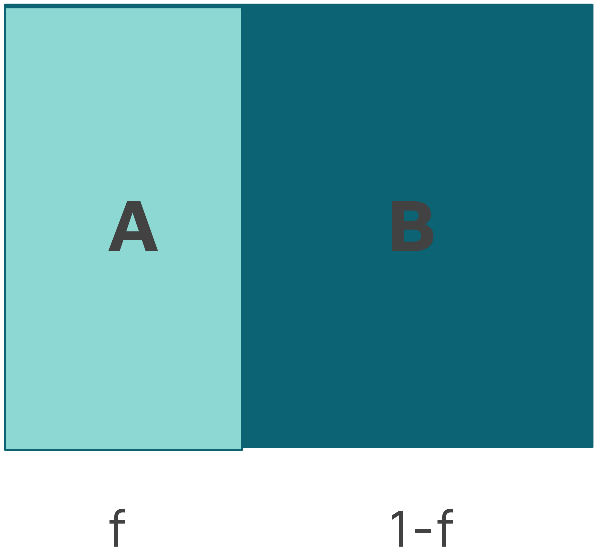
Projeto 1 de SME0123

Problema 5: uma enquete eleitoral

Cenário



Entrevista

- 3 perguntas
- Respostas quantitativas
- Cada resposta de eleitores de A tem distribuição normal centrada em 0
- Cada resposta de eleitores de B tem distribuição normal centrada em 1
- O desvio padrão é igual para ambos

Logo, todas as perguntas têm a mesma distribuição.

Primeira questão

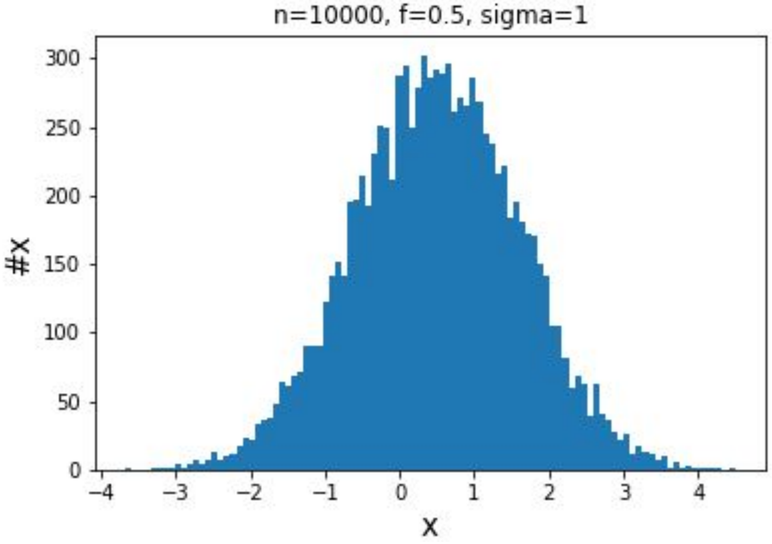
- X: respostas da primeira pergunta
- Y: respostas da segunda pergunta
- Z: respostas da terceira pergunta
- Ax: respostas da primeira pergunta dos eleitores de A
- Bx: respostas da primeira pergunta dos eleitores de B

- Para f parcela da população, X ~ N(0, sigma). Para f-1, X ~ N(1, sigma).
- Na curva teórica, X = f*Ax + (f-1)*Bx
- Na amostragem, para cada resultado aleatório da normal deve ser sorteado um μ (de acordo com um experimento de Bernoulli com chance f-1)

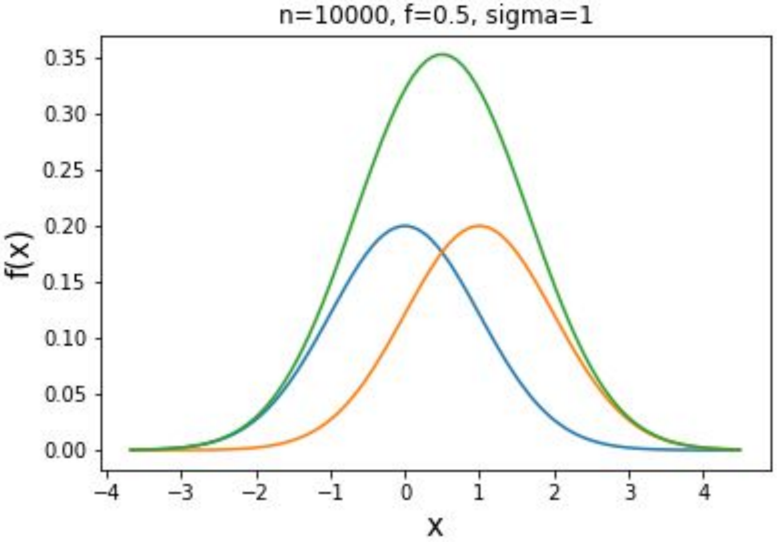
Código

```
x_amostra = [np.random.normal(np.random.binomial(1, 1-f), scale=sigma) for i in range(n)]
k = np.linspace(np.min(x_amostra), np.max(x_amostra), 100)
plt.hist(x_amostra, bins=k)
```

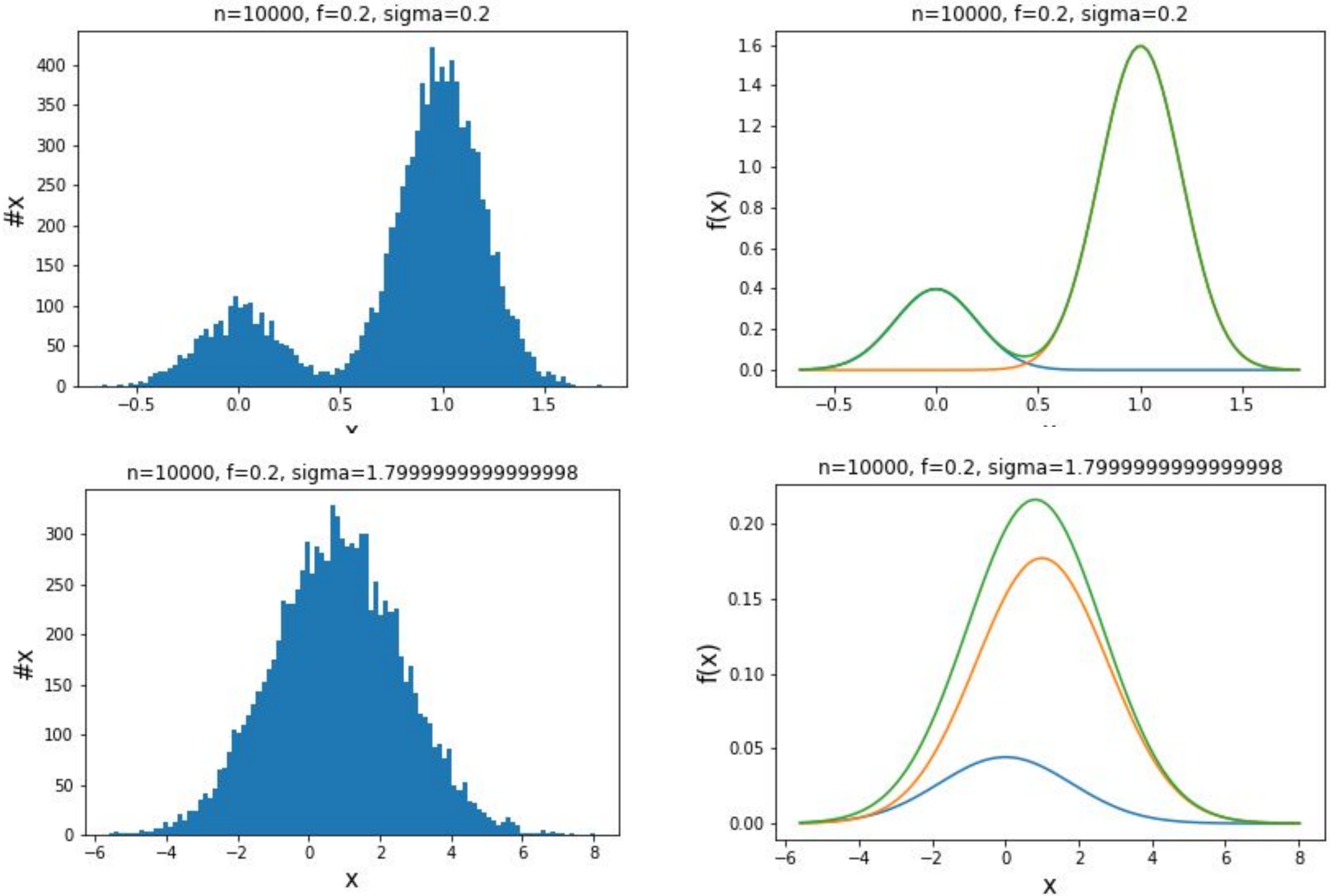
```
a = f*scipy.stats.norm.pdf(k, 0, sigma)
b = (1-f)*scipy.stats.norm.pdf(k, 1, sigma)
s = [x + y for x, y in zip(a, b)]
plt.plot(k, a)
plt.plot(k, b)
plt.plot(k, s)
```

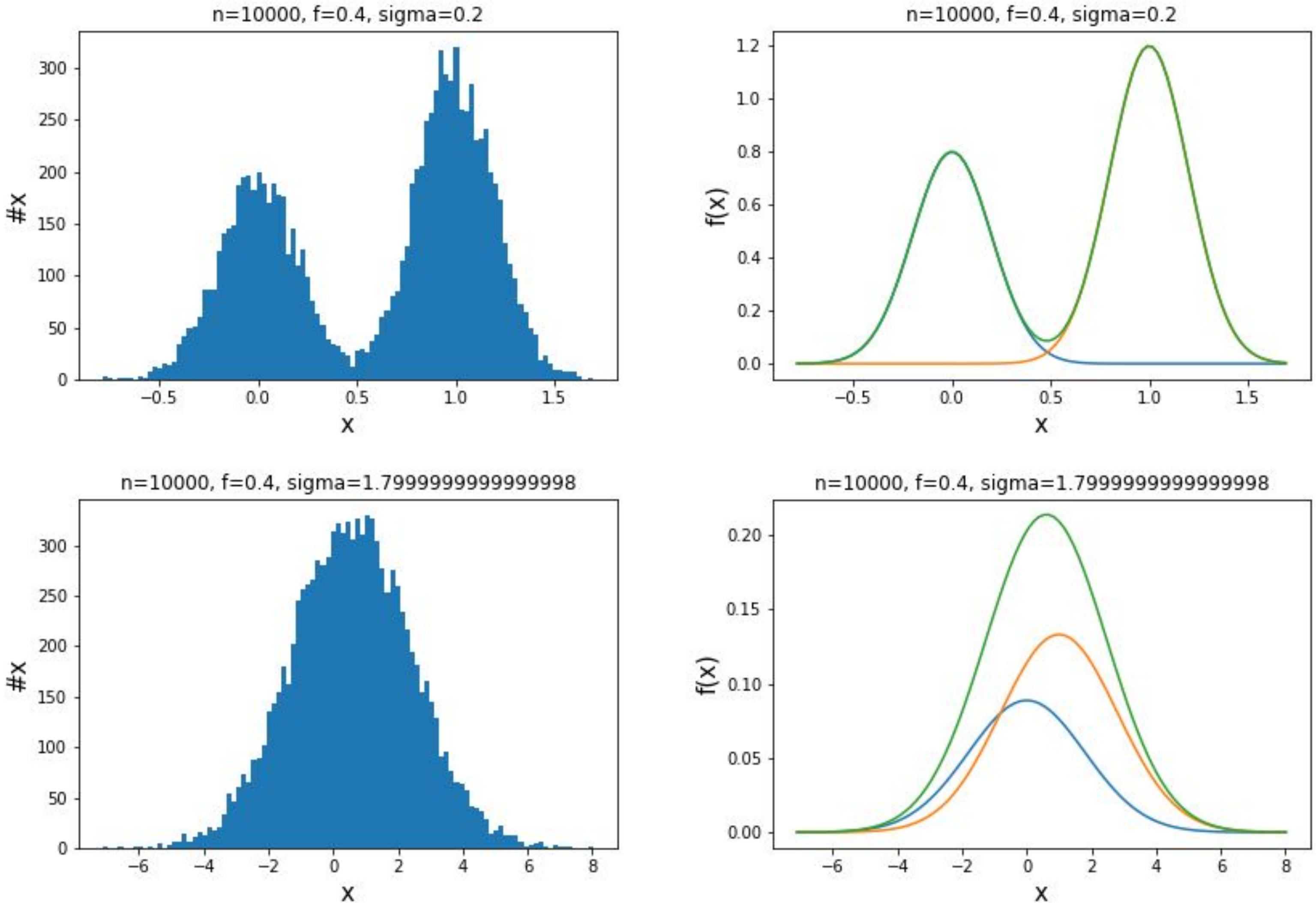


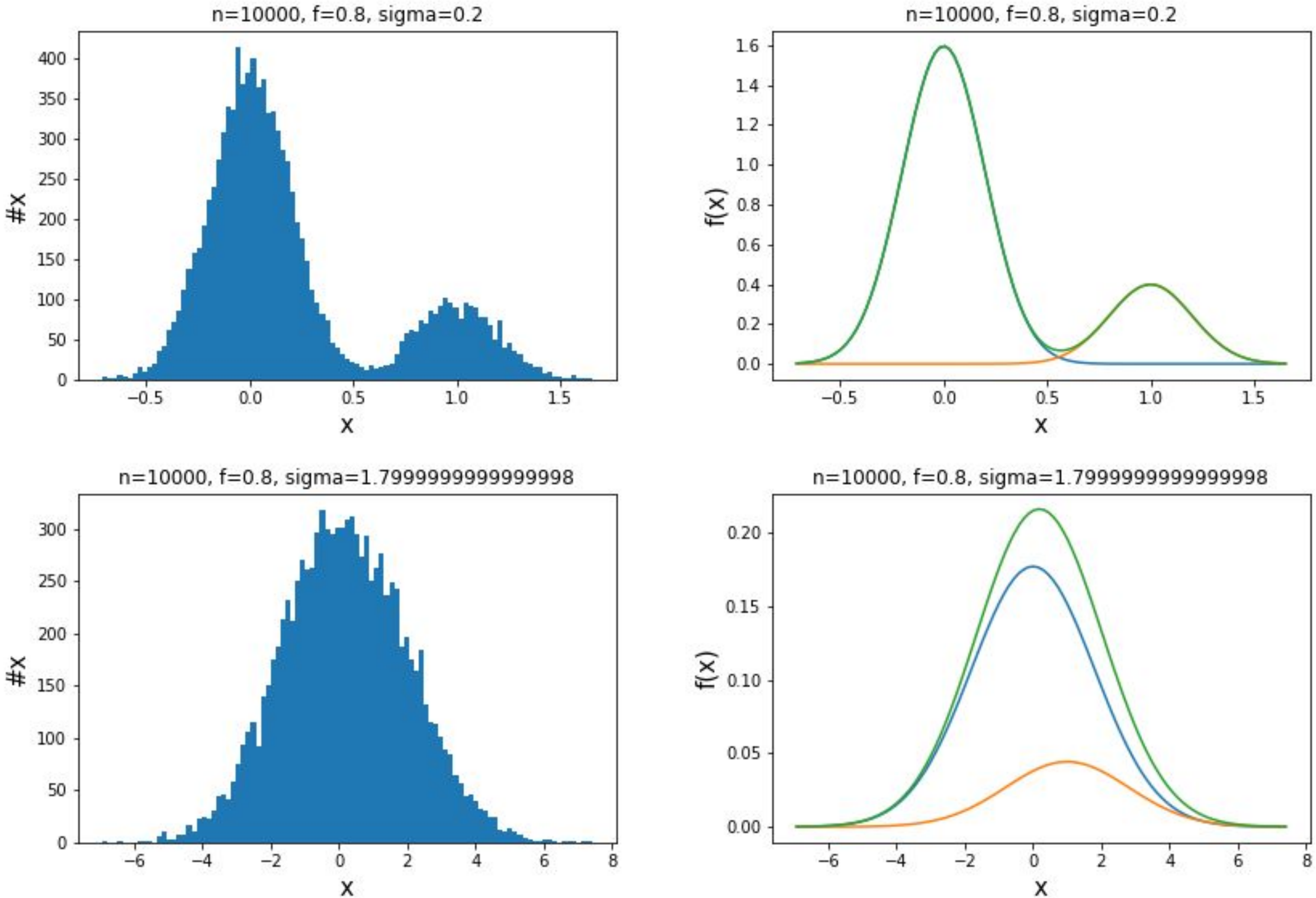
n=10000, f=0.5, sigma=1
#x
x

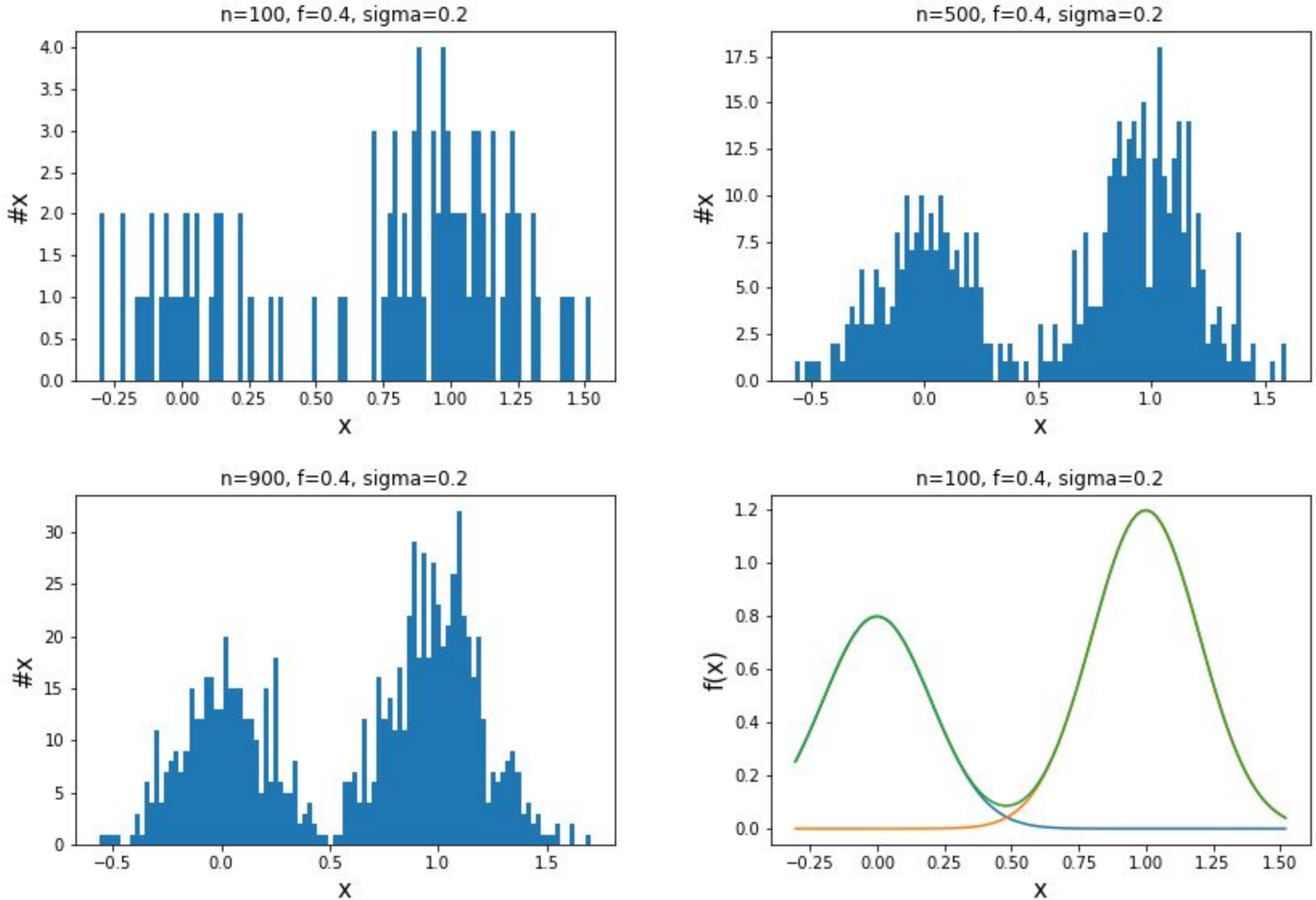


n=10000, f=0.5, sigma=1
f(x)
x


n=10000, f=0.2, sigma=0.2
#x
400
350
300
250
200
150
100
50
0
−0.5
0.0
0.5
1.0
1.5
n=10000, f=0.2, sigma=0.2
f(x)
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.5
0.0
0.5
1.0
1.5
n=10000, f=0.2, sigma=1.7999999999999998
#x
300
250
200
150
100
50
0
−6
−4
−2
0
2
4
6
8
x
n=10000, f=0.2, sigma=1.7999999999999998
f(x)
0.20
0.15
0.10
0.05
0.00
−6
−4
−2
0
2
4
6
8
x


n=10000, f=0.4, sigma=0.2
#x
x
n=10000, f=0.4, sigma=0.2
f(x)
x
n=10000, f=0.4, sigma=1.7999999999999998
#x
x
n=10000, f=0.4, sigma=1.7999999999999998
f(x)
x


n=10000, f=0.8, sigma=0.2
#x
x
n=10000, f=0.8, sigma=0.2
f(x)
x
n=10000, f=0.8, sigma=1.7999999999999998
#x
x
n=10000, f=0.8, sigma=1.7999999999999998
f(x)
x


n=100, f=0.4, sigma=0.2
n=500, f=0.4, sigma=0.2
n=900, f=0.4, sigma=0.2
n=100, f=0.4, sigma=0.2
#x
f(x)
x

Segunda questão

- Distribuição conjunta de X, Y e Z
- Inviabilidade de plotar um gráfico x * y * z * f(x, y, z)
- Plotar C = X + Y + Z

Código amostral

```
for i in range(n):
  mu = np.random.binomial(1, 1-f)
  x = np.random.normal(mu, scale=sigma)
  y = np.random.normal(mu, scale=sigma)
  z = np.random.normal(mu, scale=sigma)
  p1.append(x)
  p12.append(x+y)
  p123.append(x+y+z)

k = np.linspace(min(np.min(p1), np.min(p12), np.min(p123)), max(np.max(p1), np.max(p12), np.max(p123)), 100)
plt.hist(p1, bins=k, label='x')
plt.hist(p12, bins=k, label='x+y')
plt.hist(p123, bins=k, label='x+y+z')
```

Código curva teórica

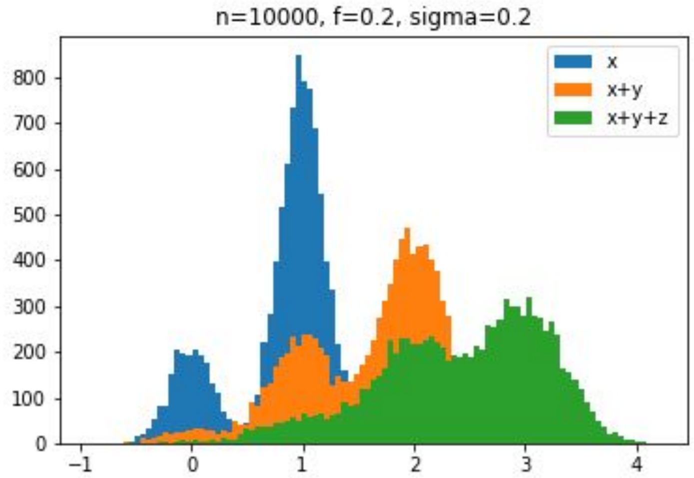
- A: soma das respostas dos eleitores de A
- B: soma das respostas dos eleitores de B
- C = A + B

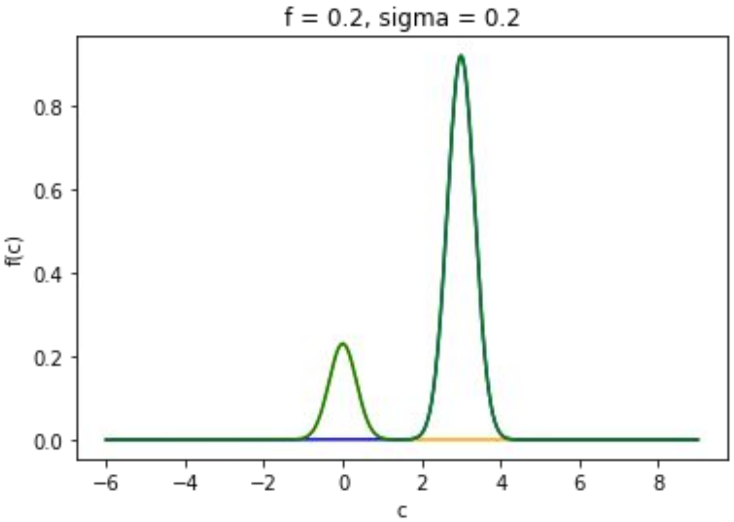
$$B = X_B + Y_B + Z_B = (1 - f) \cdot \mathcal{N}(3, 3\sigma^2)$$
$$A = X_A + Y_A + Z_A = f \cdot \mathcal{N}(0, 3\sigma^2)$$

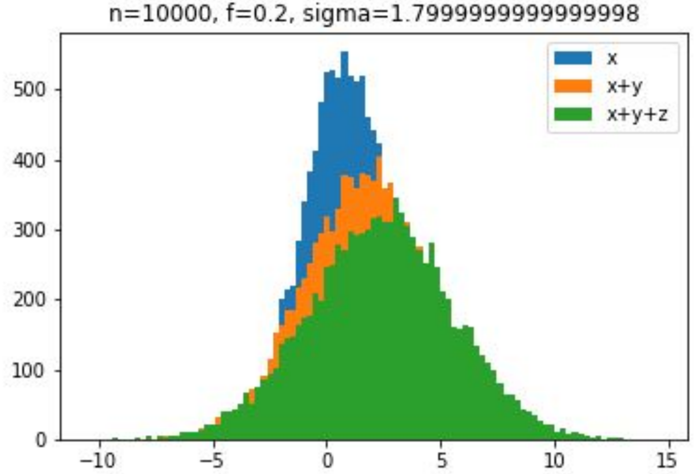
Código curva teórica

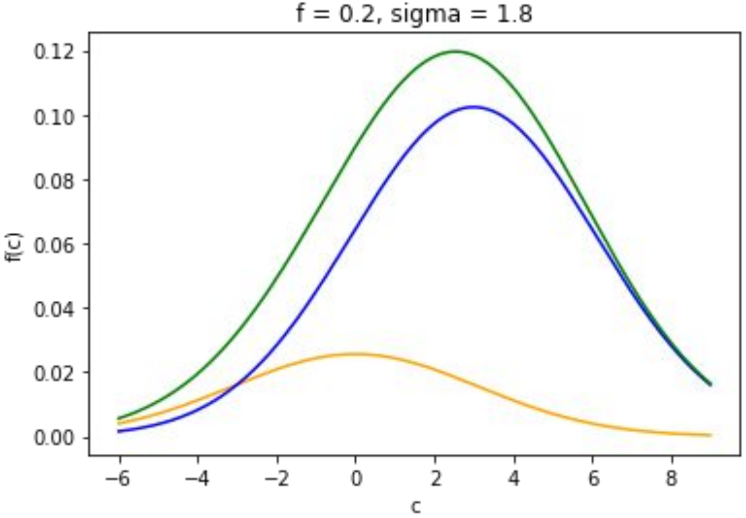
```
n = 2000
for f in [0.2, 0.4, 0.8]:
 for sigma in [0.2, 1.8, 3.4, 5]:
   xf = np.linspace(-6, 9, n)
   Fa = f*scipy.stats.norm.pdf(xf, 0, scale=np.sqrt(3)*sigma)
   Fb = (1-f)*scipy.stats.norm.pdf(xf, 3, scale=np.sqrt(3)*sigma)
   Fs = [x + y for x, y in zip(Fa, Fb)]

   plt.plot(xf, Fa, color = 'orange')
   plt.plot(xf, Fb, color = 'blue')
   plt.plot(xf, Fs, color = 'green')
```

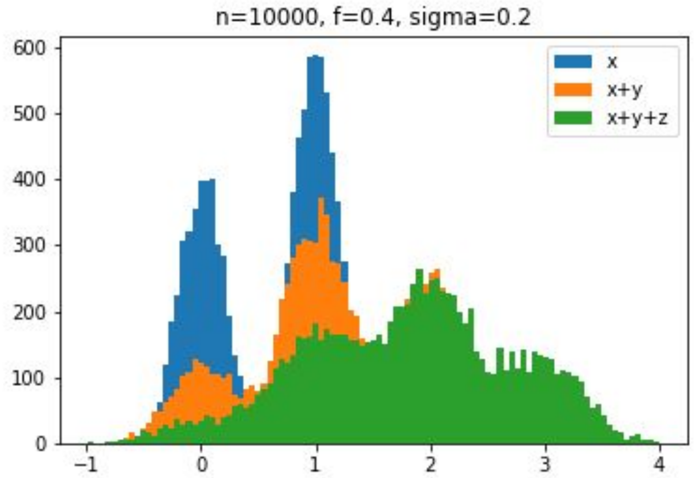
É propriedade das distribuições normais que a soma de duas normais é igual a uma normal com média igual à soma das duas que foram somadas para formá-la, e variância igual à soma das variâncias (e por consequência, desvio padrão igual à raiz quadrada da soma dos quadrados dos desvios padrões). Aplicando essas propriedades na distribuição de Xa, Ya e Za, e Xb, Yb e Zb, é possível determinar as somas de (X, Y, Z) para as duas metades da população, e somando essas duas distribuições, é possível encontrar a distribuição de probabilidade dessa soma. Os elementos *da soma* menores ou iguais a 1.5 compõem Fest, e podem ser comparados com f para encontrar o quão precisa a análise de fest é.

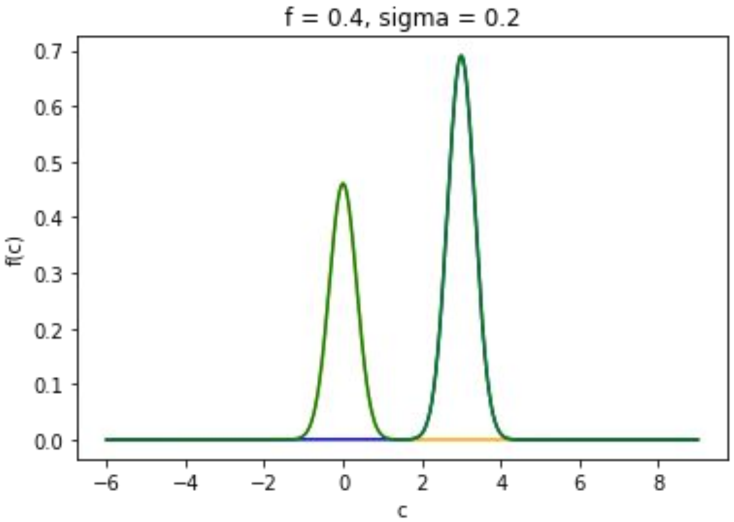

n=10000, f=0.2, sigma=0.2
x
x+y
x+y+z

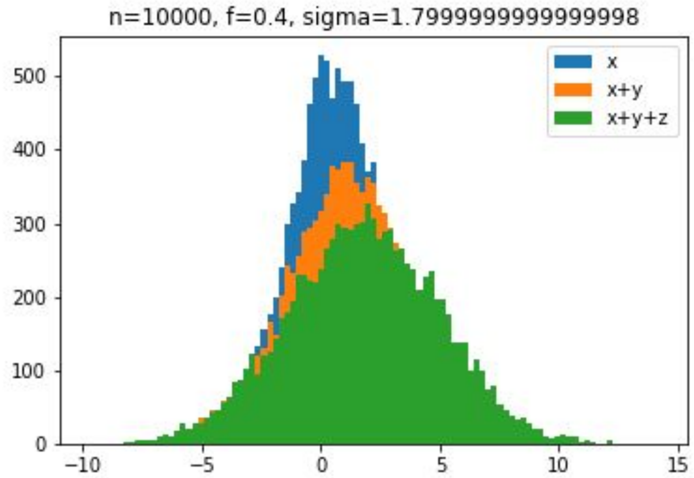

f = 0.2, sigma = 0.2
f(c)
c

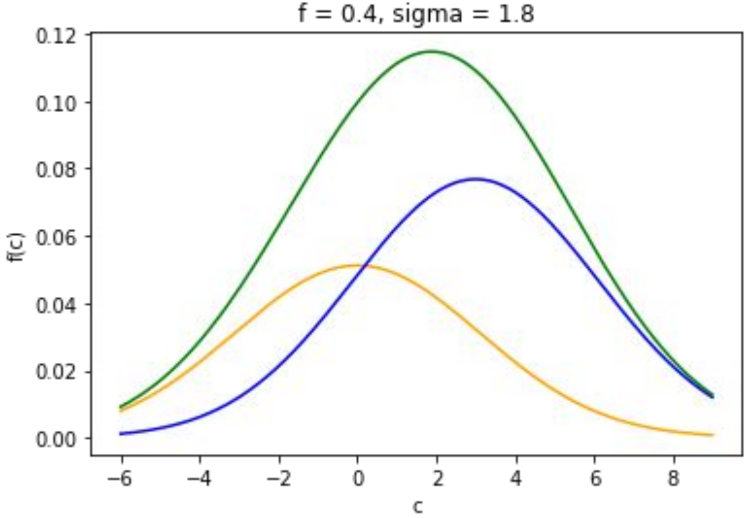

n=10000, f=0.2, sigma=1.7999999999999998
x
x+y
x+y+z

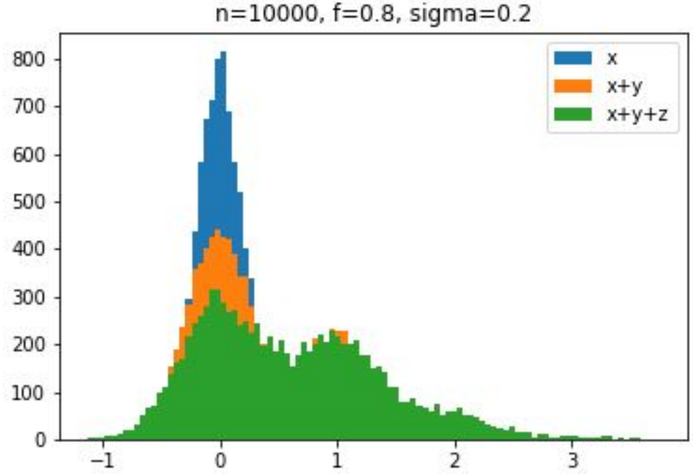

f = 0.2, sigma = 1.8
f(c)
c

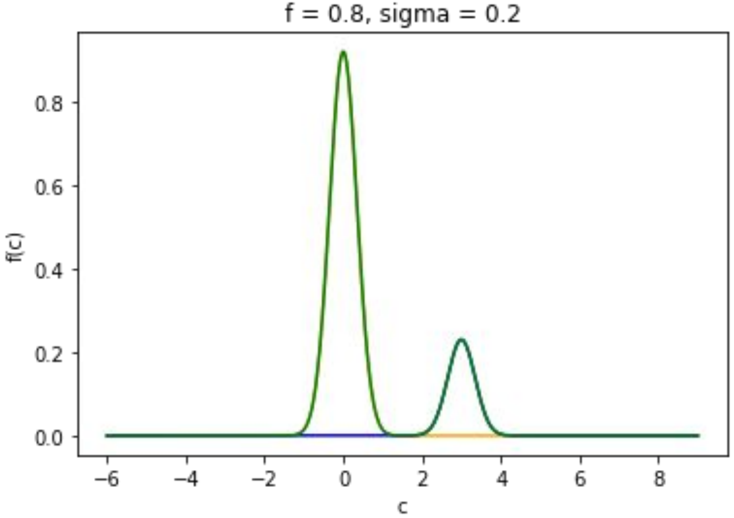

n=10000, f=0.4, sigma=0.2
x
x+y
x+y+z

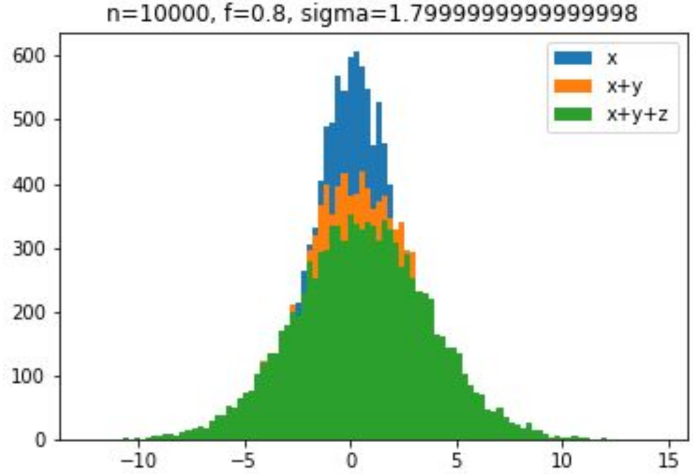

f = 0.4, sigma = 0.2
f(c)
c

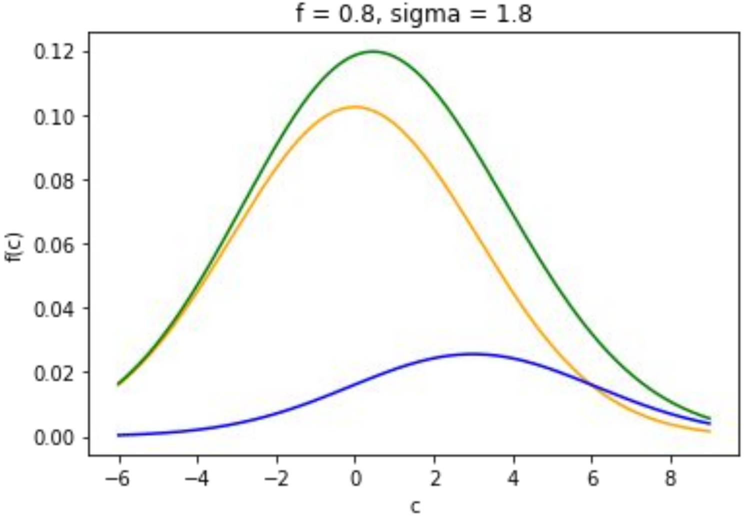

n=10000, f=0.4, sigma=1.7999999999999998
x
x+y
x+y+z


f = 0.4, sigma = 1.8
f(c)
c


n=10000, f=0.8, sigma=0.2
x
x+y
x+y+z
800
700
600
500
400
300
200
100
0
−1
0
1
2
3

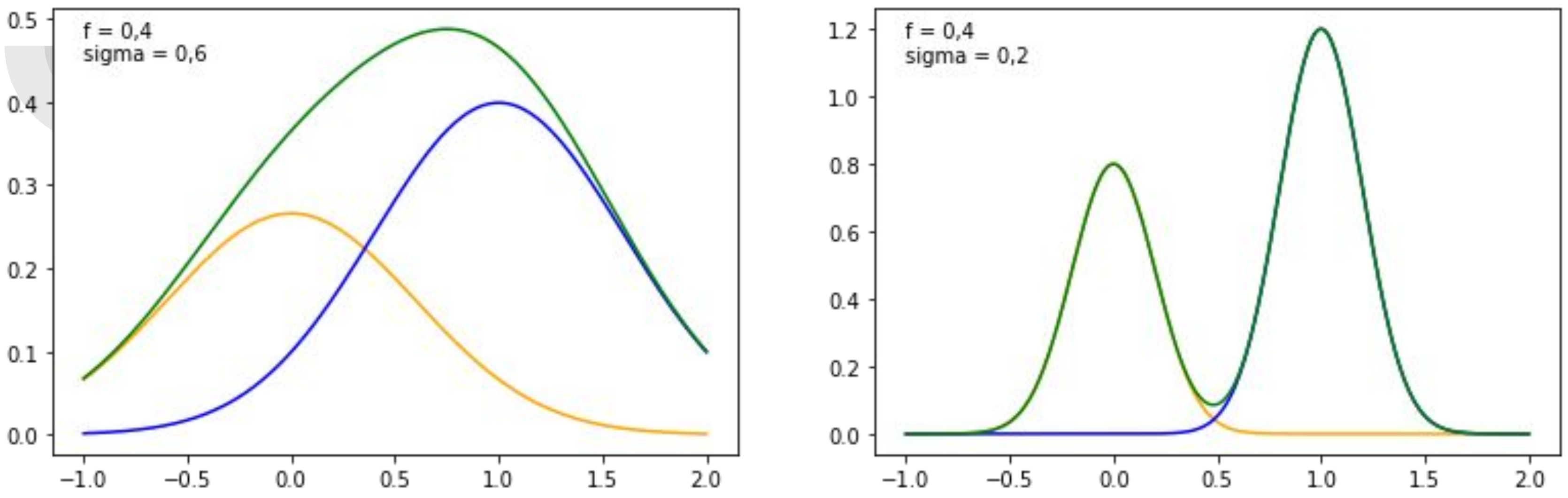

f = 0.8, sigma = 0.2
f(c)
0.8
0.6
0.4
0.2
0.0
−6
−4
−2
0
2
4
6
8
c


n=10000, f=0.8, sigma=1.7999999999999998
x
x+y
x+y+z
600
500
400
300
200
100
0
−10
−5
0
5
10
15


f = 0.8, sigma = 1.8
f(c)
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−6
−4
−2
0
2
4
6
8
c

Terceira questão

- A estimativa de candidato no qual um dado entrevistado irá votar com base nas suas respostas depende do valor de f e significativamente do valor de sigma
- Quanto maior o valor de sigma e mais próximo de 0.5 o de f, maior o overlap entre os dois grupos de eleitores

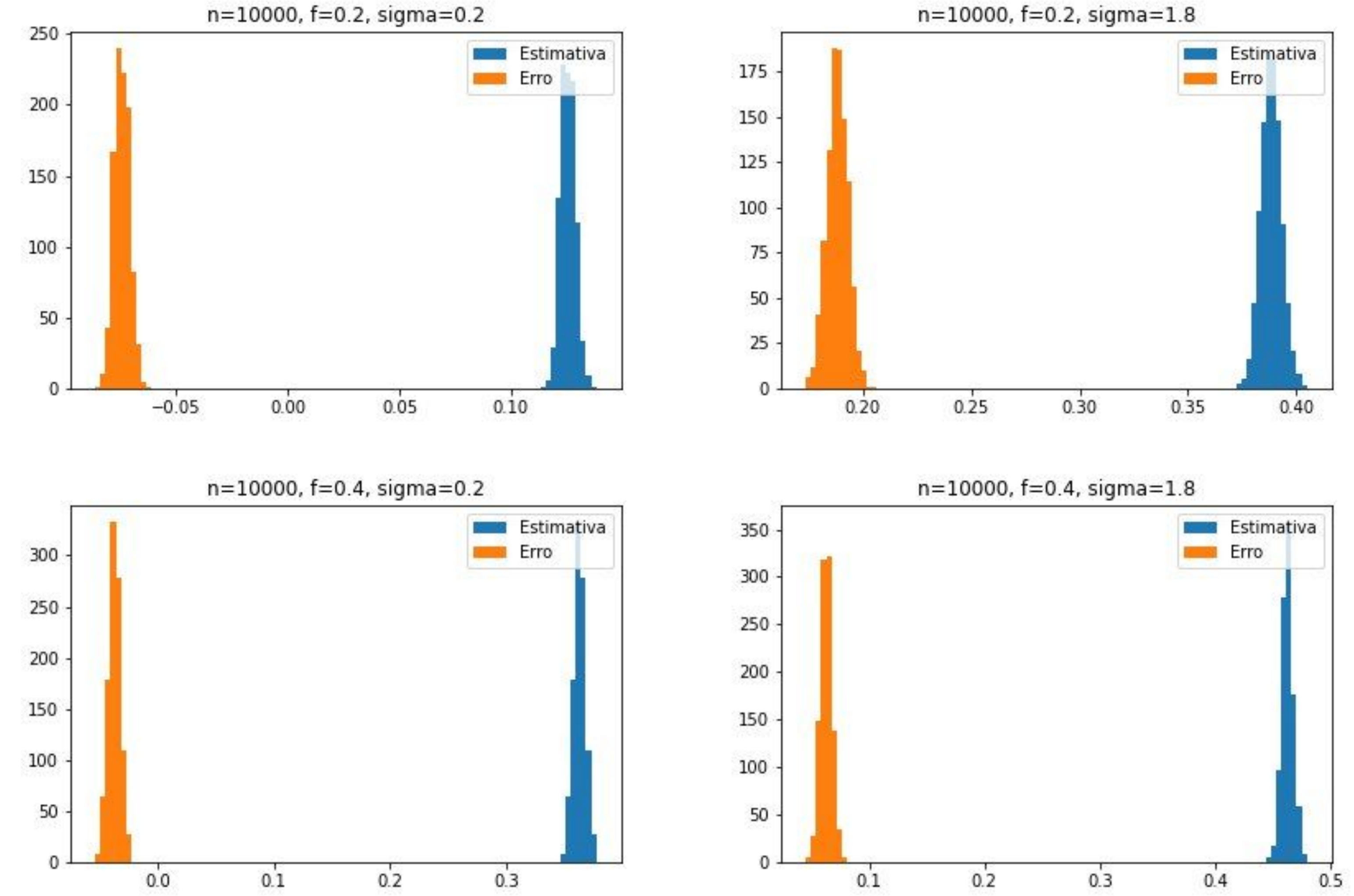

f = 0,4
sigma = 0,6
f = 0,4
sigma = 0,2

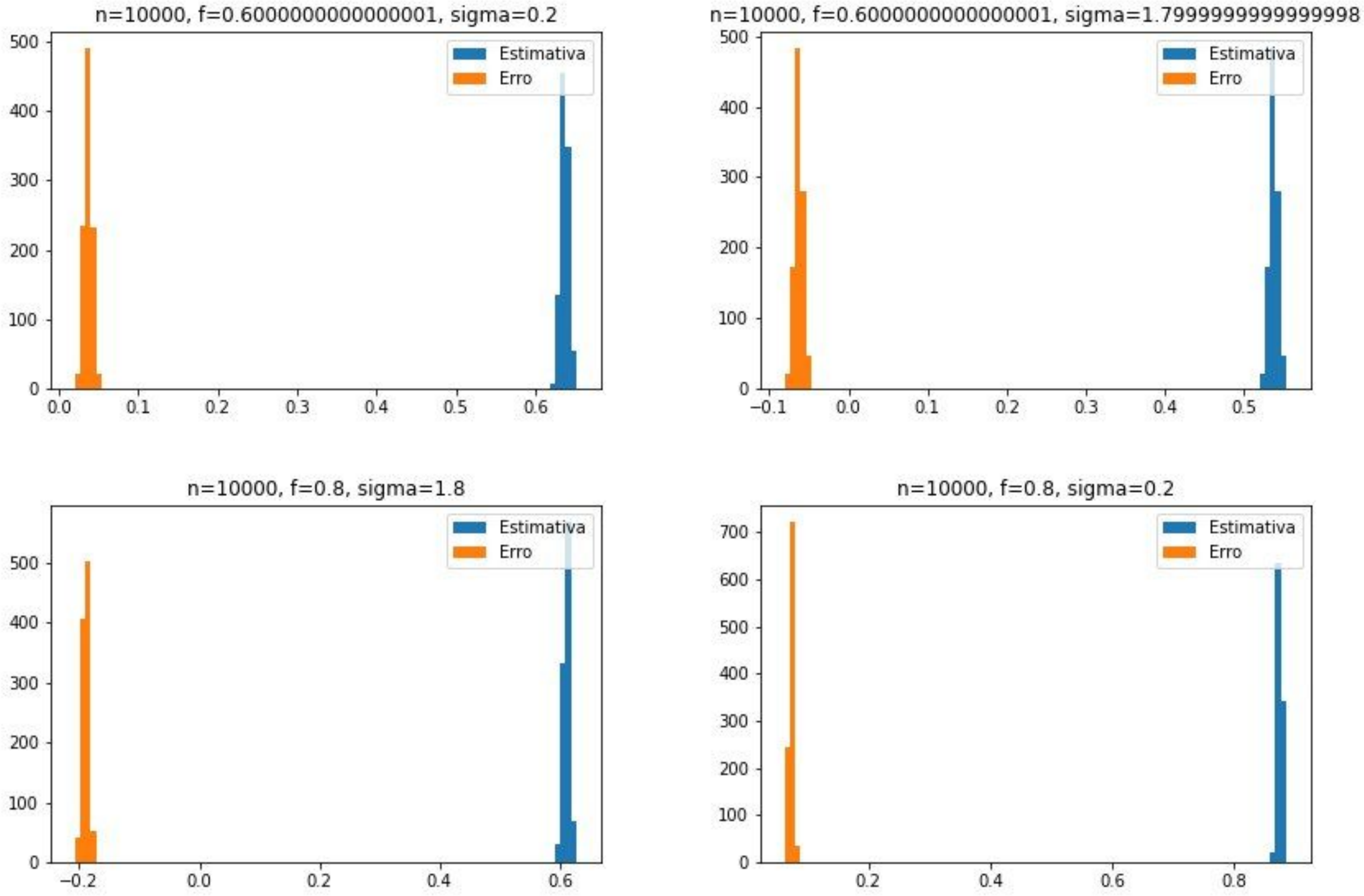
Quarta questão

- f estimado = P(C > 1.5)

Código

```
estimativas = []
erros = []
for i in range(repeticoes):
  est = 0
  for i in range(n):
    mu = np.random.binomial(1, 1-f)
    x = np.random.normal(mu, scale=sigma)
    y = np.random.normal(mu, scale=sigma)
    z = np.random.normal(mu, scale=sigma)
    if(x+y+z<1.5):
      est = est + 1
  estimativas.append(est/n)
  erros.append((est/n)-f)
k = np.linspace(min(np.min(estimativas), np.min(erros)), max(np.max(estimativas), np.max(erros)), 100)
plt.hist(estimativas, bins=k, label='Estimativa')
plt.hist(erros, bins=k, label='Erro')
```


n=10000, f=0.2, sigma=0.2
Estimativa
Erro
n=10000, f=0.2, sigma=1.8
Estimativa
Erro
n=10000, f=0.4, sigma=0.2
Estimativa
Erro
n=10000, f=0.4, sigma=1.8
Estimativa
Erro



n=10000, f=0.6000000000000001, sigma=0.2
Estimativa
Erro
n=10000, f=0.6000000000000001, sigma=1.7999999999999998
Estimativa
Erro
n=10000, f=0.8, sigma=1.8
Estimativa
Erro
n=10000, f=0.8, sigma=0.2
Estimativa
Erro