Data Science Analysis - Assignment 3

Varenya Upadhyaya

January 30, 2023

1. The following code generates the bootstrap samples from a standard normal distributions and plots the histogram along with the gaussian fit:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
from astroML.resample import bootstrap
from astroML.stats import median_sigmaG

N = 1000
N_boot = 10000
np.random.seed(10)
dist = stats.norm(0, 1).rvs(N)
sample_boot, sigmaG = bootstrap(dist, N_boot, median_sigmaG, kwargs = dict(axis=1))
x= np.linspace(-1,1,1000)
sigma = np.sqrt(np.pi/(2*N))
pdf = stats.norm(np.mean(sample_boot), sigma).pdf(x)

#plotting the histogram/distribution
plt.figure(figsize=(7,5))
plt.hist(sample_boot, bins=20, density=True, label='Bootstrap samples histogram',color
    ='#48b5c4')
plt.plot(x, pdf,  label='Gaussian fit',color='#115f9a' )

#minor
plt.xlim(-0.3,0.3)
plt.title('Bootstrap Samples')
plt.legend()
plt.savefig('1.png')
```

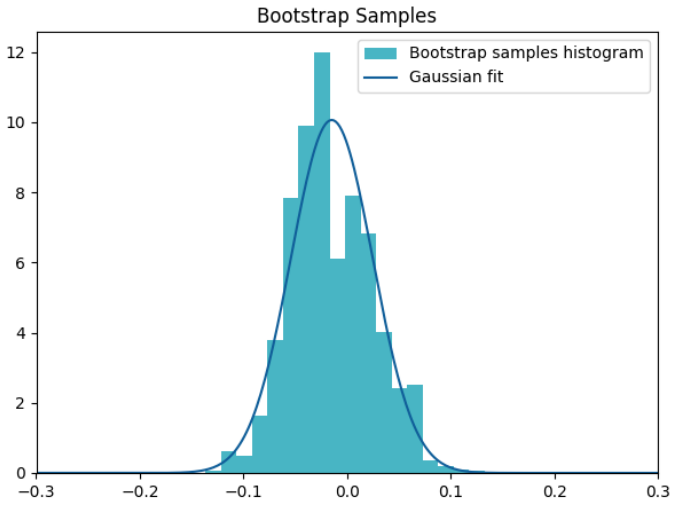

Figure 1: Data with the best fit

2. The following code plots Fig. (2) containing the data and the best fit line computed using `scipy.optimize.curve_fit`

```
from scipy.optimize import curve_fit
import numpy as np
import matplotlib.pyplot as plt

def line(x, m, c):
    return c+m*x

ID,x,y,sigma_y,sigma_x,rho_xy = np.loadtxt('data.txt', unpack=True)

#use curve_fit to perform chi^2 minimization
param, param_cov = curve_fit(line, x, y, sigma= sigma_y,absolute_sigma=True)
perr = np.sqrt(np.diag(param_cov))

#printing outputs
m = str(round(param[0],2))
c = str(round(param[1],1))
err_m = str(round(perr[0],2))
err_c = str(round(perr[1],1))
print("m = {}, err_m = {}".format(m,err_m))
print('c = {}, err_c = {} '.format(c,err_c))

#plotting the data/line
x_axis = np.linspace(0,300)
plt.figure(figsize=(9,7))
plt.text(125,150,"$y=({}\pm{})x+({}\pm{})$".format(m,err_m,c,err_c),fontsize = 16,
    fontweight='bold',color='black')
plt.errorbar(x,y,sigma_y, fmt='h', ms=6, color='#115f9a', mfc='#115f9a', mew=1, ecolor
    ='#115f9a', alpha=0.75, capsize=2.0, zorder=0, label='Data');
plt.plot(x_axis, param[0]*x_axis+param[1], '-', color='#a6d75b',label ='Best-fit line')

#minor
plt.title('Data vs. Best Fit Line',fontweight='bold',fontsize=16)
plt.xlim(0,300)
plt.ylim(0,700)
plt.legend(fontsize=16)
plt.savefig('2.png')
```

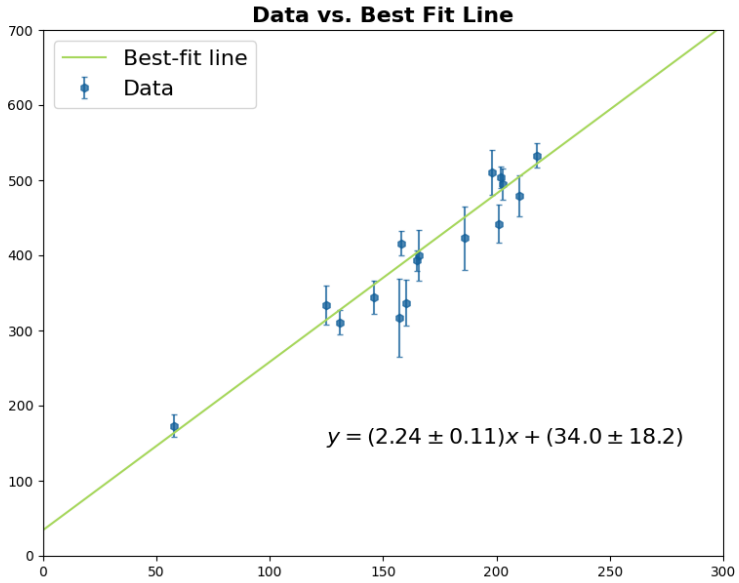

Figure 2: Data with the best fit

Code output:

```
m = 2.24, err_m = 0.11
c = 34.0, err_c = 18.2
```

3. The p-values of the four χ^2 are calculated using the following code:

```
import numpy as np
from scipy import stats

N = 50 #from the source code
chi2 = np.array([0.96,0.24,3.84,2.85])*(N-1)
p =  1 - stats.chi2.cdf(chi2,N-1)
print('p-values =',p)
```

Code output:

```
p-values = [5.52926434e-01 9.99999992e-01 0.00000000e+00 1.21072929e-10]
```

All the codes and figures used in this assignment can be found in this repository