

นายภากรณ์ ธนประชาพันธ์ 62010694

Homework #4

สถิติความล่าช้าของเที่ยวบินภายในประเทศสหรัฐอเมริกาเดือนสิงหาคม

2018 August 2018 Nationwide Airplane Delay Statistic

เลือกใช้ Column : ARR_DELAY (Arrival Delay (HHMM)) เพื่อหา Confidence Interval (CI) of Mean

Confidence Interval (CI) of Mean

90% z-critical value:

1.6448536269514722

ConfidenceLevel = 90% ---> Confidence interval:
(16.52751809125205, 17.738341891461864)

95% z-critical value:

1.959963984540054

ConfidenceLevel = 95% ---> Confidence interval:
(16.411537217814686, 17.854322764899226)

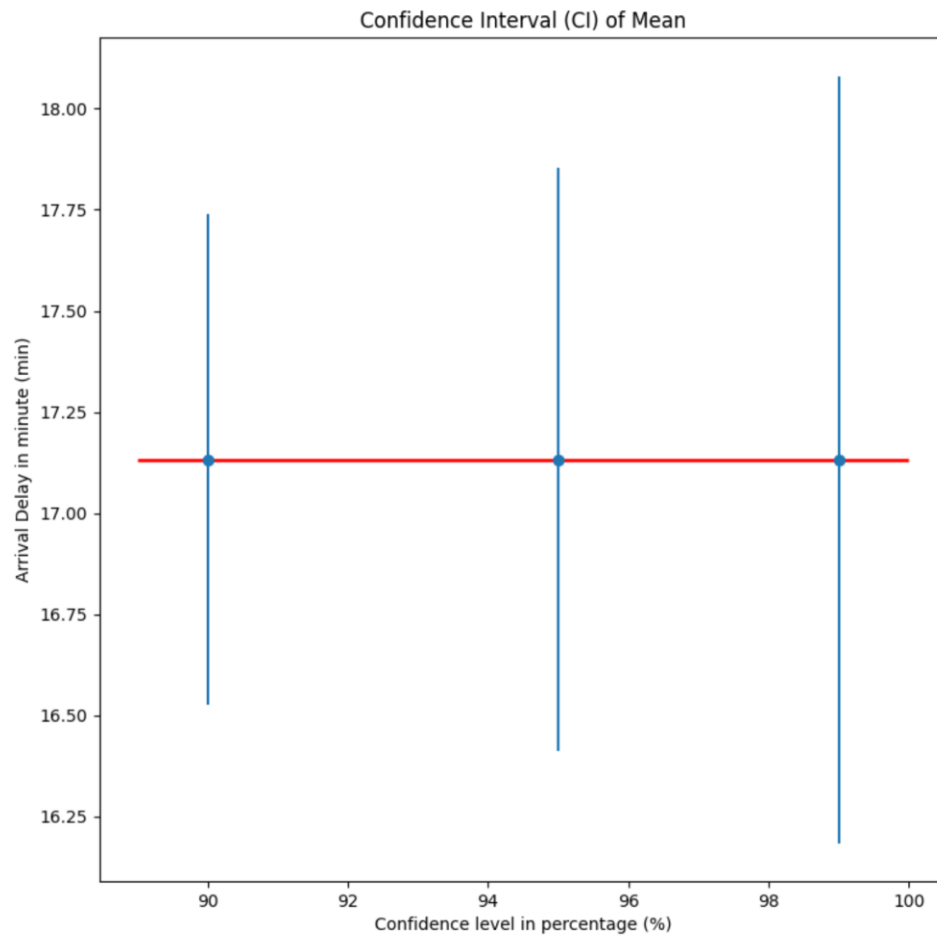
99% z-critical value:

2.5758293035489004

ConfidenceLevel = 99% ---> Confidence interval:
(16.184859179793214, 18.0810008029207)



Figure 1



บทวิเคราะห์ข้อมูลจากกราฟ : จากชุดข้อมูลที่ได้นำมาซึ่งคือ Column : ARR_DELAY จะสามารถเห็นได้ว่า ค่าเวลาล่าช้าขาเข้า (Arrival Delay (HHMM)) ช่วงความเชื่อมั่นหากใช้ Confidence Level เป็น 90%,95%,99% จะอยู่ในช่วง (16.52751809125205, 17.738341891461864) , (16.411537217814686, 17.854322764899226) , (16.184859179793214, 18.0810008029207) ตามลำดับ โดยทั้งหมดนี้อยู่ในหน่วยนาที และเส้นแดงที่ขนานในแกน x หมายถึง Sample Mean เราจึงสามารถค่อนข้างมั่นใจได้ว่า เทียบกับนั้นจะมีค่าเวลาล่าช้าขาเข้าอยู่ในช่วงแต่ละช่วง ตาม % Confidence Level ที่ต้องการนั่นคือ 90%,95%,99% และอยู่ที่ประมาณ Sample Mean ซึ่งคือ 17.132 นาทีอีกด้วย การที่เราหาช่วงความเชื่อมั่นนี้เพราะจะสามารถเพิ่มความมั่นใจว่าผลลัพธ์ที่เราประมาณออกมานั้นจะมีโอกาสใกล้เคียงกับความเป็นจริงมากขึ้น จากกราฟเราจะสามารถสังเกตได้อีกว่า หากเลือก Confidence Level เยอะ ช่วงความเชื่อมั่นจะมีค่าช่วงเยอะมากขึ้นอีกตามกันไปแบบแปรผันตรง

Source Code

```
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import random
import math
import Converter

data = pd.read_csv("August_2018_Nationwide_Airplane_Delay_Statistic.csv")

x = data["ARR_DELAY (HHMM)"]
mean = Converter.convertToMin(str(x.mean()))
size = len(x)

intervals = []
sample_means = [mean, mean, mean]

z_critical_90 = stats.norm.ppf(q = 0.95) #90
z_critical_95 = stats.norm.ppf(q = 0.975) #95
z_critical_99 = stats.norm.ppf(q = 0.995) #99

pop_stdev = Converter.convertToMin(str(x.std()))

margin_of_error_90 = z_critical_90 * (pop_stdev/math.sqrt(size))
margin_of_error_95 = z_critical_95 * (pop_stdev/math.sqrt(size))
margin_of_error_99 = z_critical_99 * (pop_stdev/math.sqrt(size))

confidence_interval_90 = (mean - margin_of_error_90,
| | | | |
| | | | | mean + margin_of_error_90)
intervals.append(confidence_interval_90)
confidence_interval_95 = (mean - margin_of_error_95,
| | | | |
| | | | | mean + margin_of_error_95)
intervals.append(confidence_interval_95)
confidence_interval_99 = (mean - margin_of_error_99,
| | | | |
| | | | | mean + margin_of_error_99)
intervals.append(confidence_interval_99)

print("90% z-critical value:")
print(z_critical_90)
print("ConfidenceLevel = 90% ---> Confidence interval:")
print(confidence_interval_90)
print()
print("95% z-critical value:")
print(z_critical_95)
print("ConfidenceLevel = 95% ---> Confidence interval:")
print(confidence_interval_95)
print()
print("99% z-critical value:")
print(z_critical_99)
print("ConfidenceLevel = 99% ---> Confidence interval:")
print(confidence_interval_99)
```

```

print("90% z-critical value:")
print(z_critical_90)
print("ConfidenceLevel = 90% ---> Confidence interval:")
print(confidence_interval_90)
print()
print("95% z-critical value:")
print(z_critical_95)
print("ConfidenceLevel = 95% ---> Confidence interval:")
print(confidence_interval_95)
print()
print("99% z-critical value:")
print(z_critical_99)
print("ConfidenceLevel = 99% ---> Confidence interval:")
print(confidence_interval_99)

plt.figure(figsize=(9,9))
plt.errorbar(x=[90,95,99],
             y=sample_means,
             yerr=[(top-bot)/2 for top,bot in intervals],
             fmt='o',
             label=["ConfidenceLevel = 90%", "ConfidenceLevel = 95%", "ConfidenceLevel = 99%"])

plt.hlines(xmin=89, xmax=100,
           y=mean,
           linewidth=2.0,
           color="red");

plt.title("Confidence Interval (CI) of Mean")
plt.xlabel("Confidence level in percentage (%)")
plt.ylabel("Arrival Delay in minute (min)")
plt.show()

```