# AGA 0505 - Análise de Dados em Astronomia

## 4. Amostragem e Simulações

Laerte Sodré Jr.

1o. semestre, 2023

# aula de hoje: o método de Monte Carlo

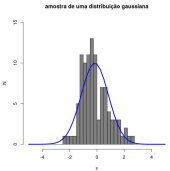
1. introdução: populações e amostras
2. o método de Monte Carlo
3. amostragem de distribuições de probabilidades
4. integração por Monte Carlo
5. simulação de Monte Carlo: esfera uniforme
6. MCMC- Markov Chain Monte Carlo
7. amostragem por bootstrap

*O registro de um mês de roleta em Monte Carlo nos dá material para discutir os fundamentos do conhecimento*

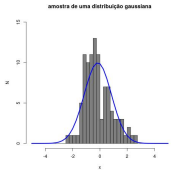
*Karl Pearson*

# variáveis aleatórias e amostragem

- variáveis aleatórias
  - variável (contínua ou discreta) cujos valores representam o resultado de fenômenos aleatórios
  - supõe-se que uma variável aleatória represente uma propriedade $x$ de uma **população** e obedeça a uma distribuição de probabilidades $P(x)$




- amostragem (*sampling*)
  - dessa população, extraímos uma **amostra** aleatória, representativa de $P(x)$
  - a distribuição da amostra é denominada *distribuição amostral*
  - é importante distinguir as propriedades da população e da amostra!



# propriedades da população e da amostra

- média e desvio padrão da população e da amostra:
  - considere uma população descrita por uma variável $x$ com distribuição normal, $N(\mu, \sigma)$
  - considere uma amostra de $n$ objetos obtida a partir dessa distribuição
  - média $\bar{x}$ da amostra:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

- desvio padrão não viesado da amostra:

$$\sigma_s = \left[ \frac{\sum (x_i - \bar{x})^2}{n-1} \right]^{1/2}$$

- $\mu, \sigma$: parâmetros da população
- $\bar{x}, \sigma_s$: parâmetros da amostra

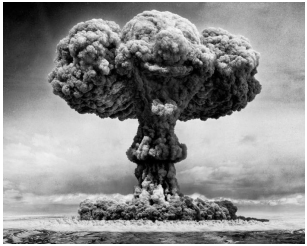
# o método de Monte Carlo

- método de resolução de problemas baseado em amostragem aleatória de distribuições de probabilidades

- inventado por Stanislaw Ulam, John von Neuman e Nicholas Metropolis durante o projeto Manhattan
  - Ulam (um dos que desenharam a bomba de hidrogênio) bolou o método em 1946, pensando nas probabilidades de se ganhar um jogo de cartas de paciência
  - Metropolis é o responsável pelo nome Monte Carlo

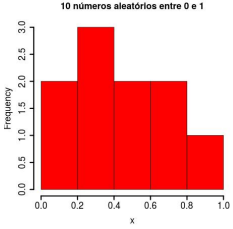
aplicações:

- amostragens de PDFs
- integração numérica
- otimização
- simulação de sistemas complexos
- ...



# amostragem de distribuições de probabilidades

- objetivo da amostragem:
  dada uma distribuição de probabilidades $P(x)$, gerar $N$ amostras $\{x_i\}$ distribuidas como $P(x)$
- exemplo: sequência de números aleatórios uniformemente distribuídos entre 0 e 1




- note que números gerados pelos "geradores de números aleatórios" são muitas vezes *pseudo-aleatórios*
- é possível produzir números aleatórios por "hardware":
  ex.: um dado não viesado pode gerar números aleatórios inteiros entre 1 e 6

# amostragem de distribuições de probabilidades por MC

- $u$: números aleatórios uniformemente distribuidos entre 0 e 1
- vamos supor que a fração de pontos gerados entre $u$ e $u + du$ seja igual a $P(x)dx$
- nesse caso, $du = P(x)dx$ e

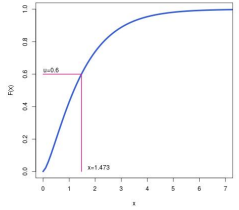
$$u = \int_{-\infty}^{x} P(x')dx' = F(x)$$

$F(x)$: distribuição cumulativa de $P(x)$

- MC: obtemos um $u_i$ uniformemente distribuido entre 0 e 1 e encontramos $x_i$ resolvendo a equação:

$$u_i - F(x_i) = 0$$

- note que $x_i = \text{quantil}(u_i)$
- repetimos este procedimento $N$ vezes para obter uma amostra de $N$ elementos
- ex.: $P(x) = x^{\alpha-1}e^{-x}/\Gamma(\alpha)$, $\alpha = 1.5$
  (distribuição gama com coeficiente de forma $\alpha$)
  se $u = 0.6$, resolvendo $u - F(x) = 0$
  temos que $x \simeq 1.473$



# exemplo: distribuição exponencial

- produção de uma amostra $\{x_i\}$ com $P(x) = e^{-x}$ $(x > 0)$
- distribuição cumulativa:

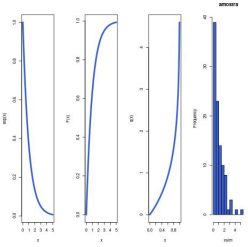
$$F(x) = \int_0^x e^{-x'} dx' = 1 - e^{-x}$$

- $u$: número aleatório uniformemente distribuido entre 0 e 1
- solução de $u - F(x) = 0$:

$$u - 1 + e^{-x} = 0 \longrightarrow x = -\ln(1-u)$$

(função quantil da distribuição exponencial)

- assim, dado um conjunto de $N$ números $u_i$ gerados uniformemente entre 0 e 1, calcula-se $x_i = -\ln(1-u_i) = -\ln \gamma_i$, onde $\gamma_i$ é também um número aleatório entre 0 e 1
- os $\{x_i\}$ resultantes estarão distribuídos com uma pdf exponencial

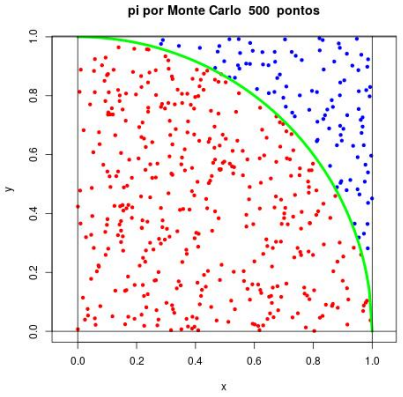


# exemplo: integração por Monte Carlo

- MC oferece uma forma simples para se integrar uma função positiva $f(x)$ por simulação numérica:
  $I = \int_a^b f(x)dx \qquad f(x) > 0$

- exemplo: cálculo de $\pi$
  - a área de um quarto de círculo unitário é $\pi/4$
  - sorteamos $N$ pontos uniformemente para $x$ entre 0 e 1 e para $y$ entre 0 e 1
  - podemos estimar $\pi$ como $\hat{\pi} \simeq 4\frac{N_{ac}}{N}$, onde $N_{ac}$ é o número de pontos que caem dentro do quarto de círculo

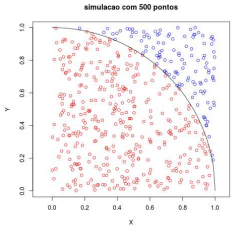
$f(x) = \sqrt{x^2 + y^2} \qquad (0 < x, y < 1)$

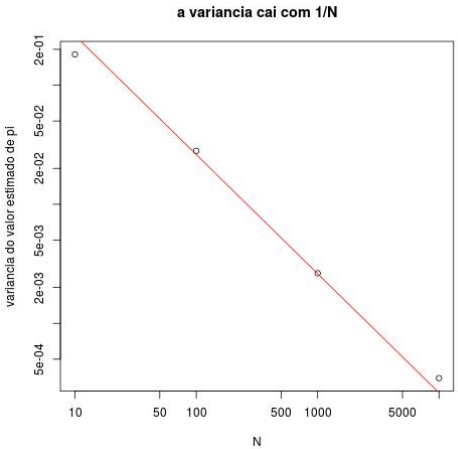



$(\hat{\pi} \simeq 3.16)$

# variância nas estimativas por MC

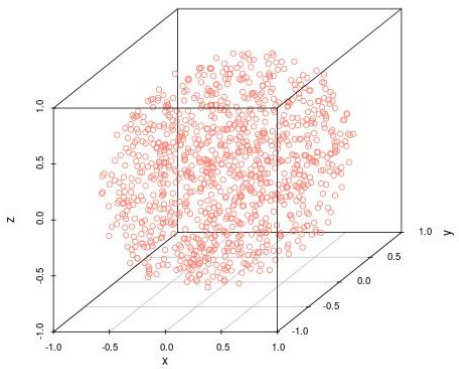
- note que a variância $V$ nos métodos de MC cai com 1/N
- o desvio padrão $\sigma_s$ é a raiz quadrada da variância: $\sigma_s = V^{1/2}$
- para reduzir $\sigma_s$ por um fator 2 deve-se multiplicar o número de simulações por um fator 4



# simulações de Monte Carlo

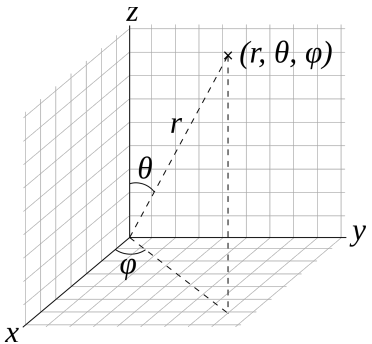
- o método de Monte Carlo é amplamente usado para simular sistemas, processos, etc
- a simulação é feita amostrando-se as variáveis aleatórias do modelo com as distribuições de probabilidade apropriadas
- exemplo: simulação de uma esfera centrada na origem com densidade uniforme de pontos- como simular $N$ coordenadas $(x, y, z)$ com distribuição uniforme?
- regra de ouro: *faça um modelo probabilístico que inclua todas as probabilidades relevantes*



# exemplo de simulação: esfera uniforme

- objetivo: simular uma distribuição uniforme de pontos dentro de uma esfera centrada na origem
- parâmetros: $N,\ R$
- o melhor é fazer a simulação em coordenadas esféricas $(r, \theta, \phi)$ e daí obter $(x, y, z)$:
  $x = r\,\mathrm{sen}(\theta)\cos(\phi)$
  $y = r\,\mathrm{sen}(\theta)\,\mathrm{sen}(\phi)$
  $z = r\cos(\theta)$
  com $0 \leq r \leq R$, $0 \leq \theta \leq \pi$ e $0 \leq \phi \leq 2\pi$

- problema: qual é a distribuição da população de $r,\ \theta,\ \phi$?
- as variáveis $(r,\ \theta,\ \phi)$ são independentes: $P(r, \theta, \phi) = P(r)P(\theta)P(\phi)$



# exemplo de simulação: esfera uniforme

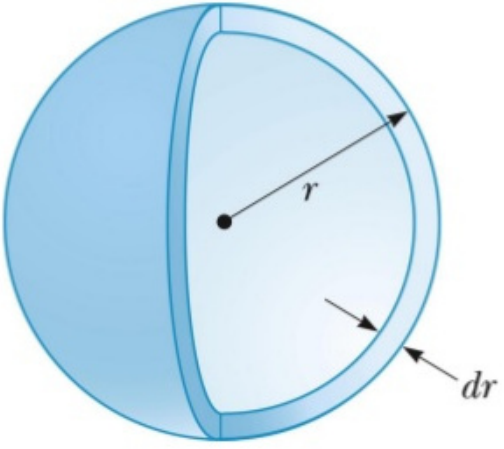
- densidade média da esfera: $n = 3N/(4\pi R^3)$
- $P(r)dr$: probabilidade de se encontrar uma galáxia entre os raios $r$ e $r + dr$:

  $P(r)dr \propto ndV \propto n4\pi r^2 dr \rightarrow P(r) \propto r^2$

  $\int_0^R P(r)dr = 1 \rightarrow P(r) = \frac{3r^2}{R^3}$
- função cumulativa:

  $F(r) = \int_0^r P(r')dr' = \left(\frac{r}{R}\right)^3$
- MC: se $u_r$ é um número aleatório uniformemente distribuído entre 0 e 1,

  $F(r) = u_r \rightarrow r = Ru_r^{1/3}$



# exemplo de simulação: esfera uniforme

- elemento de ângulo sólido:

$$d\Omega = \operatorname{sen}\theta d\theta d\phi$$

$$0 \leq \theta \leq \pi; \quad 0 \leq \phi \leq 2\pi$$

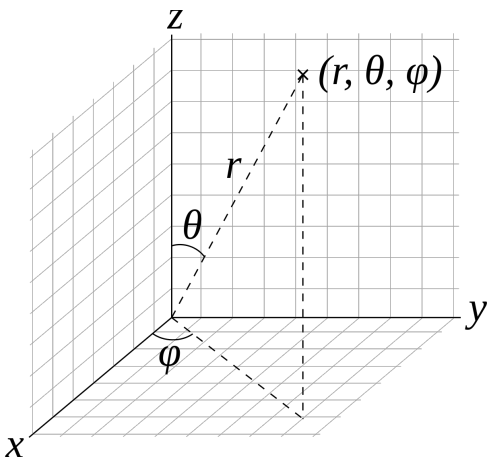
- probabilidade conjunta de $\theta$ e $\phi$:

$$P(\theta, \phi) d\theta d\phi = d\Omega / 4\pi$$

- marginalizando:

$$P(\theta) = \int_0^{2\pi} P(\theta, \phi) d\phi = \frac{1}{2} \operatorname{sen}\theta$$

$$P(\phi) = \int_0^{\pi} P(\theta, \phi) d\theta = \frac{1}{2\pi}$$



# exemplo de simulação: esfera uniforme

- funções cumulativas:

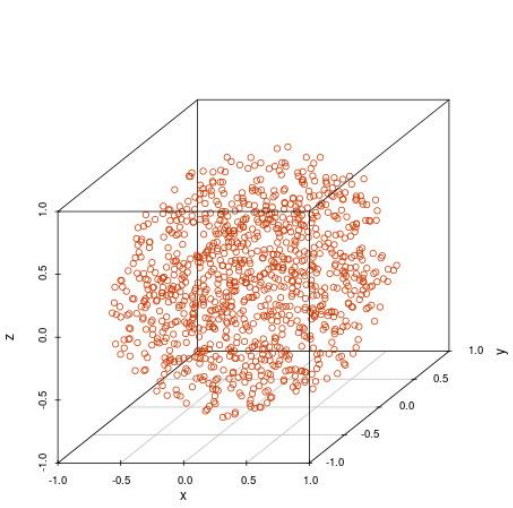
$$F(\phi) = u_\phi = \int_0^\phi \frac{d\phi}{2\pi} = \frac{\phi}{2\pi}$$
$$\longrightarrow \phi = 2\pi u_\phi$$
$$F(\theta) = u_\theta = \int_0^\theta \frac{1}{2}\,\mathrm{sen}\theta d\theta = \frac{1}{2}(1-\cos\theta)$$
$$\longrightarrow \theta = \mathrm{acos}(1-2u_\theta)$$

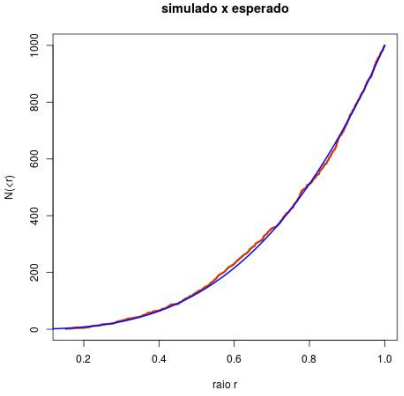
$(u_r, u_\theta, u_\phi)$: números aleatórios uniformemente distribuídos entre 0 e 1

- simulação de um ponto:
  gero $(u_r, u_\theta, u_\phi)$ e calculo:
  $r = R u_r^{1/3}$
  $\theta = \mathrm{acos}(1-2u_\theta)$
  $\phi = 2\pi u_\phi$

  e
  $x = r\,\mathrm{sen}(\theta)\cos(\phi)$
  $y = r\,\mathrm{sen}(\theta)\,\mathrm{sen}(\phi)$
  $z = r\cos(\theta)$

# exemplo de simulação: esfera uniforme




$$N(<r) = \frac{4}{3}\pi r^3 n = N\left(\frac{r}{R}\right)^3$$

# Markov Chain Monte Carlo

# Markov Chain Monte Carlo

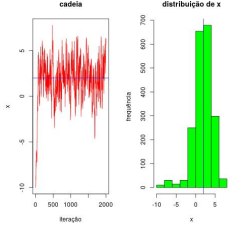
- método eficiente para amostrar distribuições de probabilidades complexas!
- em muitos casos não dá para integrar $P(x)$ para amostrar $x$ por MC simples: nesses casos adota-se MCMC
- baseado na mecânica estatística: Metropolis et al. (1953): algoritmo para determinar a equação de estado de um conjunto de partículas em interação dentro de uma caixa

THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

**Equation of State Calculations by Fast Computing Machines**

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
*Los Alamos Scientific Laboratory, Los Alamos, New Mexico*
AND
EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*
(Received March 6, 1953)

- objetivo do MCMC: amostrar $P(x)$
- $x$ pode ser um vetor
- processo iterativo- o algoritmo gera uma sequência de amostras $\mathbf{X} = \{x_1, x_2, ..., x_n\}$
- $\mathbf{X}$ forma uma cadeia de Markov: $x_{i+1}$ depende apenas de $x_i$



# Markov Chain Monte Carlo

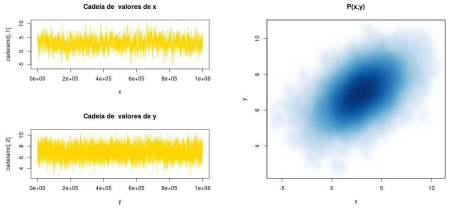
- MCMC gera uma sequência de amostras $\mathbf{X} = \{x_1, x_2, ..., x_n\}$ que forma uma cadeia de Markov: $x_{i+1}$ depende apenas de $x_i$
- dado $x_i$, como obter $x_{i+1}$?
- o algoritmo tem duas partes:
  1. dado um $x_i$, uma *função de propostas* $q(x'|x_i)$ propõe um novo ponto $x'$
  2. adota-se um critério baseado em $P(x')/P(x_i)$ para se aceitar ou não a proposta
     se a proposta é aceita, $x_{i+1} = x'$,
     se não, $x_{i+1} = x_i$
- função de propostas:
  - obedece ao *princípio do balanceamento detalhado*:
    $q(x'|x) = q(x|x')$
    é tão fácil ir de $x$ a $x'$ quanto de $x'$ a $x$
  - exemplo: gaussiana
    $q(x'|x_i) \sim N(\mu = x_i, \sigma)$
- aceitação ou não de $x'$:
  - $u$: número aleatório uniformemente distribuido entre 0 e 1
  - se $P(x')/P(x_i) > u$, aceita-se $x'$ e $x_{i+1} = x'$
  - se não, $x_{i+1} = x_i$

# o algoritmo de Metropolis-Hastings

- **algoritmo de Metropolis-Hastings**:
  dado $x_i$, para se obter $x_{i+1}$:
  - obtenha uma proposta $x'$ usando $q(x'|x_i)$
  - obtenha um número aleatório $0 < u < 1$ com distribuição uniforme
  - se $P(x')/P(x_i) > u$, $x_{i+1} = x'$
  - se não, $x_{i+1} = x_i$
- tendo uma cadeia de amostras $\mathbf{X} = \{x_1, x_2, ...\}$, podemos estimar a distribuição de $x$ e estatísticas de interesse:

$$E[x] \simeq \frac{1}{N}\sum_{k=1}^{N} x_k \qquad E[f(x)] \simeq \frac{1}{N}\sum_{k=1}^{N} f(x_k)$$

$f(x)$: função arbitrária de $x$



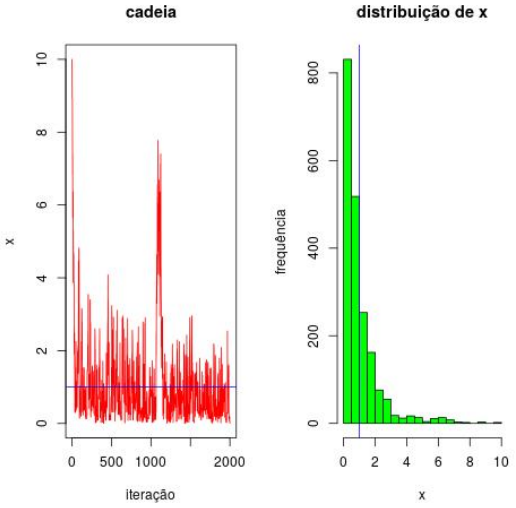
# exemplo: amostragem de uma distribuição exponencial

- amostragem de $P(x) = e^{-x}$, $x > 0$
- defino a função de propostas: por exemplo

$$q(x, x_i) \sim N(\mu = x_i, \sigma = 1)$$

note que neste exemplo $x$ deve ser maior que 0!

- inicialização do algoritmo:
  defino $x_0$ e o número de iterações $N$
- gero a cadeia $\mathbf{X} = \{x_1, x_2, ...\}$ com Metropolis-Hastings
- analiso a convergência
- elimino as cadeias iniciais (*burn-in*)
- analiso os resultados



# mcmc: alguns detalhes

- a escala do problema é importante: muitas vezes é melhor amostrar o logaritmo de $P(x)$
- *burn-in*: o algoritmo precisa ser inicializado; muitas vezes o ponto inicial não é "típico" e várias iterações são necessárias antes de se poder fazer inferências estatísticas
- essas iterações iniciais são denominadas *burn-in* e devem ser removidas das análises estatísticas
- devido à natureza markoviana do método, amostras próximas são correlacionadas: uma boa amostragem de $P(x)$ pode exigir longas cadeias de amostras
- em alguns casos, onde a função é multi-modal, pode ser necessário se rodar várias cadeias, cada uma começando de um ponto diferente
- até onde se deve rodar uma cadeia? usar *diagnósticos de convergência* e analisar graficamente os resultados

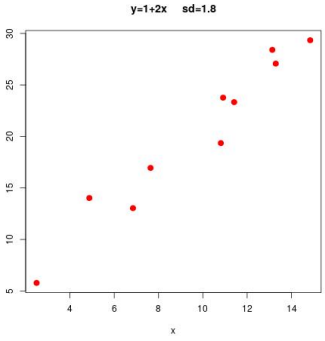
# *bootstrap*: simulando dados com os próprios dados

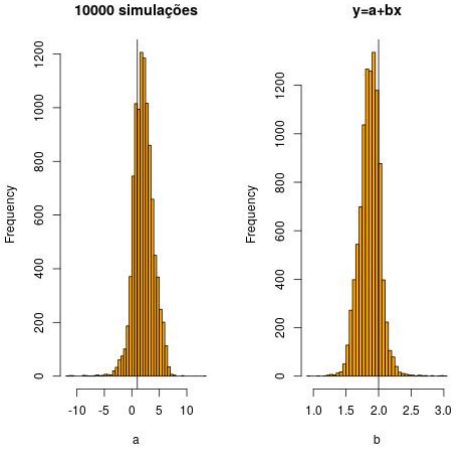
- técnica baseada em reamostragem dos dados (Efron, 1979)
- suponha que tenhamos um conjunto de dados e queremos determinar a distribuição de alguma estatística $w$, determinada a partir desses dados
- exemplo: temos um conjunto de pontos $(x, y)$ e queremos ajustar uma reta a eles, $y = a + bx$
- *bootstrap* permite determinar os erros e intervalos de confiança para os parâmetros $(a, b)$

- ideia básica do *bootstrap*:
  - seja $\mathcal{D}_0$ o conjunto de dados
  - um novo conjunto de dados $\mathcal{D}_i$ é simulado a partir de $\mathcal{D}_0$ por reamostragem *com substituição*
  - a estatística $w_i$ é determinada a partir desses dados simulados
  - pode-se fazer isso muitas vezes e assim obter-se muitas amostras de $w$
  - estas amostras podem então ser usadas para estimar os erros e intervalos de confiança de $w$

# bootstrap: incertezas nos parâmetros de um modelo

- exemplo: ajuste de uma reta: $y = a + bx$
- estimativas de $a$ e $b$ em 10000 simulações



# referências úteis


- A Primer for the Monte Carlo Method - Ilya Sobol (disponível online)
- Data analysis recipes: Using Markov Chain Monte Carlo - D. W. Hogg & D. Foreman-Mackay- arXiv:1710.06068
- A Conceptual Introduction to Markov Chain Monte Carlo Methods - J. Speagle- arXiv:1909.12313
- Convergence diagnostics for Markov chain Monte Carlo - Vivekananda Roy - arXiv:1909.11827