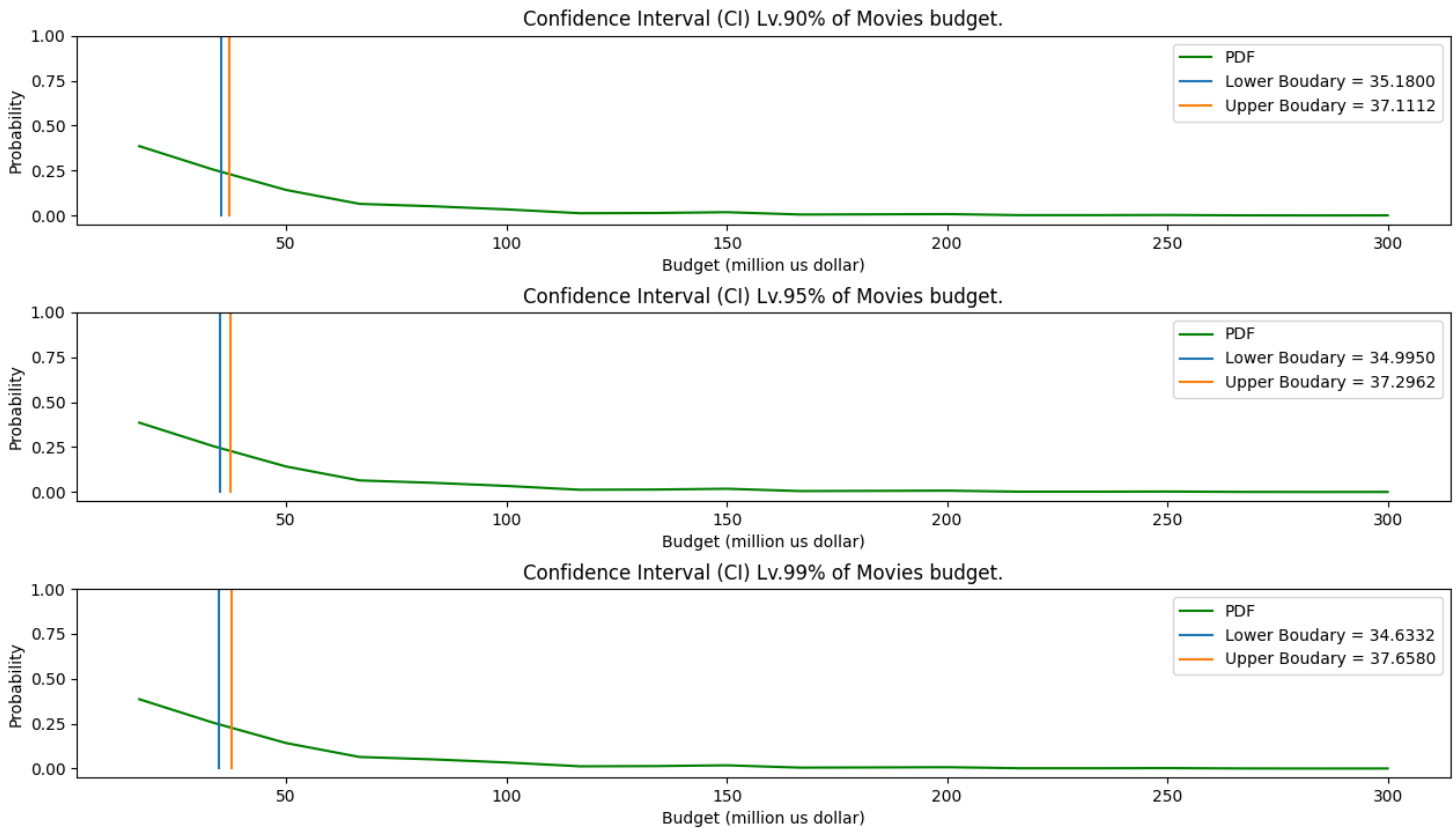


## Homework 4

## Confidence Interval



กราฟแสดงช่วงของความเชื่อมั่นในระดับต่างๆ เทียบกับกราฟ PDF

ช่วงความเชื่อมั่น (confidence interval) หมายถึง ช่วงของค่าประมาณที่ประกอบไปด้วยค่าต่ำสุด และค่าสูงสุด ที่คำนวณขึ้นมาจากสูตรข้างต้น ที่บ่งบอกค่าเฉลี่ยที่บอกระดับความเชื่อมั่น โดยอิงจากกลุ่มตัวอย่าง (Sample) ที่สามารถใช้อ้างอิงถึงข้อมูลทั้งหมดได้ (Population)

$$Z_{\alpha/2} \times \frac{\sigma}{\sqrt{n}}$$

standard deviation

confidence level

confidence coefficient

sample size

สูตรการคำนวณหา Confidence Interval (CI)

```

sample size = 4638
---- Value of 90%----
Mean : 36.1456
Lower - Upper boudary : 35.1800 - 37.1112
---- Value of 95%----
Mean : 36.1456
Lower - Upper boudary : 34.9950 - 37.2962
---- Value of 99%----
Mean : 36.1456
Lower - Upper boudary : 34.6332 - 37.6580

```

ค่าที่ได้จากการคำนวณ

จากการคำนวณโดยใช้ช่วงระดับความเชื่อมั่น 3 ช่วง ได้แก่ 90% 95% และ 99% ได้ค่าดังนี้

1. เปอร์เซ็นต์ความเชื่อมั่น 90% จะมีต้นทุนอยู่ที่ 35.18 – 37.11 ล้านบาท
2. เปอร์เซ็นต์ความเชื่อมั่น 95% จะมีต้นทุนอยู่ที่ 34.99 – 37.29 ล้านบาท
3. เปอร์เซ็นต์ความเชื่อมั่น 99% จะมีต้นทุนอยู่ที่ 34.63 – 37.65 ล้านบาท

จากการคำนวณช่วงความเชื่อมั่นที่ระดับความเชื่อมั่น ทั้ง 3 ระดับ ได้แก่ 90%, 95% และ 99% จะได้ค่าของความเชื่อมั่นตามข้อมูลข้างต้น จึงสรุปได้ว่า ในช่วงต้นทุนที่ 35.18 – 37.11 ล้านบาท จะครอบคลุมช่วงข้อมูลทั้งหมดที่ 90% ในช่วงต้นทุนที่ 34.99 – 37.29 ล้านบาท จะครอบคลุมช่วงข้อมูลทั้งหมดที่ 95% และถ้าใช้ข้อมูลในช่วง 34.63 – 37.65 ล้านบาท จะครอบคลุมข้อมูลทั้งหมดถึง 99%

สรุปได้ว่า หากมีการผลิตภาพยนตร์เพิ่มขึ้น แล้วมีการสุ่มกลุ่มตัวอย่าง (sample size) ใหม่อีกครั้ง แล้วนำมาคำนวณค่าความเชื่อมั่นอีกครั้ง จะพบว่าค่าเฉลี่ยที่ได้จะต้องยังคงอยู่ในช่วงความเชื่อมั่นที่คำนวณได้ข้างต้นตามระดับความเชื่อมั่นข้างต้น เช่น ถ้าใช้ช่วงความเชื่อมั่นที่ 95% โอกาสที่สุ่มใหม่แล้วได้ค่าอยู่ในช่วงเดิมก็จะอยู่ที่ 95% โดยถ้าใช้ช่วงข้อมูลที่ 34.63 – 37.65 ล้านบาท กลุ่มข้อมูลตัวอย่างใหม่ที่สุ่มมาต้องอยู่ในช่วงนี้อย่างแน่นอน เนื่องจากมีค่าความเชื่อมั่นถึง 99%

## Source code

```

import pandas
import matplotlib.pyplot as plt
import numpy as np

import scipy.stats

plt.style.use('bmh')
columns = pandas.read_csv('../Lab2/moviesfilter.csv')

# budget
x = columns['budget']
budget = x.to_list()

#format data to million dollar
for i in range(0, len(budget)):
    budget[i] = budget[i]/1000000

print('sample size =',len(budget))

def mean_confidence_interval(data, confidence=0.95):
    a = 1.0 * np.array(data)
    n = len(a)
    m, se = np.mean(a), scipy.stats.sem(a)
    h = se * scipy.stats.t.ppf((1 + confidence) / 2., n-1)
    print('---- Value of {:.0f}%----'.format(confidence*100))
    print('Mean :',m)
    print('Lower - Upper boudary {:.4f} - {:.4f}'.format(m-h,m+h))
    return m, m-h, m+h

m1, lB1, uB1 = mean_confidence_interval(budget,0.90)
m2, lB2, uB2 = mean_confidence_interval(budget,0.95)
m3, lB3, uB3 = mean_confidence_interval(budget,0.99)

count, bins_count = np.histogram(budget, bins=18)
pdf = count / sum(count)

figure, func = plt.subplots(3, 1, figsize=(8, 10))
plt.tight_layout(pad=5,h_pad=5.0)

y = np.linspace(0,1)
title1,title2 = 'Confidence Interval (CI) Lv.', '% of Movies budget.'
xlabel = "Budget (million us dollar)"
ylabel = "Probability"

```

```

func[0].set_title(title1+'90'+title2)
func[0].set_xlabel(xlabel)
func[0].set_ylabel(ylabel)
func[0].plot(bins_count[1:], pdf, color="green", label="PDF" )
x1,x2 = np.linspace(lB1,lB1),np.linspace(uB1,uB1)
func[0].plot(x1,y, label="Lower Boudary = {:.4f}".format(lB1))
func[0].plot(x2,y, label="Upper Boudary = {:.4f}".format(uB1))
func[0].legend()
func[0].axis(ymax=1)

func[1].set_title(title1+'95'+title2)
func[1].set_xlabel(xlabel)
func[1].set_ylabel(ylabel)
func[1].plot(bins_count[1:], pdf, color="green", label="PDF" )
x1,x2 = np.linspace(lB2,lB2),np.linspace(uB2,uB2)
func[1].plot(x1,y, label="Lower Boudary = {:.4f}".format(lB2))
func[1].plot(x2,y, label="Upper Boudary = {:.4f}".format(uB2))
func[1].legend()
func[1].axis(ymax=1)

func[2].set_title(title1+'99'+title2)
func[2].set_xlabel(xlabel)
func[2].set_ylabel(ylabel)
func[2].plot(bins_count[1:], pdf, color="green", label="PDF" )
x1,x2 = np.linspace(lB3,lB3),np.linspace(uB3,uB3)
func[2].plot(x1,y, label="Lower Boudary = {:.4f}".format(lB3))
func[2].plot(x2,y, label="Upper Boudary = {:.4f}".format(uB3))
func[2].legend()
func[2].axis(ymax=1)

plt.show()

```