MCMC na modelagem da variação diária do dólar americano em 2020

José Arthur de Souza

Abril de 2021

1 Resumo

O objetivo deste artigo é modelar a variação diária da cotação do preço de compra dólar americano em reais brasileiros; o preço de compra é igual ao preço de venda mais uma constante. Para isso utilizarei um método de estimação de parâmetros conhecido como MCMC - Monte Carlo Markov Chains. Também usarei uma versão levemente modificada do método que acabei considerando melhor.

2 Introdução

Bom, antes de mais nada precisarei de dados sobre a cotação do dólar americano. Estes dados foram conseguidos em [1]. O site disponibiliza uma planilha com os dados de no máximo um semestre por vez, então foi necessário baixar duas vezes e juntar todos os dados após. Claro, ainda houve uma limpeza dos dados, e algumas construções afim de visualizar melhor os dados. A planilha com os dados arrumados para nos servir melhor é a cotacao2.csv.

Para que você não precise abrir a planilha, a cotação do dólar e a variação a cada dia podem ser observada a seguir (ver Figuras 1 e 2).

Figura 1: Preço de compra do dólar

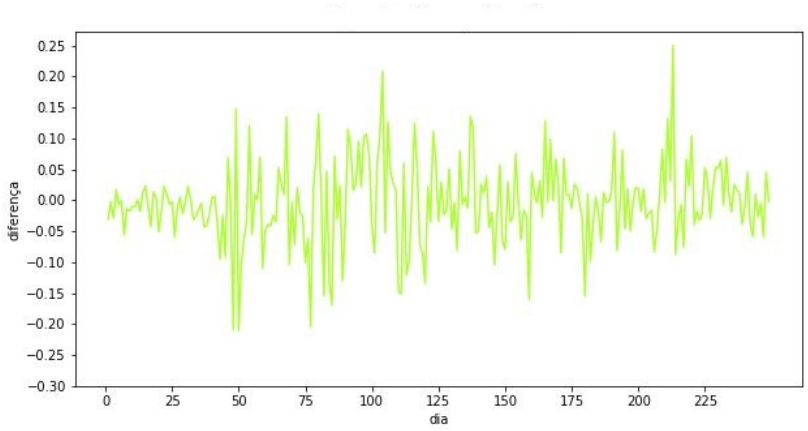

Figura 2: Variação do preço em cada dia

Como nos interessa modelar a variação do preço, é mais útil observar a frequência com que as diferenças de preço se encontram em certos intervalos. Fazemos isso construindo um histograma dos valores da variação distribuídos em, por exemplo, 30 intervalos (ver Figura 3).

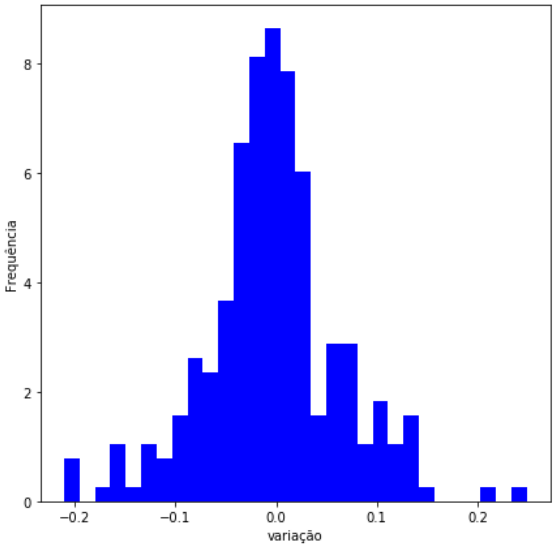

Figura 3: Variação do preço em cada dia

A partir disso, podemos supor alguma distribuição de probabilidade para

os dados; poemos até supor mais de uma, mas esse não é o foco deste trabalho. Tendo algum conhecimento prévio sobre o formato das distribuições mais clássicas é razoável assumir que uma distribuição Normal ou uma Gamma pode estar por trás desse fenômeno. Por motivos de simplicidade - porque a Gamma exige o cálculo de uma função não muito amigável - escolhi modelar como uma normal.

OBS: faz sentido usar simplesmente uma distribuição de probabilidade como modelo pois os dados são claramente influenciados muito mais por algum fator aleatório do que qualquer outra coisa (regra ou lei, e.g.).

Sendo o modelo uma distribuição de probabilidade e, mais do que isso, sendo uma distribuição simples como é a normal talvez haja um impulso em se estimar os parâmetros de alguma outra maneira já conhecida, por exemplo, maximizando-se a verossimilhança analiticamente. No entanto, como dito anteriormente, o objetivo é também explorar o método MCMC, que na verdade usa a mesma ideia, maximizar a verossimilhança, porém numéria e computacionalmente. Então vamos ao método.

3 MCMC: o que é?

MCMC, abreviação para Monte Carlo Markov Chains, é um método, um algoritmo mais precisamente, para estimar os parâmetros que maximizam a verossimilhança de uma distribuição dada uma amostra de dados retirados dessa distribuição. À princípio, pode parecer um pouco inútil uma vez que isso pode ser feito analiticamente gerando um resultado mais preciso, mas para distribuições de probabilidade complicadas e com vetor de parâmetros de alta dimensão isso pode ser praticamente impossível; nesses casos o MCMC pode ser a melhor solução.

Imaginemos que se tenha uma distribuição com vetor de parâmetros $\theta = (\alpha, \beta, \gamma, ...)$ que se quer estimar. Imaginemos também que essa distribuição tem função de probabilidade $f(X|\theta)$ e que é suposta uma priori $\epsilon(\theta)$ para θ.

Então a verossimilhança de θ com os dados $\vec{X}$ é $L(\theta|\vec{X}) = (\prod_{i=1}^{n} f(X_i|\theta))\epsilon(\theta)$

A estratégia do algoritmo para encontrar um θ^* que maximize (ou pelo menos quase) a verossimilhança $L(\theta|\vec{X})$ é, partindo de um θ_0 inicial, ir percorrendo o espaço de parâmetros tomando sempre um vizinho θ_1 de θ_0, comparando as suas verossimilhanças e atualizando ou mantendo o θ_0, buscando sempre atingir um "pico"da verossimilhança.

Esse vizinho θ_1 pode ser conseguido tomando-se para cada componente o valor de uma variável aletória com distribuição Normal de média igual à componente correspondente em θ_0 e uma variância qualquer. Por exemplo, se α é um dos parâmetros a serem estimados, fazemos $\alpha_1 \sim N(\alpha_0, \sigma)$; esse σ dita o quão grande queremos que seja o passo de um θ a outro. Abaixo (ver figuras 4 e 5), são mostrados gráficos com distribuições normais com médias 2 e 1.5 e variâncias 1.5 e 0.3 respectivamente e um gráfico com a distribuição conjunta dessas duas variáveis; essas são as distribuições de onde poderiamos escolher um θ_1 sendo $\theta_0 = (2, 1.5)$. As variâncias são, em um certo nível, arbitrárias.

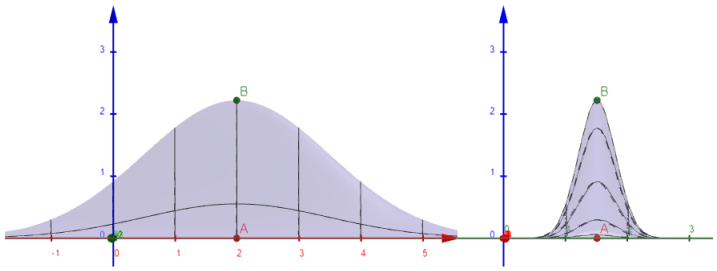

Figura 4: distribuições de μ e σ respctivamente

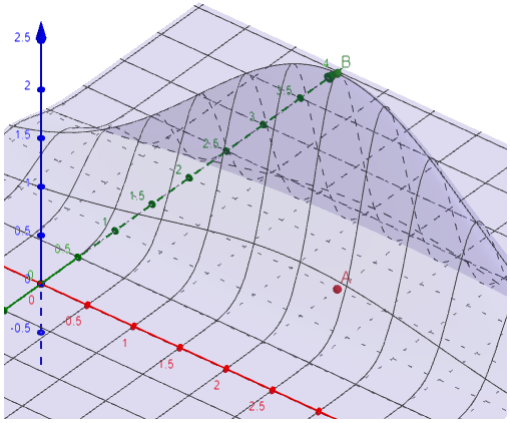

Figura 5: distribuição conjunta de μ e σ ou distribuição de θ

Tendo θ_0 e θ_1 podemos calcular suas verossimilhanças $L(\theta_0|\vec{X})$ e $L(\theta_1|\vec{X})$. Aí podemos comparar:

- Se $L(\theta_1|\vec{X}) \geq L(\theta_1|\vec{X})$, então atualizamos θ_0 fazendo-o ser igual a θ_1. Isso nos dá o aspecto de cadeia de markov (Marvov Chain), pois tomando sempre o θ de maior verossimilhança tendemos a convergir para algum máximo local no espaço de parâmetros.

- Se não, atualizamos θ_0 fazendo-o ser igual a θ_1 com probabilidade $\frac{L(\theta_1|\vec{X})}{L(\theta_0|\vec{X})}$.

  Isso dá o aspecto de Monte Carlo, pois tem um elemento aleatório. É isso também que tenta evitar que o θ fique preso num máximo local e permite que possa chegar a um máximo absoluto da função verossimilhança.

  Note que quanto maior é $L(\theta_0|\vec{X})$ em relação a $L(\theta_1|\vec{X})$, menor é a probabilidade de θ_1 ser o novo θ_0.

Com o possivelmente novo θ_0, repetimos o processo (muitas e muitas vezes) e esperamos que convirja.

OBS: Comumente acaba sendo melhor lidar com a função $\ln(L(\theta_0|\vec{X}))$ *- isso porque maximizar essa função é o mesmo que maximizar a verossimilhança, mas em geral, a expressão fica mais simpática. Se isso for feito, não se pode esquecer de aplicar a inversa,* e^x*, na hora de comparar as verossimilhanças.*

4 Modelagem

Supomos inicialmente que $\vec{X} \sim N(\mu, \sigma)$, isto é, que os dados seguem uma distribuição normal com média μ e variância σ. A função de densidade de probabilidade de uma normal com esses parâmetros é $\phi(X) = \frac{1}{\sqrt{2\pi}\sigma} exp[-\frac{1}{2}\frac{(X-\mu)^2}{\sigma^2}]$ e, portanto, a verossimilhança é dada por $L(\theta|\vec{X}) = (\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}])\epsilon(\mu, \sigma)$.

Assim, temos:

$\ln(L(\theta|\vec{X})) = \ln((\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}])\epsilon(\mu, \sigma))$

$\ln(L(\theta|\vec{X})) = (\sum_{i=1}^{n} \ln(\frac{1}{\sqrt{2\pi}\sigma} exp[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}]) + \ln(\epsilon(\mu, \sigma))$

$\ln(L(\theta|\vec{X})) = (\sum_{i=1}^{n} \ln(1) - \ln(\sqrt{2\pi}) - \ln(\sigma) - \frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}) + \ln(\epsilon(\mu, \sigma))$

Como $\ln(1) = 0$ e $\ln(\sqrt{1\pi}) = c$, $c \in R$ constante, podemos apenas desconsiderar esses termos, pois na hora de aplicar a inversa e fazer a razão das verossimilhanças eles acabam sumindo. Desta maneira, ficamos com:

$\ln(L(\theta|\vec{X})) = (\sum_{i=1}^{n} - \ln(\sigma) - \frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}) + \ln(\epsilon(\mu, \sigma)) + k$, $k \in R$ constante.

A Priori será definida de maneira que não haja "preferência"sobre quaisquer valores, mas nada impediria que fosse de outra forma. Ainda assim, gostaríamos que σ fosse sempre positivo, pois é uma medida de dispersão, inerentemente positiva. Isso será traduzido da seguinte forma:

$$\epsilon(\mu, \sigma) = \begin{cases} 0, & \text{se } \sigma \leq 0; \\ 1, & \text{se não} \end{cases}$$

Além disso, podemos escrever uma função de "aceitação"do θ_1 que resulta em 1 se θ_0 for substituído e 0 se não do seguinte modo:

$$\Lambda(L(\theta_0|\vec{X}), L(\theta_1|\vec{X})) = \begin{cases} 0, & \text{se } L(\theta|\vec{X}) < L(\theta|\vec{X}); \\ X, X \sim Be(L(\theta_1|\vec{X})/L(\theta_0|\vec{X})), & \text{se nao} \end{cases},$$

Lembrando que $\theta_1 = (\mu_1, \sigma_1)$ com $\mu_1 \sim N(\mu_0, q)$ e $\sigma_1 \sim N(\sigma_0, q)$

5 Implementação

A implementação pode ser encontrada no arquivo A1modest.ipynb . A seguir veremos os resultados desta.

O código teve como chute inicial $\theta = (1, 2)$ e 5000 iterações.

Abaixo, as figuras mostram o caminho percorrido pelo espaço de parâmetros (no caso da normal, o R^2) nas primeiras 50 alterações de θ_0, nas 50 últimas (ver Figura 6) e em todas as iterações (ver Figura 7). As estrelas amarelas

representam os θ_1 aceitos. Os X's vermelhos representam os θ_1 não aceitos ou rejeitados. Os traços em verde são o caminho do θ até o θ^* encontrado pelo algoritmo. Os traços cinzas são as tentativas falhas de ir para outros lados.

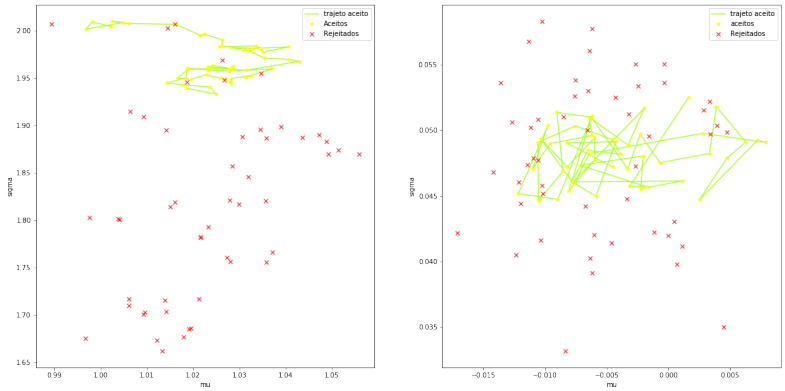

Figura 6: Primeiras e últimas 50 alterações

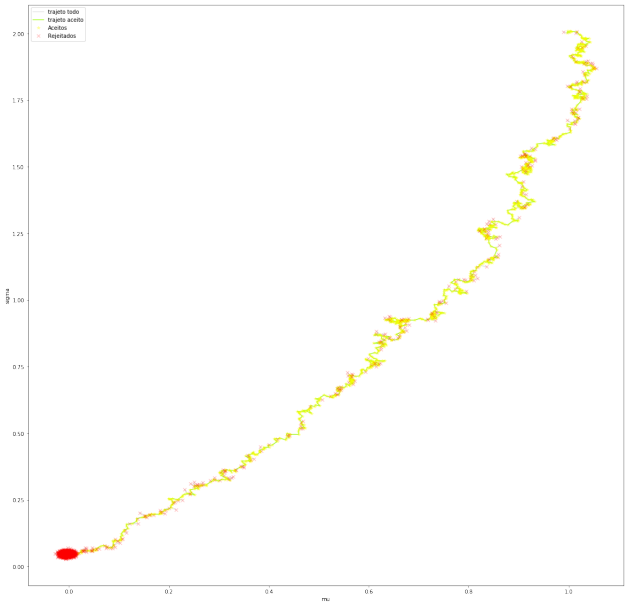

Figura 7: todas as iterações

Note como nas primeiras alterações parece haver um claro viés para que o próximo θ_0 seja tal que o σ seja menor que o anterior. Isso porque o σ

inicial é muito mais alto do que o σ^* ótimo. No entanto, nas últimas alterações parece não haver viés para direção nenhuma, uma vez que qualquer θ naquela vizinhança está próximo de θ^*. Note também como há muito mais rejeição perto de onde está o θ^*, aquela região vermelha.

A seguir, os gráficos que mostram os valores assumidos por μ e σ nas últimas 150 alterações de θ, bem como o histograma desse valores distribuídos em 20 intervalos (ver Figuras 8 e 9).

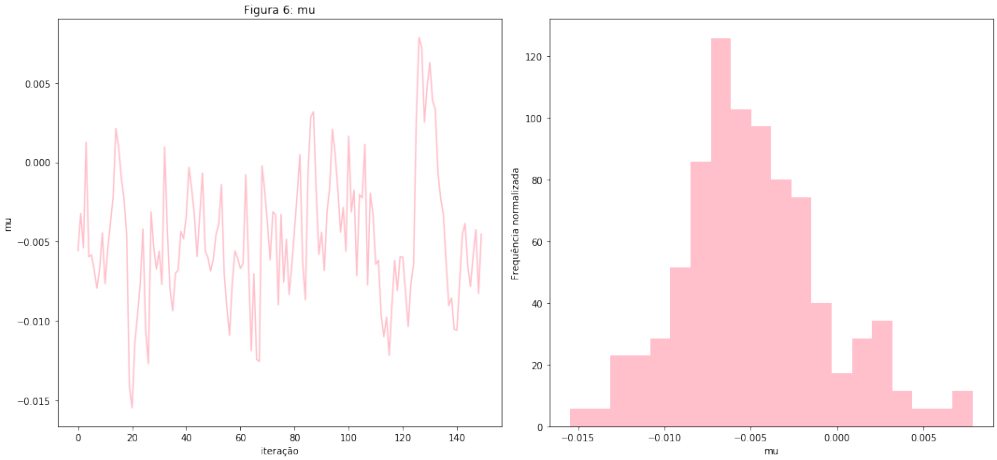

Figura 8: valores de μ e histograma destes

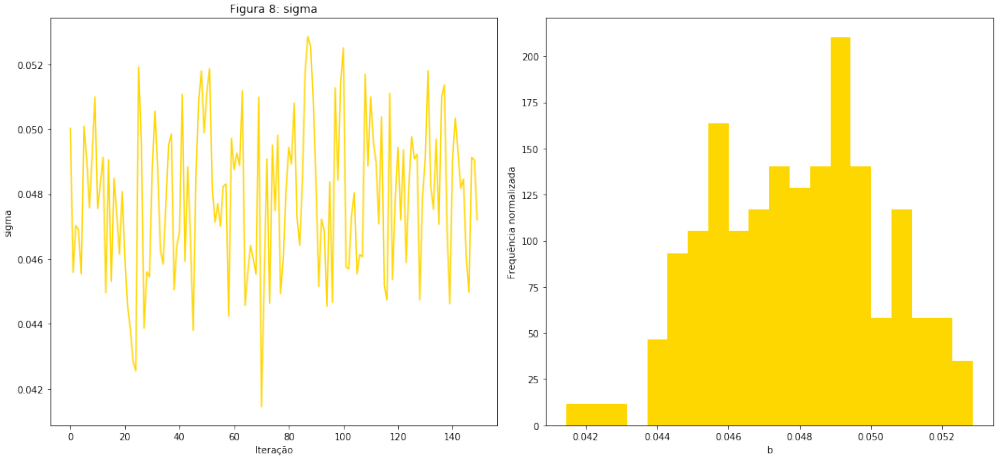

Figura 9: valores de σ e histograma destes

A Figura 10 mostra a "distribuição conjunta"de μ e σ. As partes mais claras represntam os θ's que mais apareceram nas ´ltimas alterações, o que é um indício de que estejam mais perto de θ^* (ver Figura 10).

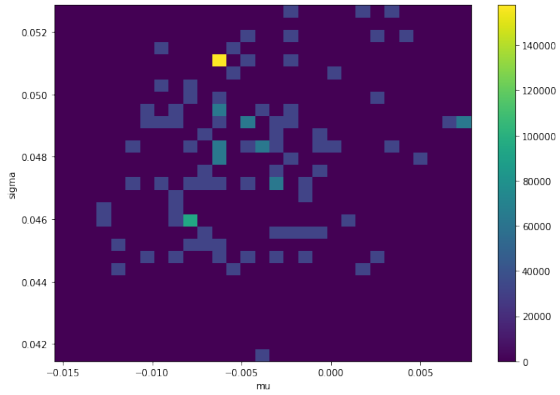

Figura 10: "distribuição"conjunta de μ e σ

E, finalmente, a Figura 11 mostra o "encaixe"entre os dados reais observados e uma predição baseada na estimação (dados gerados a partir da média dos últimos 50 θ's encontrados pelo algoritmo).

Afinal, é uma boa aproximação, provando a utilidade do método.

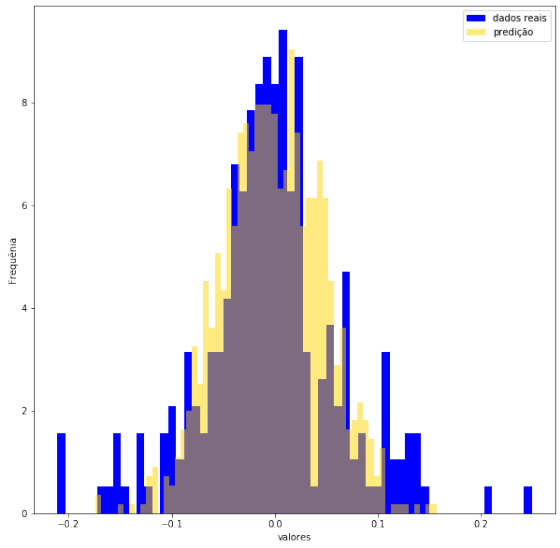

Figura 11: comparação entre predição e os dados reais observados

6 Aprimoramento do método

Fazendo uma pequena alteração no jeito como escolhemos um novo θ_1 podemos melhorar um pouco a eficiência do algoritmo. Fazendo $\mu_1 \sim (\mu_0, q(i))$, $\sigma_1 \sim (\sigma_0, q(i))$, com $q(i) = 1,005^{-1}$, i o número da iteração, conseguimos uma precisão maior sobre o θ^*.

O efeito que isso tem é que no começo, com i pequeno, os passos entre um θ e outro é grande, permitindo que se explore mais o espaço de parâmetros, porém quanto maior fica o i, mais pequeno fica a distância entre um θ e outro. O resultado é que ao final, o algoritmo fica muito mais minuscioso, o que é ótimo!

A seguir, as imagens mostram o mesmo processo feito anteriormente, porém com essa alteração, chute inicial $\theta = (7, 12)$ e 2000 iterações (ver Figuras 12 a 17).

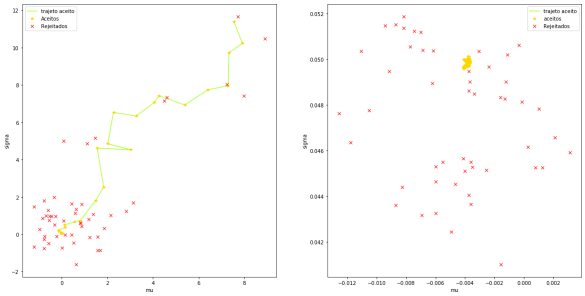

Figura 12: primeiras e últimas 50 alterações

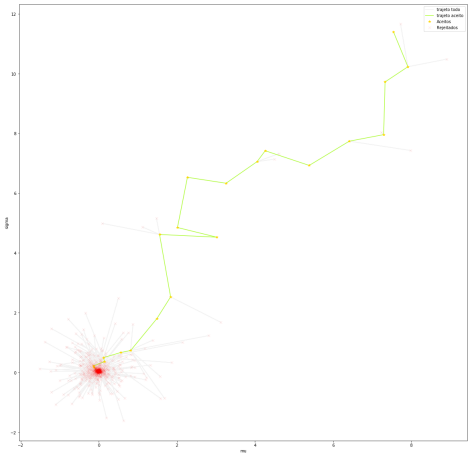

Figura 13: todas as iterações

Note como os primeiros passos parecem muito largos, ao passo que os últimos parecem minúsculos comparativamente. O algoritmo modificado é capaz de achar mais rápido um ponto ótimo. Além disso, veja como nas últimas iterações parece que que todas as alterações se acumulam num ponto.

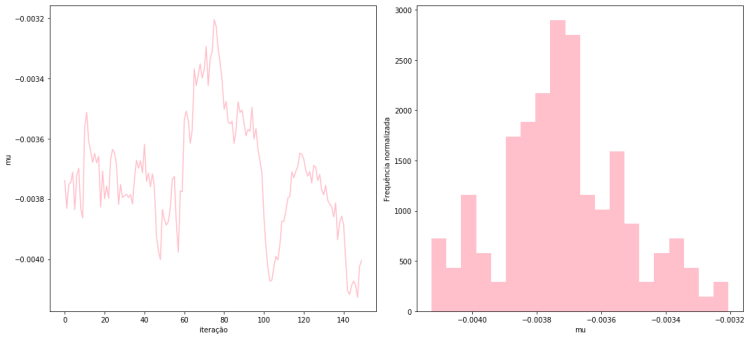

Figura 14: valores de μ e histograma destes

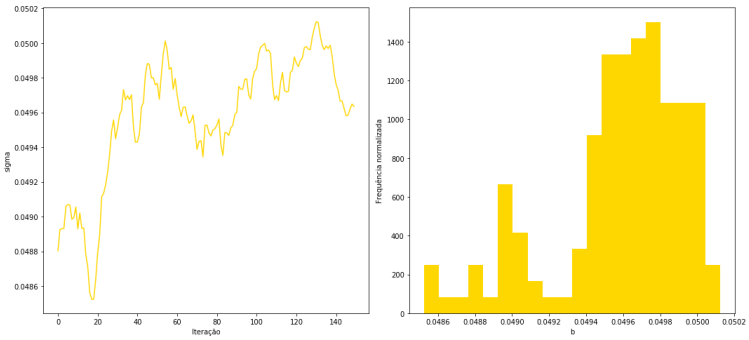

Figura 15: valores de σ e histograma destes

A predição feita ao final também é feita usando só a média dos últimos 10 θ's aceitos, uma vez que eles já estão bem acumulados (ver Figura 17).

Referências

[1] "Cotações e Boletins". Em: *bcb.gov.br* (). URL: `https://www.bcb.gov.br/estabilidadefinanceira/historicocotacoes`.

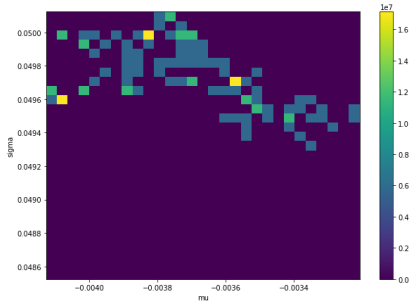

Figura 16: "distribuição"conjunta de μ e σ

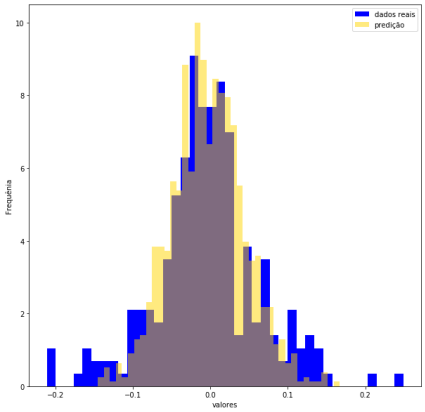

Figura 17: comparação entre predição e os dados reais observados