \frac{1}{10} = 0.1$ , cuja função densidade é dada por:

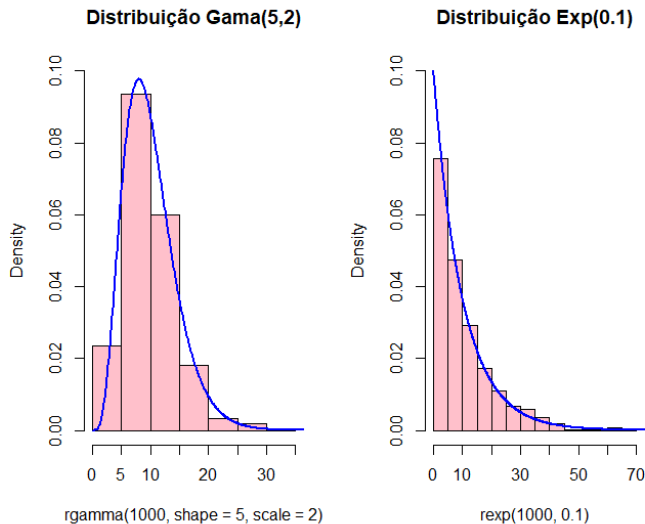
- $\lambda e^{-\lambda x}$ , com  $\lambda > 0$  e suporte  $x \in (0, \infty)$

Onde o valor esperado e a variância são, respectivamente,

- $E(X) = \frac{1}{\lambda} = 10$
- $VAR(X) = \frac{1}{\lambda^2} = 100$

**Abaixo, temos a representação gráfica para uma melhor visualização da distribuição de interesse e da distribuição proposta.**

# Gráficos das distribuições de interesse e proposta



**Figura:** Gráficos das distribuições  $Gama(k = 5, \theta = 2)$  e  $Exponencial(\lambda = 0.1)$



# Taxa de aceitação

A taxa de aceitação é, basicamente, o número de valores aceitos pela probabilidade de aceitação dividido pelo número de iterações. É importante escolher uma taxa de aceitação ideal que permita uma proporção suficientemente alta de propostas aceitas, para garantir que a cadeia de amostras se aproxime gradualmente para a distribuição desejada. No entanto, essa taxa não deve ser excessivamente alta, para evitar que o algoritmo gere amostras altamente correlacionadas.

Para o nosso caso, com 1000 interações do algoritmo MH cadeia independente, a taxa foi de 0.541, ou seja, a cada 100 valores propostos, o algoritmo aceita mais da metade deles, nos dando um indicativo de que a cadeia não está ficando parada em um mesmo valor por muito tempo.

# Verificando a convergência

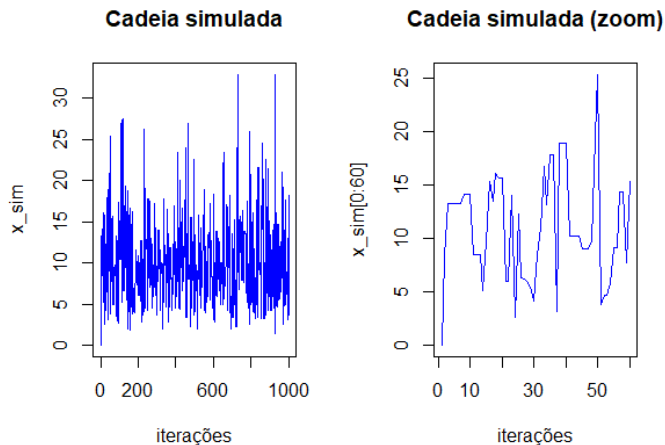
Apesar da cadeia simulada ter como distribuição de equilíbrio a distribuição desejada, essa só é alcançada depois que a cadeia convergiu. Logo, as primeiras iterações do algoritmo geralmente não são provenientes da distribuição desejada e a convergência precisa ser verificada para darmos continuidade no algoritmo MH independente.

Podemos verificar a convergência utilizando métodos gráficos baseados em medidas e testes estatísticos disponíveis em pacotes no software R. Consequente, alguns testes e tipos de gráfico para verificação da convergência:

- Trace plot;
- Diagnóstico de Geweke;
- Sobreposição de histogramas;



# Verificando burn in - Trace Plot



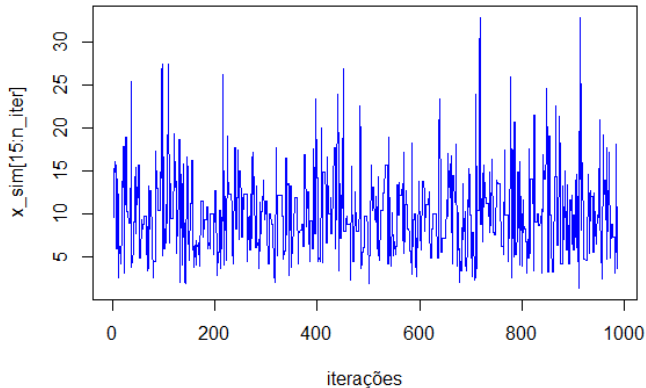
**Figura:** Gráfico trace plot da cadeia simulada

# Verificando burn in - Resultados

A parte inicial da sequência simulada, quando a convergência ainda não atingiu uma condição aceitável, deve ser descartada.

No gráfico trace plot da cadeia simulada, conseguimos ter uma ideia a partir de qual iteração a cadeia atingiu a convergência. Assumimos que a convergência acontece quando os traços oscilam em torno de uma linha horizontal e não apresentam nenhuma tendência óbvia. Como temos um número muito grande de iterações, foi feito um zoom nas 60 primeiras iterações da cadeia para melhor visualização e escolha do burn in. Depois de cerca de 14 iterações, no entanto, os traços se estabilizam e parecem oscilar em torno de uma linha horizontal.

# Trace plot pós burn in



**Figura:** Gráfico trace plot da cadeia simulada pós burn in

# Trace plot pós burn in - Resultados

O gráfico trace plot da cadeia simulada pós burn in nos mostra que com as 986 iterações restantes a cadeia já está estabilizada, ou seja, os valores simulados oscilam em torno de uma linha horizontal e não apresentam nenhuma tendência óbvia.

# Diagnóstico de Geweke

O diagnóstico de Geweke é utilizado na análise de MCMC para avaliar a estabilidade da cadeia e verificar se as amostras geradas estão corretas e são representativas da distribuição de interesse, através da comparação das médias das amostras geradas no início da cadeia com as médias das amostras geradas no final da cadeia. Se as médias são estatisticamente diferentes, então isso indica que a cadeia não está estabilizada e que as amostras geradas no final da cadeia não representam a distribuição de interesse corretamente.

Geweke sugere que se o valor dessa estatística teste for  $> 1.96$  ou  $< -1.96$ , consideramos que a cadeia simulada ainda não convergiu.

- A estatística de Geweke para a nossa cadeia simulada foi de **0.5944**, ou seja, a nossa cadeia pós burn in já atingiu a convergência.

# Autocorrelação e Saltos

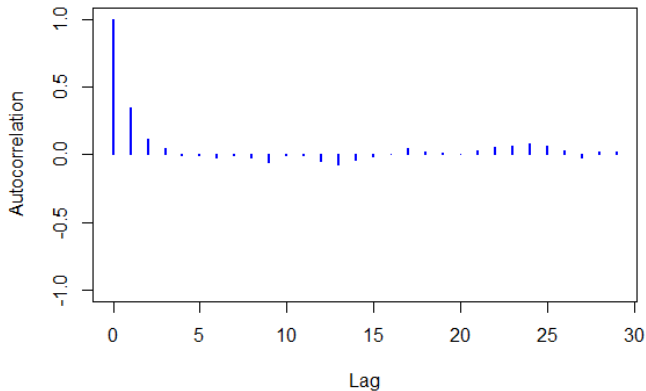
Estamos realizando uma simulação para gerar amostras de uma densidade proposta que represente a densidade de interesse utilizando o método MH cadeia independente, ou seja, a distribuição proposta não depende do valor atual da cadeia. Apesar disso, ainda temos uma cadeia de Markov, pois a probabilidade de aceitação em cada passo depende do passo anterior, ou seja, os valores da cadeia simulada continuam dependentes. Dessa forma, precisamos checar a existência de autocorrelação entre os valores simulados.

Podemos utilizar algumas medidas de autocorrelação para verificarmos o quanto os valores simulados na cadeia estão correlacionados e definir o número de saltos necessários para obter uma amostra independente a partir da amostra MCMC caso seja necessário.

Algumas medidas de correlação são:

- Gráfico de autocorrelação entre os valores simulados com diferentes lags;
- Effective sample size (tamanho efetivo da amostra);

# Gráfico de autocorrelação



**Figura:** Gráfico de autocorrelação da cadeia simulada

# Autocorrelações

**Tabela:** Autocorrelações

Lag	Autocorrelação
Lag 0	1.000000000
Lag 1	0.344238315
Lag 2	0.11777041
Lag 10	-0.00129560
Lag 50	0.02081907



# Autocorrelação - Resultados

O gráfico da autocorrelação da cadeia simulada nos mostra que não existe dependência entre os valores simulados em tempos diferentes, com exceção do lag 0 (que é a autocorrelação do valor inicial da cadeia com ele mesmo, resultando sempre em 1). A partir do lag 3, a autocorrelação é muito próxima de zero.

Com base nos resultados obtidos através do ACF, implementaremos saltos de tamanho 2, desconsiderando, assim, os lags que apresentaram maior autocorrelação. Essa decisão foi tomada porque, ao desconsiderar os lags com maior correlação, espera-se que a cadeia possa se mover de forma mais eficiente pelo espaço de amostragem, gerando amostras independentes e mais representativas da distribuição desejada.

# Tamanho efetivo da amostra (ESS)

É definido como o número de valores simulados efetivamente independentes da distribuição desejada. Grande discrepância entre o ESS e o tamanho da amostra testada indica um *poor mixing* da cadeia e autocorrelações altas entre os valores simulados.

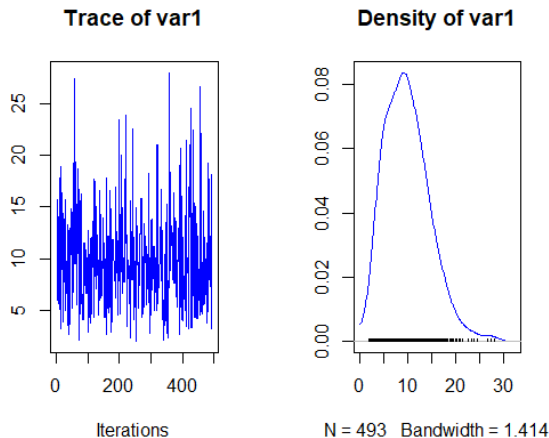
O tamanho efetivo da amostra foi de 480.6741 com um salto de 2.080412.

Com os resultados apresentados acima, a cadeia de Markov simulada apresentou uma boa capacidade em gerar amostras independentes.

# Conclusão para o tamanho da amostra

Com base na análise da autocorrelação, foi observado que a partir do lag 3 as amostras geradas são significativamente independentes. Ademais, o segundo resultado nos sugeriu um salto de 2.080412. Portanto, a escolha de um salto de tamanho 2 foi considerada uma estratégia apropriada para garantir a independência das amostras geradas. A nossa amostra final foi de tamanho 493.

# Verificando a convergência na amostra final - Gráficos



**Figura:** Gráfico dos valores e densidade da cadeia simulada

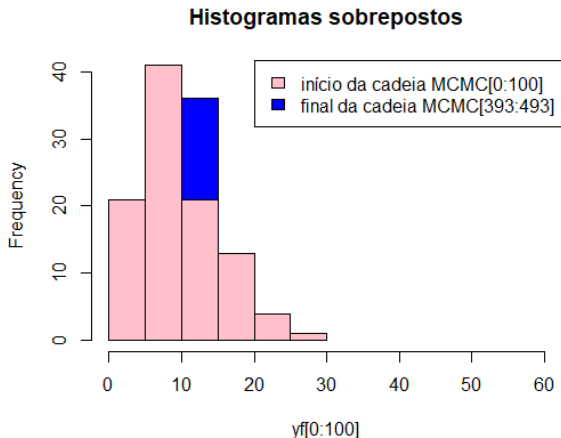
# Verificando a convergência da amostral final - Resultados

Após a realização do burn in e da escolha do salto apropriado, se observarmos o gráfico dos valores e densidade da cadeia simulada para uma amostra da densidade proposta Exponencial com parâmetro  $\lambda = 0.1$ , conseguimos ver que os valores simulados estão estabilizados e que a densidade de kernel estimada da densidade proposta é muito próxima da densidade de interesse Gama com parâmetros  $k = 5$  e  $\theta = 2$ , sendo  $k = 5$  o parâmetro de forma e  $\theta = 2$  o parâmetro de escala.

# Sobreposição dos histogramas

Outra forma de vermos a convergência na amostra final é através do gráfico dos histogramas sobrepostos da cadeia simulada em seus valores iniciais e finais. Se a convergência já foi atingida desde o início da cadeia MCMC, espera-se que o comportamento dos valores no início da sequência seja parecido com o comportamento dos valores do final da cadeia e os histogramas estejam muito parecidos e sobrepostos.

# Sobreposição dos histogramas - Gráfico



**Figura:** Gráfico dos histogramas sobrepostos da cadeia simulada em seus valores iniciais e finais

# Sobreposição dos histogramas - Medidas descritivas

**Tabela:** Medidas descritivas: valores iniciais vs. valores finais

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Var.
val. iniciais	3.143	5.624	8.647	10.049	14.075	26.666	20.48299
val. finais	2.100	6.206	9.751	10.004	12.799	27.403	28.88029

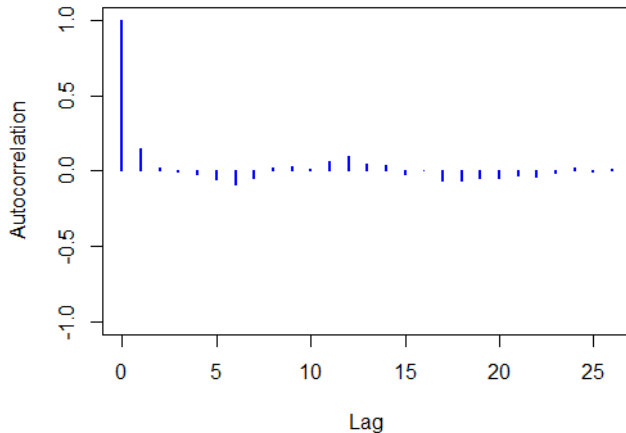


# Sobreposição dos histogramas - Resultados

O gráfico dos histogramas sobrepostos da cadeia simulada em seus valores iniciais e finais e da tabela das medidas descritivas, nos mostra que o comportamento dos valores do início da cadeia são bem parecidos com o comportamento dos valores no final da cadeia. Os histogramas também são muito parecidos em termos de formato, além de estarem sobrepostos.

Podemos perceber também, que a média nos resultou em valores praticamente iguais, que também é exatamente a média da nossa distribuição de interesse, a gama. Ou seja, com exceção da diferença da frequência em que cada valor aparece no início e no final da cadeia simulada, os histogramas nos dão um indicativo de que a cadeia simulada final, convergiu.

# Verificando a autocorrelação na amostra final



**Figura:** Gráfico de autocorrelação da amostra final

# Verificando a autocorrelação na amostra final

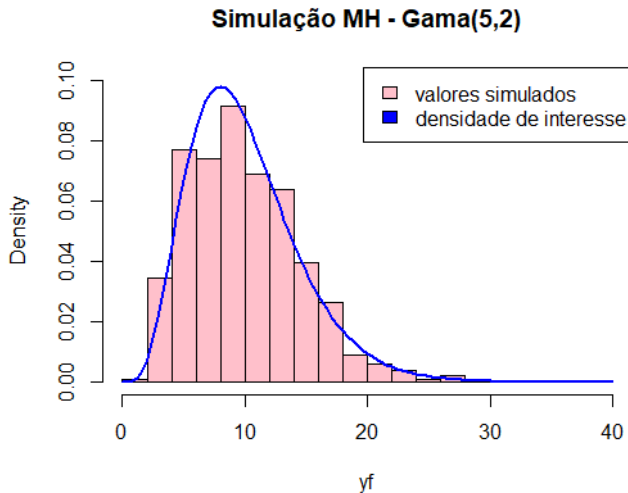
**Tabela:** Autocorrelações

Lag	Autocorrelação
Lag 0	1.00000000
Lag 1	0.15146776
Lag 2	0.01683209
Lag 10	0.01374374
Lag 50	0.05553060

# Verificando a autocorrelação na amostra final - Resultados

Analizando o gráfico de autocorrelação e os resultados apresentados pela tabela da amostra final podemos perceber que não existe dependência entre os valores, temos uma correlação no lag 1 muito próxima de 0, nos indicando uma baixa correlação. Assim, concluímos que estamos conseguindo gerar amostras independentes.

# Simulação final - Gráfico



**Figura:** Gráfico histograma dos valores simulados com a curva da densidade de interesse

# Simulação final - Medidas de resumo

**Tabela:** Medidas de resumo: distribuição de interesse vs. distribuição final

	Média	Variância
Dist.Interesse	10	20
Dist.Final	9.913185	21.24091

# Simulação final - Resultados

O gráfico histograma dos valores simulados com a curva da densidade de interesse e a tabela de medidas de resumo nos mostra que a densidade *Exponencial*(0.1) está gerando amostras independentes que realmente representam a densidade *Gama*(5, 2), uma vez que os valores simulados da distribuição final apresentam um comportamento bem próximo aos valores da densidade de interesse.

Com base nos valores obtidos da distribuição final, podemos constatar que os resultados foram bem próximos da distribuição de interesse.

# MH - Passeio aleatório

Uma outra maneira prática de construir um algoritmo MH é usar o valor de  $x$  atual para propor um movimento da cadeia, ou seja, considerar uma exploração da vizinhança em torno do valor atual da cadeia e não uma sugestão completamente independente dele.

No passeio aleatório, o valor proposto para a transição é simulado como  $X_{prop} = x + Z$ , em que  $x$  é o valor atual da sequência e  $Z$  é uma variável aleatória simétrica com média zero e densidade  $f(\cdot)$ .



# Distribuição proposta

No MH passeio aleatório, o nosso interesse ainda está em simular valores de uma distribuição *Gamma* com parâmetros  $k=5$  e  $\theta = 2$ , cuja f.d.p (a nossa  $\pi(x)$ ), já foi vista anteriormente.

A diferença da abordagem MH cadeia independente é que agora iremos simular amostras de uma distribuição proposta  $f(\cdot)$  dependente. A escolha para distribuição proposta foi usual e o valor adequado da variância foi escolhido empiricamente (por tentativa e erro). Escolhemos como proposta uma distribuição *Normal* com parâmetros  $\mu = 0$  e  $\sigma^2 = 5$ . Sua f.d.p é dada por:

- $f(\cdot) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$  com  $\sigma^2 > 0$  e suporte  $x \in \mathbb{R}$ .

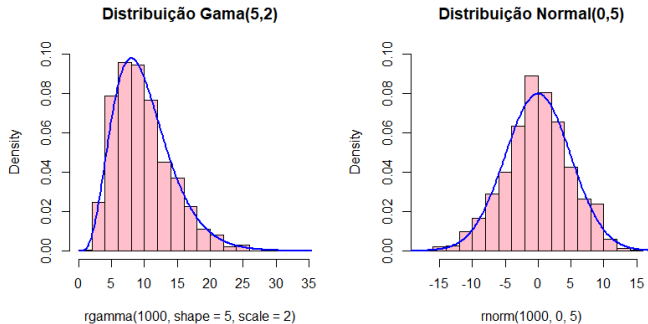
Onde o valor esperado e a variância são, respectivamente,

- $E(X) = \mu = 0$
- $VAR(X) = \sigma^2 = 5$

O valor proposto deve ser simulado pelo passeio aleatório  $x_{prop} \sim N(x_{t1}, v)$  em que  $x_{t1}$  e  $v$  são, respectivamente, o valor inicial da cadeia e a variância proposta.

**Abaixo, temos a representação gráfica para uma melhor visualização da distribuição de interesse e da distribuição proposta.**

# Gráficos das distribuições de interesse e proposta

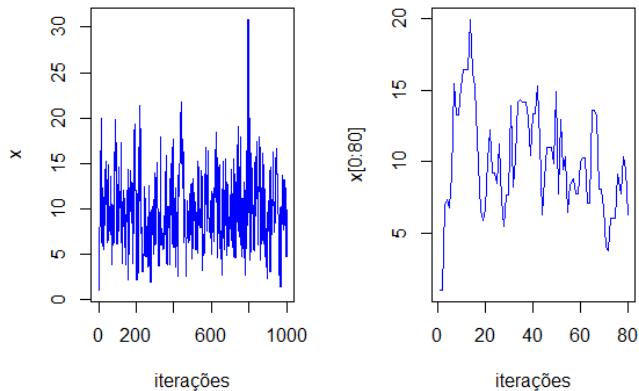


**Figura:** Gráficos das distribuições  $Gama(k = 5, \theta = 2)$  e  $Normal(\mu = 0, \sigma^2 = 5)$

# Taxa de aceitação

Para o nosso caso, com o algoritmo MH passeio aleatório e 1000 iterações, a taxa foi de 0.67, ou seja, a cada 100 valores propostos, o algoritmo aceita mais da metade deles, nos indicando que a cadeia não está ficando parada em um mesmo valor por muito tempo.

# Verificando burn in - Trace plot

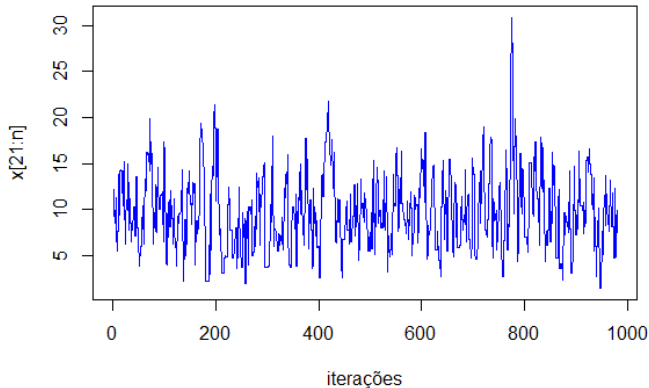


**Figura:** Gráfico trace plot da cadeia simulada

# Verificando burn in - Resultados

No gráfico trace plot da cadeia simulada, conseguimos ter uma idéia a partir de qual iteração a cadeia atingiu a convergência. Assumimos que a convergência acontece quando os traços oscilam em torno de uma linha horizontal e não apresentam nenhuma tendência óbvia. Como temos um número muito grande de iterações, foi feito um zoom nas 80 primeiras iterações da cadeia para melhor visualização e escolha do burn in. Depois de cerca de 19 iterações, no entanto, os traços se estabilizam e parecem oscilar em torno de uma linha horizontal.

# Trace plot pós burn in



**Figura:** Gráfico trace plot da cadeia simulada pós burn in

# Trace plot pós burn in - Resultados

O gráfico trace plot da cadeia simulada pós burn in nos mostra que com as 981 iterações restantes a cadeia já está estabilizada, ou seja, os valores simulados oscilam em torno de uma linha horizontal e não apresentam nenhuma tendência óbvia.

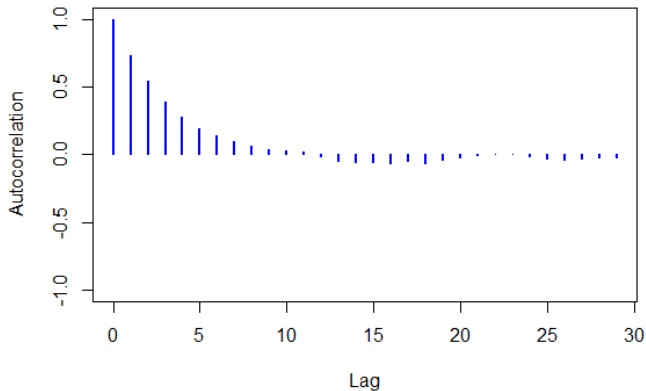
# Diagnóstico de Geweke

De acordo com a sugestão de Geweke, se o valor da estatística de teste for maior do que 1,96 ou menor do que -1,96, podemos concluir que a cadeia simulada ainda não atingiu a convergência. Esses valores críticos correspondem a um nível de significância de 5% e são baseados na distribuição normal padrão.

- A estatística de Geweke para a nossa cadeia simulada foi de **0.6976**, ou seja, a nossa cadeia pós burn in já atingiu a convergência.



# Verificando Saltos - Autocorrelação



**Figura:** Gráfico de autocorrelação da cadeia simulada

# Autocorrelações

**Tabela:** Autocorrelações

Lag	Autocorrelação
Lag 0	1.000000000
Lag 1	0.734334406
Lag 2	0.544737187
Lag 3	0.389373663
Lag 6	0.139880028
Lag 50	-0.005032324

# Autocorrelação - Resultados

O gráfico de autocorrelação revela que a autocorrelação diminui à medida que o lag aumenta, o que sugere que a cadeia está se tornando cada vez mais independente.

Analisando o gráfico, para o lag 3 temos uma autocorrelação muito pequena. Isso indica que, a cada três passos as amostras não estão mais correlacionadas com as amostras anteriores.

# Tamanho efetivo da amostra (ESS)

É definido como o número de valores simulados efetivamente independentes da distribuição desejada. Grande discrepância entre o ESS e o tamanho da amostra testada indica um *poor mixing* da cadeia e autocorrelações altas entre os valores simulados.

O tamanho efetivo da amostra foi de 150.1164 com um salto de 6.66.

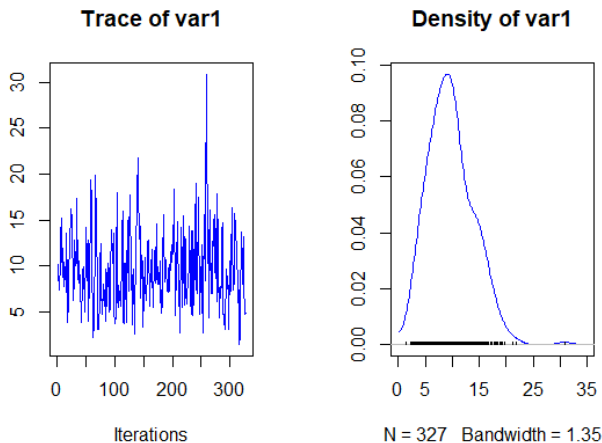
Com os resultados apresentados acima, a cadeia de Markov simulada não apresentou uma boa capacidade em gerar amostras independentes.

# Conclusão para o tamanho da amostra

Com base no gráfico de autocorrelação, foi observado que a partir do lag 4 a correlação é bem pequena, nos sugerindo um salto de 3. Porém, o segundo resultado nos sugeriu um salto de 6.66.

Portanto, a escolha de um salto de tamanho 3 foi considerada uma estratégia apropriada para garantir a independência das amostras geradas, pelo fato de não haver muita perda de informação. A nossa amostra final foi de tamanho 327.

# Verificando a convergência na amostra final - Gráficos



**Figura:** Gráfico dos valores e densidade da cadeia simulada

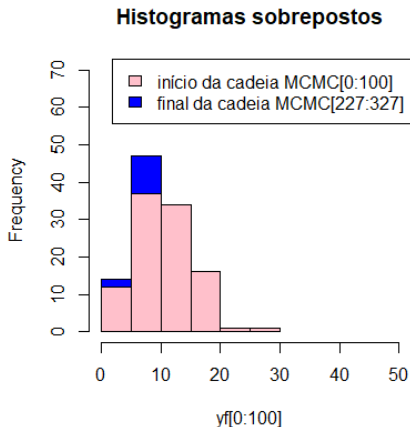
# Verificando a convergência na amostra final - Resultados

Temos que, a figura formada pelo gráfico de traço da amostra final apresenta um comportamento aleatório, sem tendências.

Além disso, a densidade estimada pelo método de Kernel da amostra final aproxima-se da densidade de uma Gamma.

Observamos também que, devido à autocorrelação nas amostras, a eficiência da amostragem é equivalente a uma amostra de tamanho 327.

# Sobreposição dos histogramas - Gráfico



**Figura:** Gráfico dos histogramas sobrepostos da cadeia simulada em seus valores iniciais e finais



# Sobreposição dos histogramas - Medidas descritivas

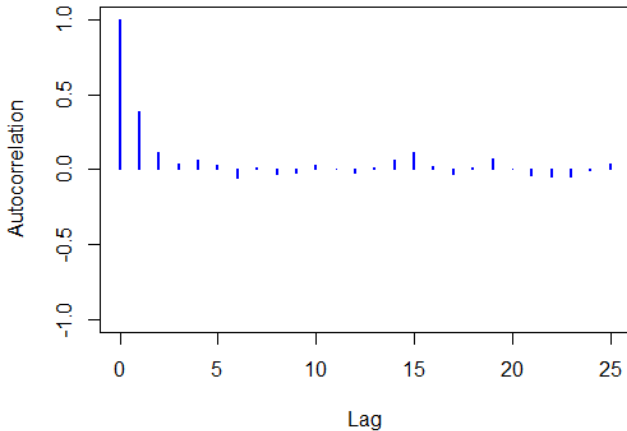
**Tabela:** Medidas descritivas: valores iniciais vs. valores finais

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Var.
val. iniciais	1.909	6.893	9.012	9.388	11.636	19.807	16.23529
val. finais	1.426	7.135	10.131	10.379	13.772	25.476	20.6686

# Sobreposição dos histogramas - Resultados

O gráfico dos histogramas sobrepostos da cadeia simulada em seus valores iniciais e finais nos mostra que o comportamento dos valores do início da cadeia são bem parecidos com o comportamento dos valores no final da cadeia. Os histogramas também são muito parecidos em termos de formato, além de estarem sobrepostos. Ou seja, os histogramas nos dão um indicativo de que a cadeia simulada final, convergiu.

## Verificando a autocorrelação na amostra final



**Figura:** Gráfico de autocorrelação da amostra final

# Verificando a autocorrelação na amostra final

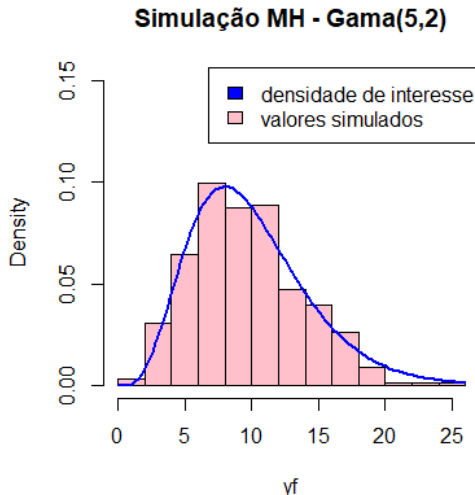
**Tabela:** Autocorrelações

Lag	Autocorrelação
Lag 0	1.00000000
Lag 1	0.38795098
Lag 2	0.11083862
Lag 3	0.03383230
Lag 6	-0.05297126
Lag 50	-0.08038930

# Verificando a autocorrelação na amostra final - Resultados

Analizando o gráfico de autocorrelação e os resultados apresentados pela tabela da amostra final podemos perceber que não existe dependência entre os valores, temos uma correlação no lag 1 de 0.39, nos indicando uma correlação relativamente baixa. Assim, concluímos que estamos conseguindo gerar amostras independentes.

# Simulação final



**Figura:** Gráfico histograma dos valores simulados com a curva da densidade de interesse

# Simulação final - Medidas de resumo

**Tabela:** Medidas de resumo: distribuição de interesse vs. distribuição final

	Média	Variância
Dist.Interesse	10	20
Dist.Final	9.670077	16.85251

# Simulação final - Resultados

O gráfico histograma dos valores simulados com a curva da densidade de interesse e a tabela de medidas de resumo nos mostram que a densidade *Exponencial*(0.1) está gerando amostras independentes que realmente representam a densidade *Gama*(5, 2), uma vez que os valores simulados da distribuição final apresentam um comportamento bem próximo aos valores da densidade de interesse.

Com base nos valores obtidos da distribuição final, podemos constatar que os resultados foram bem próximos da distribuição de interesse.



# Comparação entre os dois algoritmos MH utilizados

A cadeia independente gera amostras diretamente da distribuição desejada, sem precisar depender das amostras anteriores. Por outro lado, o passeio aleatório gera amostras dependentes, onde cada nova amostra é gerada a partir da amostra anterior, de forma aleatória.

A cadeia independente é boa porque é mais rápida e não tem problemas de autocorrelação, que acontecem quando as amostras dependem muito das anteriores. Porém, o passeio aleatório é mais flexível e consegue lidar com distribuições mais complicadas e de muitas dimensões.

# Comparação entre os dois algoritmos MH utilizados

Com o algoritmo de cadeia independente foi obtida uma amostra final de tamanho 493, ou seja, não houveram muitas informações perdidas. Por outro lado, com o algoritmo do passeio aleatório foi obtida uma amostra final de tamanho 327. Isso acontece pelo fato desse último algoritmo ser mais utilizado para gerar amostras de distribuições complexas. De fato, era esperado que o algoritmo de cadeia independente seria mais eficiente, pois estamos simulando uma distribuição relativamente simples.

A convergência foi mais rápida para o MH cadeia independente, e houve uma baixa correlação na amostra final, o que nos indicou que os nossos valores simulados eram provenientes de amostras independentes. Sendo assim, consideramos que o algoritmo Metrópoles Hastings com a proposta Cadeia independente é eficiente computacionalmente.

# Comparação entre os dois algoritmos MH utilizados

No MH passeio aleatório a amostra final obtida foi relativamente pequena, neste caso, o algoritmo é computacionalmente menos eficiente. Para melhorar o algoritmo, uma opção seria realizar um aumento no número de iterações, o que faria com que obtivessemos uma amostra maior, com convergência mais rápida e, conseqüentemente mais precisa.

Embora realizar um aumento no número de iterações possa ser uma opção para melhorar em alguns aspectos o algoritmo, no entanto, o seu tempo de execução também pode aumentar, o que o tornaria computacionalmente menos eficiente.

Dessa forma, devemos encontrar um equilíbrio entre todos os aspectos de acordo com a necessidade.