Problema 1

Simular $n = 5$ y $n = 35$ v.a Bernoulli $Be(1/3)$; sea r el número de éxitos en cada caso.

Simulamos n ensayos bernoulli con probabilidad $p = 1/3$ usando la librería **scipy** y encontramos el número de exitos al sumar los resultados con **np.sum()**.

Algorithm 1: Simular

```
# Simulated sum of n bernoulli random variables
r = np.sum(stats.bernoulli.rvs(size=n, p=p))

# Plot binomial trial
print('n = {}, p = {}, r(success) = {}'.format(n, p, r))
```

Problema 2

Implementar el algoritmo Metropolis-Hastings para simular de la posterior

$$f(p|\bar{x}) \propto p^r(1-p)^{n-r}\cos(\pi p)\mathcal{I}_{[0,1/2]}(p),$$

con los dos casos de n y r de arriba. Para ello poner la propuesta $(p'|p) = p' \sim Beta(r+1, n-r+1)$ y la distribución inicial de la cadena $\mu \sim U(0, 12)$.

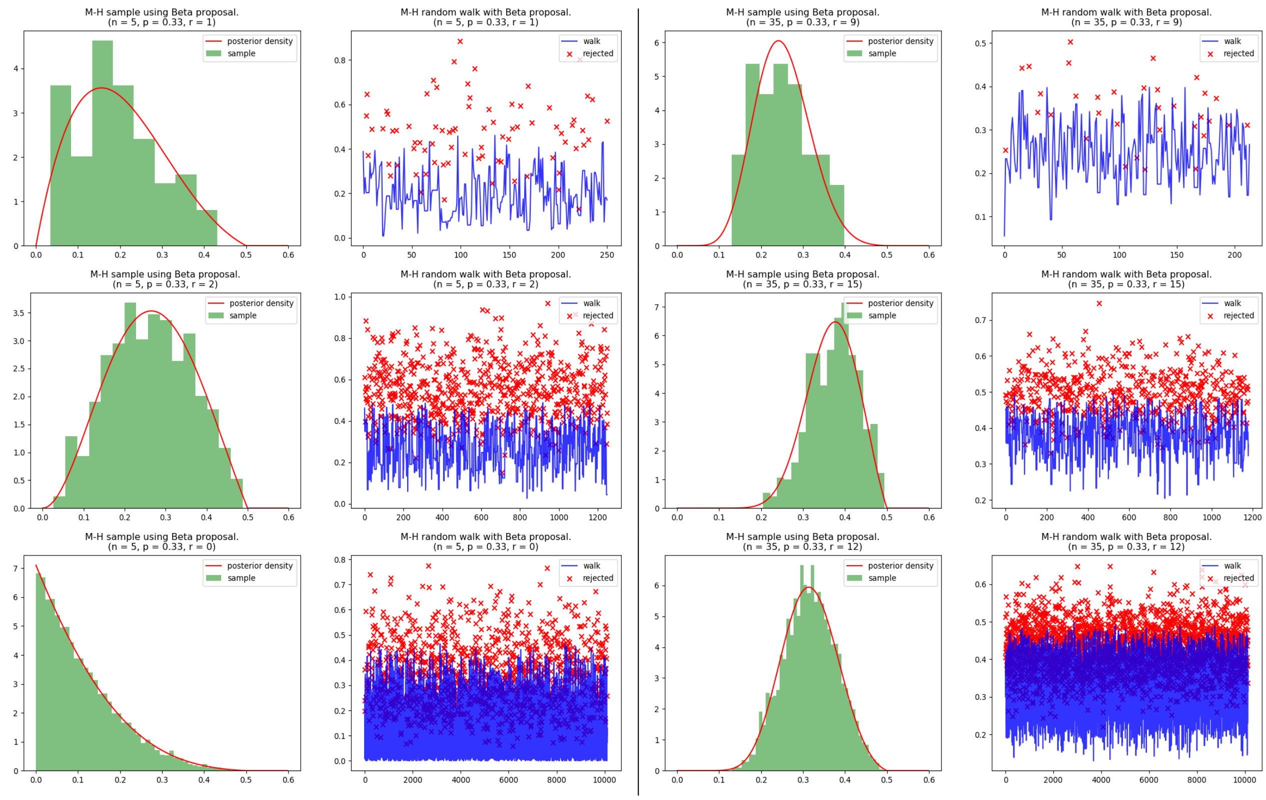
A continuación presentamos un extracto de la implementación en python del algoritmo.

Algorithm 2: Implementación en python del algoritmo Metropolis-Hastings. Extracto del archivo **metropolis_hastings.py**.

```
# Sample until desired number of samples generated
while len(sample) < sample_size:

    # Generate random step from proposed distribution
    x_p = step()

    # Calculate the acceptance ratio
    alpha = acceptance_ratio(x_p, x_t, posterior, proposal)

    # Random uniform sample
    u = np.random.uniform()

    if u < alpha:  # Accept
        x_t = x_p

        # Burn-in
        ...
```

```
# Add sample
sample.append(x_t)

# Next step
t += 1
```

Estudiemos los resultados de las aproximaciónes obtenidas a la distribución posterior al ejecutar el algoritmo de Metropolis-Hastings para 3 tamaños de muestra $N = 100, 1000, 10000$ habiendo hecho burn-in de 100 iteraciones. Las gráficas de la caminata aleatoria muestran los todos los estados de la cadena y los propuestas que fueron rechazadas.



Notemos que la distribución de la muestra se comienza a parecer mucho en el histograma a la densidad (posterior) real cuando el tamaño de la muestra crece. Podemos ver que el soporte de la propuesta *Beta* es $(0, 1)$ de modo que cubre todo el soporte de la densidad posterior, que es $(0, 1/2)$. Como la distribución que escogimos $Beta(r + 1, n - r + 1)$ no es simétrica, claramente podemos ver que los únicos rechazos ocurrieron del "lado" derecho de la media (arriba cuando graficamos respecto al tiempo). Es sorprendete que solo fue necesario evaluar la densidad que queremos simular en diferentes puntos.

Problema 3

Argumentar porque la cadena es *f-irreducible* y porque es ergódica. Implementar el algoritmo con los datos descritos y discutir los resultados.

Primero presentamos el pseudo-código del algoritmo de Metropolis-Hastings tomadolo de la página 270 del libro *Monte Carlo Statistical Methods* [1].

Algorithm 1 Metropolis-Hastings

1: Dado $x^{(t)}$,
2: Generar $Y_t \sim q(y|x^{(t)})$.
3: Tomar,

$$X^{(t+1)} = \begin{cases} Y_t & \text{con probabilidad } \rho(x^{(t)}, Y_t), \\ x^{(t)} & \text{con probabilidad } 1 - \rho(x^{(t)}, Y_t), \end{cases}$$

donde

$$\rho(x^{(t)}, Y_t) = min \left\{ 1, \frac{f(y)}{f(x)} \frac{q(x|y)}{q(y|x)} \right\}$$

Notemos que en cada paso la probabilidad de moverse a un estado $y \in S_{q(.|x)}$ el soporte de $q(.|x^{(t)})$, es igual a

$$\begin{aligned} P[X_{n+1} = y | X_n = x] &= q(y|x) \cdot \rho(x, y), \\ &= q(y|x) \cdot \min \left\{ 1, \frac{f(y)}{f(x)} \frac{q(x|y)}{q(y|x)} \right\}, \\ &= \min \left\{ q(y|x), \frac{f(y)}{f(x)} q(y|x) \right\}, \end{aligned} \tag{1}$$

puesto que estamos juntando las probabilidades de las etapas de **propuesta** y de **aceptación-rechazo**.

Ahora desmostraremos que es *f-irreducible*. Sea $\mathcal{X}$ el espacio de estados posibles de la cadena de Markov (X_n) determinada por el algoritmo de Metropolis-Hastings. Supongamos que el soporte $S_f = \{x \in \mathcal{X} | f(x) > 0\}$ de la distribución **posterior** f está contenido en el soporte $S_{q(\cdot|x)}$ de la distribución **propuesta** $q(\cdot|x)$. Siendo $A \in Borel(\mathcal{X})$ un posible evento de la cadena de Markov tal que $f(A) > 0$ y $x \in \mathcal{X}$ un estado arbitrario, tenemos que

$$P[X_{n+1} \in A | X_n = x] = \min \left\{ q(A|x), \frac{f(A)}{f(x)} q(x|A) \right\} > 0 \tag{2}$$

puesto que $f(A) > 0$ implica que $q(A|x) > 0$, ya que supusimos que el soporte de $f(.)$ está contenido en el de $q(.|x)$. Demostrando que la cadena de Markov es $f-irreducible$.

Para demostrar que (X_n) es ergódica, tenemos que verificar que sea **aperiodica** y **positiva recurrente**.

La cadena (X_n) es **aperiodica** puesto que la probabilidad de regresar al estado actual en un paso es mayor que cero, osea que,

$$P[X_{n+1} = x^{(t)}|X_n = x^{(t)}] = q(y|x) \cdot (1 - \rho(x^{(t)}, y)) > 0 \tag{3}$$

Esto lo sabemos "intuitivamente" por que queremos que el algortimo algunas veces rechaze las propuesta y, osea que $1 - \rho(x, y) > 0$. Podemos verificar esto en nuestra construcción particular, para alguna propuesta con distribución beta $y \in (0, 1)$, tenemos que

$$\begin{aligned}\rho(x, y) &= min\left\{1, \frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}\right\}\\ &= min\left\{1, \frac{y^r(1-y)^{n-r}\cos(\pi y)\mathcal{I}_{[0,1/2]}(y)}{x^r(1-x)^{n-r}\cos(\pi x)\mathcal{I}_{[0,1/2]}(x)}\frac{x^{\alpha-1}(1-x)^{\beta-1}}{y^{\alpha-1}(1-y)^{\beta-1}}\right\}\\ &= min\left\{1, \frac{\cos(\pi y)}{\cos(\pi x)}\right\}\end{aligned} \tag{4}$$

Y podemos encontrar una propuesta en $y \in (0, 1)$ tal que $\cos(\pi y) < \cos(\pi x)$, por lo que $1 - \rho(x, y) > 0$. Y como el periodo de la cadena se define como $k = \gcd\{n > 0 : \Pr(X_n = i \mid X_0 = i) > 0\}$ y el periodo es $k = 1$, entonces la cadena es **aperiodica**. Así que sin importar el estado de la cadena es posible regresar a cierto estado anterior x, pero no necesariamente ocurre esto con un periodo especifico. Notemos que como la probabilidad de regresar a un estado anterior es positiva (por que podemos quedarnos en el mismo estado en el siguiente paso $X_{n+1} = x_n$), entonces la cadena también es *recurrente*.

Una cadena (X_n) es **postiva recurrente** si se puede asegurar que el número de pasos necesarios para regresar a un estado $x \in \mathcal{X}$ es finito (la esperanza del tiempo de retorno es finita). Ya sabemos que la cadena es recurrente y también es positiva por que la probabilidad de ir a cualquier estado y en el soporte de $q(.|x)$ desde cierto estado x es siempre mayor que cero, $q(y|x) > 0$. Esto se formaliza con la proposicón 6.36 y la definición 6.35,mencionado en la página 273 del libro *Monte Carlo Statistical Methods* [1].

Verificando que la cadena es **f-irreducible** y **ergódica**. Con estas propiedades subsecuentemente se demuestra que efectivamente la distribución estacionaria de la cadena tendra densidad f y esta distribución estacionaria es única.

Problema 4

Implementar el algoritmo Metropolis-Hastings con la posterior de arriba tomando una propuesta diferente.

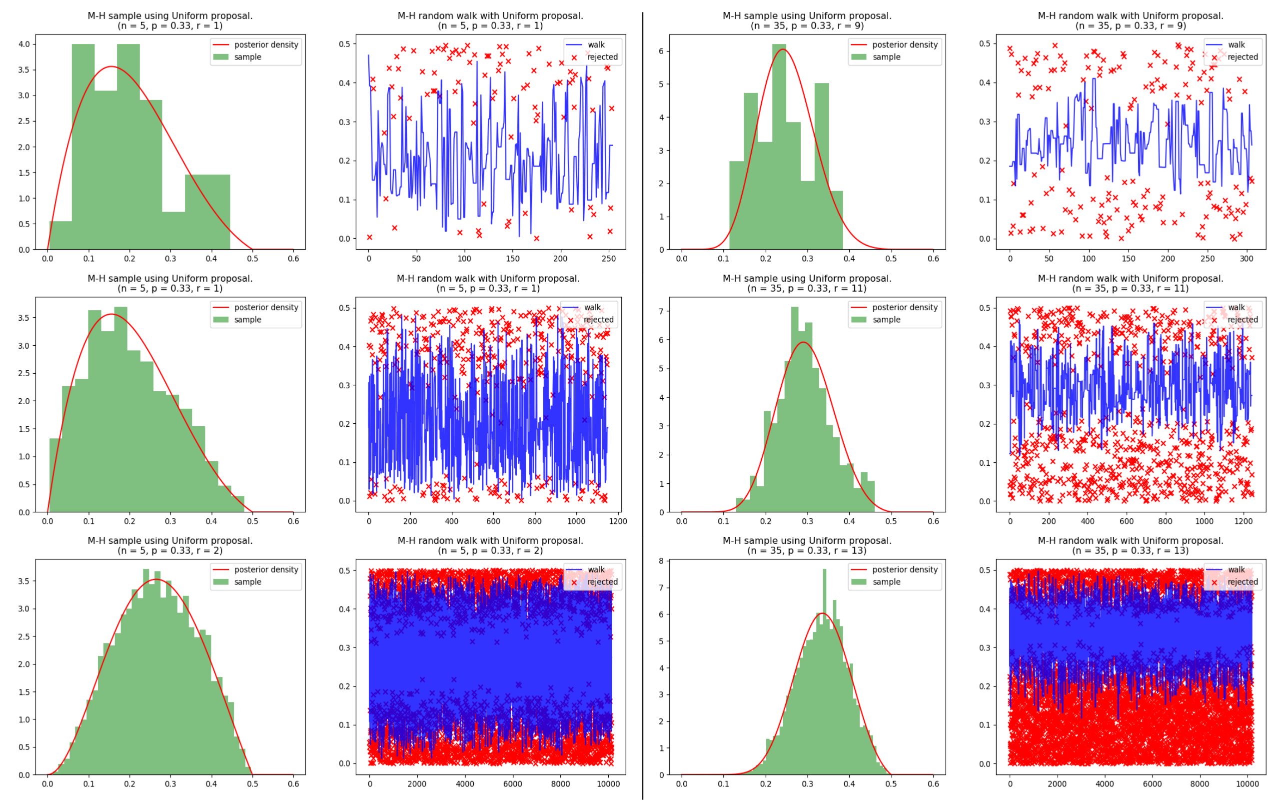
Consideraremos dos propuestas, una con distribución **uniforme** en el rango $(0, \frac{1}{2})$ y otra con distribución **normal** con media $\mu = x^{(t)}$ igual al estado actual de la cadena y una varianza fija σ^2. Formalizemos lo anterior.

Propuesta uniforme

En este caso utilizamos la propuesta con distribución uniforme en el rango $(0, 1/2)$.

$$(p'|p) = p' \sim \mathcal{U}(0, 1/2) \tag{5}$$

Estudiemos los resultados de las aproximaciónes obtenidas a la distribución posterior al ejecutar el algoritmo de Metropolis-Hastings para 3 tamaños de muestra $N = 100, 1000, 10000$ habiendo hecho burn-in de 100 iteraciones. Las gráficas de la caminata aleatoria muestran los todos los estados de la cadena y las propuestas que fueron rechazados.



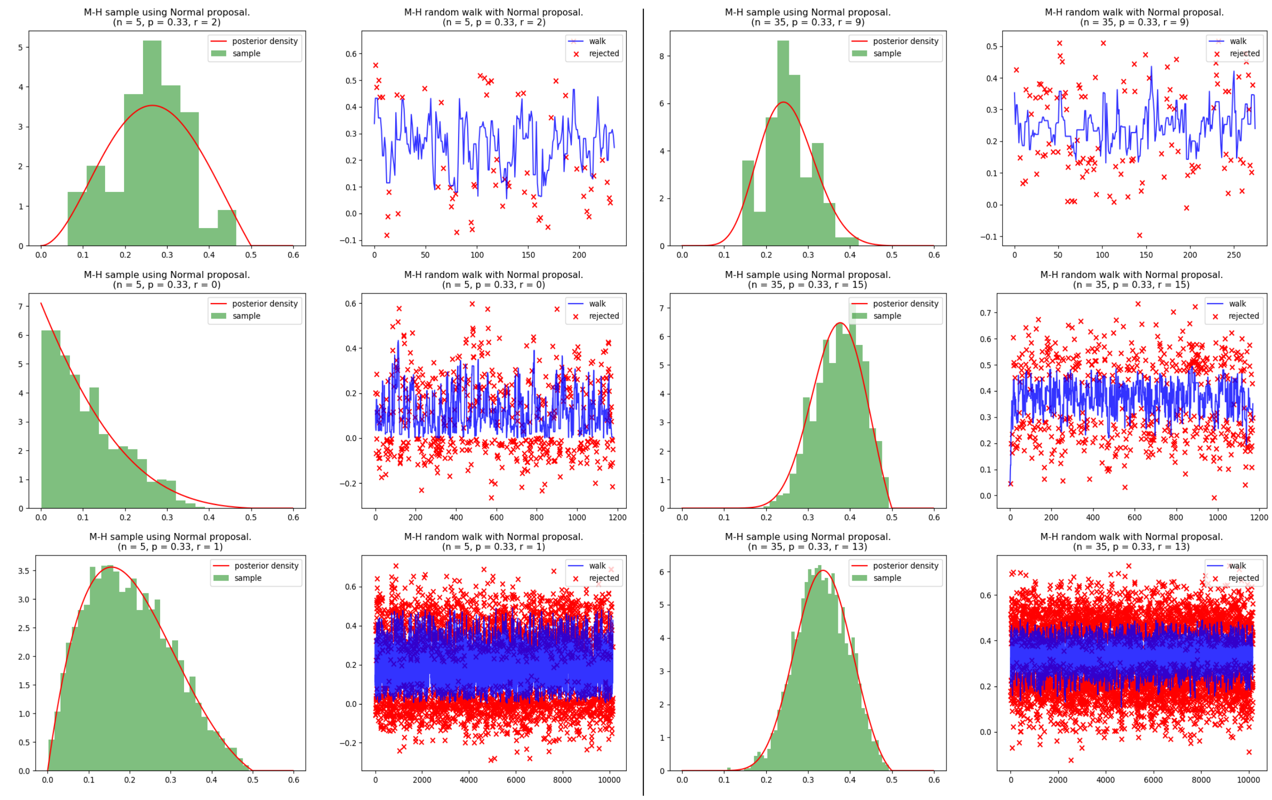
Es claro que la distribución de todas la propuestas es uniforme porque los puntos acceptados y rechazados en la gráfica para $N = 10000$ se reparten igual en toda la gráfica. Notamos que hay dos nubes separadas de puntos rechazados alrededor de la "caminata". Como comenzamos el algoritmo con un punto dentro del soporte de $S_f = (0, 1/2)$, no se ve un gran salto inicial para que se perciba un comportamiento donde ya se convergio a la distribución estacionaria.

Propuesta normal

En este caso utilizamos la propuesta con distribución normal alrededor del estado actual $x^{(t)} = p$ de la cadena,

$$(p'|p) \sim \mathcal{N}(p, \sigma^2) \tag{6}$$

Siendo una caminata aleatoria con un "paso" normal. Fijamos $\sigma^2 = 0.1$ como un parámetro externo al modelo. Estudiemos los resultados de las aproximaciónes obtenidas a la distribución posterior al ejecutar el algoritmo de Metropolis-Hastings para 3 tamaños de muestra $N = 100, 1000, 10000$, habiendo hecho burn-in de 100 iteraciones. Las gráficas de la caminata aleatoria muestran los todos los estados de la cadena y las propuestas que fueron rechazadas.



Este ejemplo es diferente a los otros en el sentido de que la distribución propuesta es simétrica en el sentido que $q(x|y) = q(y|x)$. También por que la propuesta ya no es independiente del estado acutal y en realidad parece más una caminata aleatoria, en vez de asemejarse al método de Aceptación-Rechazo. No percibimos visualmente una diferencia grande entre las muestras resultantes pues todas se asemejan a la densidad esperada con un tamaño de muestra grande. Una diferencia con la propuesta uniforme es que parece que hay muchos menos rechazos en este caso.

En nuestra implementación de nuestro algoritmo rechazamos inmediatamente a un propuesta fuera del soporte de la posterior. Como esta propuesta es la única de las presentadas que generara valores menores que cero, podemos observar la claramente estos rechazos en el ejemplo $n = 5$ con tamaño de muestra 1000. Notamos que hay dos nubes separadas de puntos rechazados alrededor de la "caminata". Como comenzamos el algoritmo con un punto dentro del soporte de $S_f = (0, 1/2)$, no se ve un gran salto inicial para que se perciba un comportamiento donde ya se convergio a la distribución estacionaria.

Referencias

[1] (Springer Texts in Statistics) - Monte Carlo Statistical Methods (2004), Christian Robert, George Casella