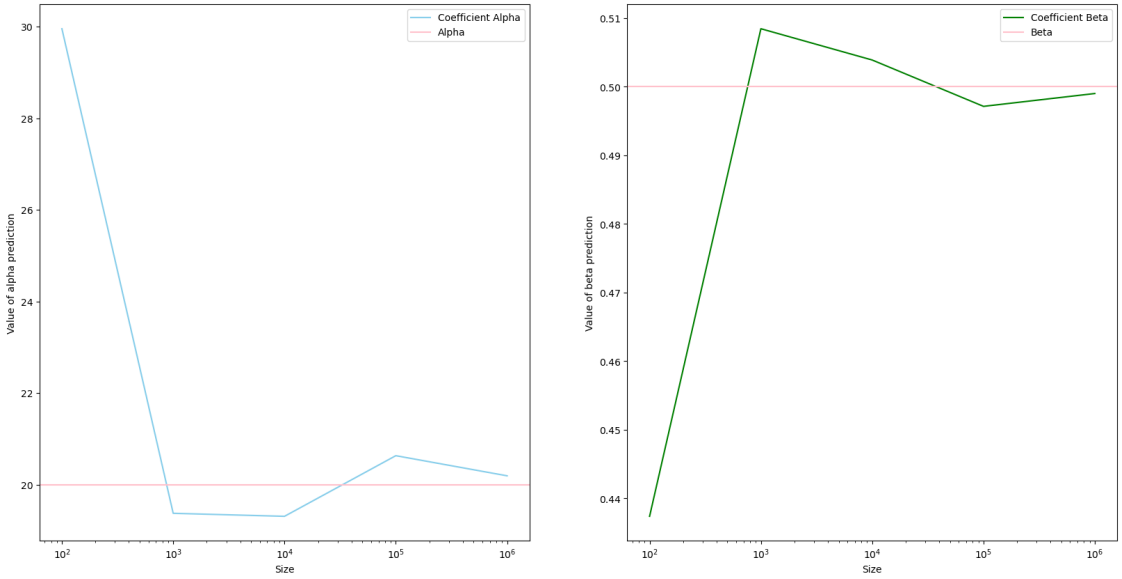
bonus_question

September 17, 2023

[ ]:
```
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
alpha = 20
beta = 0.5
sizes = [10**2, 10**3, 10**4, 10**5, 10**6]
alpha_pred =[]
beta_pred = []


for n in sizes:
    list_x = []
    list_y = []
    for a in range(n):
        x = np.random.normal(168,30)
        epsilon = np.random.normal(0,20)
        y = alpha + beta * x + epsilon
        list_x.append(x)
        list_y.append(y)

    x_array = np.array(list_x).reshape(-1,1)
    y_array = np.array(list_y).reshape(-1,1)
    model = LinearRegression()
    model.fit(x_array, y_array)
    coef = model.coef_
    intercept = model.intercept_
    alpha_pred.append(intercept)
    beta_pred.append(coef[0][0])



plt.figure(figsize=(20,10))
plt.subplot(121)
plt.plot(sizes,alpha_pred,color = 'skyblue', label= 'Coefficient Alpha')
plt.axhline(y=alpha, color = 'pink', label = 'Alpha')
plt.xscale('log')
plt.xlabel("Size")
```

```
plt.ylabel("Value of alpha prediction")
plt.legend()
plt.subplot(122)
plt.plot(sizes,beta_pred,color = 'green', label= 'Coefficient Beta')
plt.axhline(y=beta, color = 'pink', label = 'Beta')
plt.xscale('log')
plt.xlabel("Size")
plt.ylabel("Value of beta prediction")
plt.legend()
```

[ ]: <matplotlib.legend.Legend at 0x15022ca90>



Based in the pic, we can see that the coefficients will converges to $ a $ and $ b $ in the data generating process.