ព្រះរាជាណាចក្រកម្ពុជា

ជាតិ សាសនា ព្រះមហាក្សត្រ

វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា

ដេប៉ាតឺម៉ង់ថ្នាក់ឆ្នាំសិក្សាមូលដ្ឋាន

របាយការណ៍ពិសោធន៍ ប៉ោលទោល

ក្រុម **I1-03**

ឈ្មោះនិស្សិត	**លេខសម្គាល់ខ្លួននិស្សិត**	**ពន្ទុ**
1. ជា ហាក់យ័របី	e20240472	....
2. ជា ហេងលី	e20240386	....
3. ជា កាប់ព្រីស៊ីយោ	e20241096	....
4. ជា លីម៉េង	e20240522	....
5. ជា លីហាក់	e20241231	....

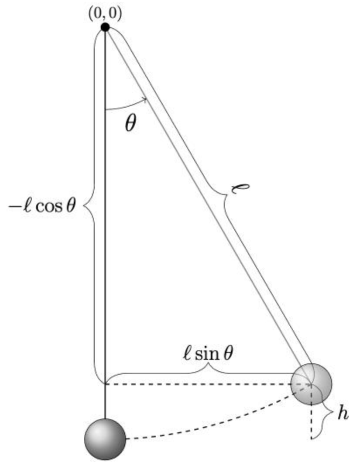
ឈ្មោះគ្រូ ៖ ស៊ុន សុផក្តិ

ឆ្នាំសិក្សា ២០២៤ - ២០២៥

មតិការ

I. សេចក្តីផ្តើម

នៅក្នុងការពិសោធន៍នេះ យើងកំណត់តម្លៃសំទុះទំនាញផែនដី g ដោយពិសោធន៍។ យើងប្រើសមីការដូចខាងក្រោម៖



$$T = 2\pi\sqrt{\frac{L}{g}}$$

ដោយ

- T គឺជាខួបរំយោល
- L គឺជារង្វាស់ខ្សែនៃរំយោល
- g គឺជាសំទុះទំនាញផែនដៃ

គោលបំណងនៃពិសោធន៍

ដើម្បីកំណត់ការបង្កើនល្បឿនដោយសារទំនាញ (g) ដោយវាស់ខួបនៃរំយោលនិង វិភាគ T^2 L។

សម្ភារៈប្រើប្រាស់

- ប៉ោលមួយដែលចងជាប់ទៅនឹងខ្សែ
- ខ្សែរង្វាស់ (ឬវត្ថុវាស់ប្រវែងសផ្សេងៗទៀត)
- នាឡិកា (ឬវត្ថុកំណត់ពេលវេលាផ្សេងៗទៀត)
- ប្រូត្រាក់ទ័រ (ឬវត្ថុរង្វាស់មុំផ្សេងៗទៀត)

II. ការវិភាគទិន្នន័យ

1. ការកែច្នៃទិន្នន័យ

L_1 (m)	t_10	t_10	t_10	t_10	t_10	t_avg	t_0	t_0^2
0.6	15.03	15.43	15.45	15.23	15.56	15.34	1.534	2.353156
0.7	15.65	15.89	15.7	16.57	16.91	16.144	1.6144	2.60628736
0.8	16.889	17.99	17.7	17.15	17.77	17.4998	1.74998	3.06243
0.9	18.99	19.01	18.78	18.86	19.11	18.95	1.895	3.591025
1	19.54	20.11	19.43	19.67	19.99	19.748	1.9748	3.89983504
1.1	21.03	21.13	20.93	20.85	20.78	20.944	2.0944	4.38651136
1.2	22.01	22	22.16	22.09	22.08	22.068	2.2068	4.86996624

- ការដាក់ទិន្នន័យចូលក្នុង File ដើម្បីបំលែង

```python
import pandas as pd

# Importing CSV under data
data = pd.read_csv("./pendulum.csv")
```

[27] ✓ 1.9s Python

- ទាញទិន្នន័យ

```python
# Table 1.0 : Data Table
table_1 = data.iloc[:7]

print(table_1)
```

[28] ✓ 0.7s Open 'table_1' in Data Wrangler Python

```
...  L_1 (m)    t_10  t_10.1  t_10.2  t_10.3  t_10.4    t_avg      t_0
0      0.6   15.03   15.43   15.45   15.23   15.56  15.3400  1.53400  2
1      0.7   15.65   15.89   15.70   16.57   16.91  16.1440  1.61440  2
2      0.8  16.889   17.99   17.70   17.15   17.77  17.4998  1.74998  3
3      0.9   18.99   19.01   18.78   18.86   19.11  18.9500  1.89500  3
4        1   19.54   20.11   19.43   19.67   19.99  19.7480  1.97480  3
5      1.1   21.03   21.13   20.93   20.85   20.78  20.9440  2.09440  4
6      1.2   22.01   22.00   22.16   22.09   22.08  22.0680  2.20680  4
```

- ស្រង់ទិន្នន័យដើម្បីសង់ក្រាប

```
# Table 2.0 : String length vs. Periodicity Squared
table_2 = data[["L_1 (m)", "t_0^2"]][:7]

print(table_2)
```

[29] ✓ 0.4s Open 'table_2' in Data Wrangler Python

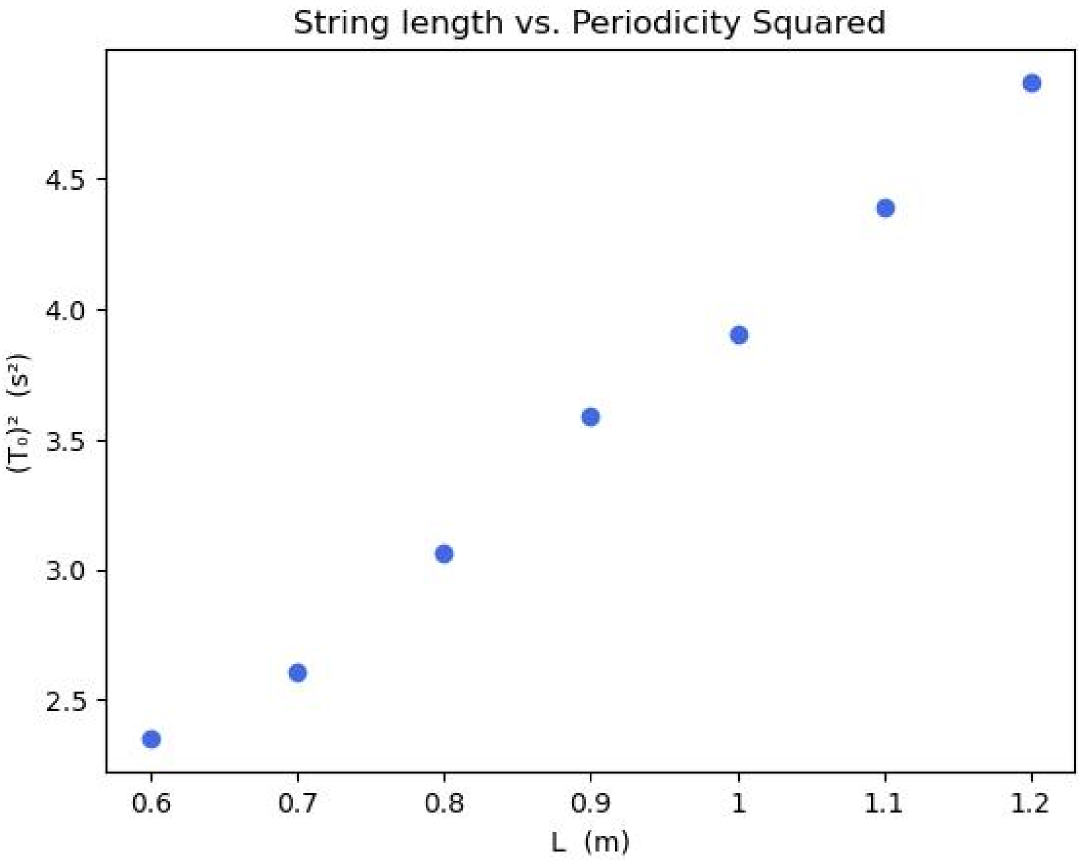
```
...   L_1 (m)     t_0^2
   0      0.6  2.353156
   1      0.7  2.606287
   2      0.8  3.062430
   3      0.9  3.591025
   4        1  3.899835
   5      1.1  4.386511
   6      1.2  4.869966
```

L_1 (m)	t_0^2
0.6	2.353156
0.7	2.60628736
0.8	3.06243
0.9	3.591025
1	3.89983504
1.1	4.38651136
1.2	4.86996624

- សង់ក្រាប

```
import matplotlib.pyplot as plt

# Plotting a scatter table with Table_2 with proper column names
plt.scatter(table_2.iloc[:, 0],
            table_2.iloc[:, 1],
            marker="o",
            linestyle="-",
            color="royalblue")

# Adding labels and title
plt.xlabel('L  (m)')
plt.ylabel('(T₀)²  (s²)')
plt.title('String length vs. Periodicity Squared')

# Show the graph
plt.show()
```

[30] ✓ 2.7s Python



2. Linear Regression

យើងអាចប្រើប្រាស់វិធីសាស្ត្រតំរែតំរង់លីនេអ៊ែរមូលដ្ឋាន ដើម្បីរកបន្ទាត់ដែលសមស្របបំផុតចំពោះចំណុចទាំងអស់។

និយមន័យនៃ Trend line

សមីការបន្ទាត់គឺ

$$y = ax + b$$

ដែល

- y គឺជាអថេរពឹងផ្អែក (អថេរដែលយើងកំពុងព្យាយាមព្យាករណ៍ ឬពន្យល់)។
- x គឺជាអថេរឯករាជ្យ (អថេរដែលយើងអាចគ្រប់គ្រង ឬវាស់វែងបាន)។
- a គឺជាមេគុណប្រាប់ទិស ។

- b គឺជាចំណុចប្រសព្វអ័ក្ស y តម្លៃនៃ y នៅពេលដែល x=0។

ស្របទៅតាមរូបមន្តរបស់យើងនោះ យើងអាចតំណាងពល់អថេរក្នុងសមីការខាងលើ។

$$T = 2\pi\sqrt{\frac{L}{g}} \quad \Rightarrow \quad {T_0}^2 = \frac{4\pi^2}{g}L$$

ក្នុងតម្រង់ $y = ax + b$ នេះយើងតាង

- $y = {T_0}^2$
- $a = \dfrac{4\pi^2}{g}$
- $x = L$
- $b = 0$

តាមវិធីសាស្ត្រវិភាគទិន្នន័យដោយប្រើផែថន [Python] យើងអាចកំណត់មេគុណប្រាប់ទិស និង លេខថេរ បានដោយងាយៗ។ យើងប្រើ `np.polyfit(x, y, 1)`។ នៅក្នុង Excel, យើងប្រើ `=SLOPE(x-range, y-range)`។

```
# Defining the trend_line
trend_line = slope * x + intercept
```

[6] ✓ 0.0s Python

```python
import numpy as np
import pandas as pd

# Define x,y
x = table_2.iloc[:,0]
y = table_2.iloc[:,1]

# Convert columns to numeric
x = pd.to_numeric(x, errors="coerce")
y = pd.to_numeric(y, errors="coerce")

# slope (a) and intercept (b)
slope, intercept = np.polyfit(x,y,1)

print(f"Slope: {slope}")
print(f"Intercept: {intercept}")
```
✓ 0.0s Python

```
Slope: 4.267244199999998
Intercept: -0.30206106571428193
```

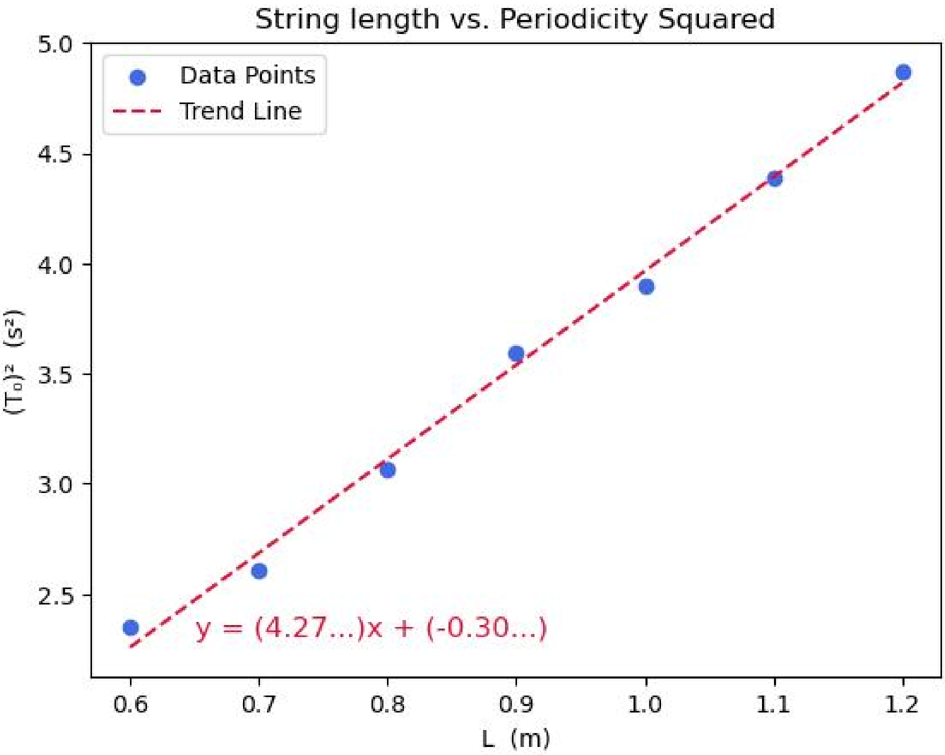
នោះ:

$$\boxed{a = 4.267}$$

$$\boxed{b = -0.302}$$

Graphing

```python
# Creating the scatter plot
plt.scatter(x, y,
            color='royalblue',
            marker='o',
            label='Data Points')

# Plotting the line of best fit
plt.plot(x, trend_line,
         color="crimson",
         linestyle="--",
         label="Trend Line")

# Adding the equation of the line of best fit
equation = f"y = ({slope:.2f}...)x + ({intercept:.2f}...)"
plt.text(x.iloc[0] + 0.05,
         trend_line.iloc[0] + 0.05,
         equation,
         fontsize=12,
         color="crimson")

# Adding labels and title
plt.xlabel('L  (m)')
plt.ylabel('(T₀)²  (s²)')
plt.title('String length vs. Periodicity Squared')

# Show the legend
plt.legend()

# Show the graph
plt.show()
```
[7] ✓ 0.2s Python



Coefficient of determination R^2

នៅក្នុងស្ថិតិ មេគុណកំណត់វាស់វែងថាតើគំរូតំរែតំរង់សមនឹងទិន្នន័យបានល្អប៉ុណ្ណា។ វាបង្ហាញពី សមាមាត្រនៃភាពប្រែប្រួលនៅក្នុងអថេរដែលត្រូវពឹងផ្អែក ដែលអាចព្យាករណ៍បានពីអថេរប្រែប្រួល (ឬច្រើន)។

អត្ថន័យ

- $R^2 = 1$: សមល្អឥតខ្ចោះ (គំរូពន្យល់ពីភាពប្រែប្រួលទាំងអស់នៅក្នុងទិន្នន័យ)។
- $R^2 = 0$: មិនសម (គំរូមិនពន្យល់ពីភាពប្រែប្រួលណាមួយនៅក្នុងទិន្នន័យទេ)។

$$R^2 = 1 - \frac{\sum\left(y_{\text{observed}} - y_{\text{predicted}}\right)^2}{\sum\left(y_{\text{observed}} - \bar{y}\right)^2}$$

ដោយ

- y_{observed} គឺជាតម្លៃពិតប្រាកដនៃទិន្នន័យ។

- $y_{\text{predicted}}$ គឺជាតម្លៃដែលបានព្យាករណ៍ដោយបន្ទាត់ដែលសមស្របបំផុត។
- $\bar{y}$ គឺជាមធ្យមនៃទិន្នន័យដែលបានសង្កេតឃើញ។

យើងនឹងគណនាតម្លៃនេះ ដោយប្រើ NumPy។
ក្នុង Excel យើងប្រើ `=CORREL(X-range, Y-range)^2`។

```python
# numerator
residual_sum = np.sum(
    (y - trend_line)**2
)

# denominator
square_sum = np.sum(
    (y - np.mean(y))**2
)

# calculating R^2
r_squared = 1 - (residual_sum / square_sum)

print(f"R^2 = {r_squared}")
```

[21] ✓ 0.0s Python

```
... R^2 = 0.99466899273353
```

នោះ៖

$$\boxed{R^2 = 99.467\%}$$

ដោយសារតែ $R^2 = 99.46\%$ ទិន្នន័យរបស់យើងមានកំហុសតិចតួចណាស់ ដោយមានភាពប្រែប្រួលតិចតួចណាស់នៅក្នុងទិន្នន័យ។

3. Standard Error σ_a

- σ_a តូច (ជិត 0): ជម្រាលដែលបានប៉ាន់ស្មានគឺត្រឹមត្រូវ ហើយតំណាងឱ្យជម្រាលពិតប្រាកដយ៉ាងជិតស្និទ្ធ។ ការប៉ាន់ស្មានគឺអាចទុកចិត្តបាន។

- σ_a ធំ: ជម្រាលដែលបានប៉ាន់ស្មានមានភាពមិនប្រាកដប្រជាច្រើនជាង ហើយប្រហែលជាមិនតំណាងឱ្យជម្រាលពិតប្រាកដបានល្អទេ។ ការប៉ាន់ស្មានគឺមិនសូវអាចទុកចិត្តបាន

ទេ។

- ទំហំដែលទាក់ទង: ប្រសិនបើ σ_a តូចជាងជម្រាលខ្លាំង ការប៉ាន់ស្មានត្រូវបានចាត់ទុកថាត្រឹមត្រូវជាង។ ប្រសិនបើ σ_a ធំទាក់ទងនឹងជម្រាល ការប៉ាន់ស្មានគឺមិនសូវអាចទុកចិត្តបានទេ។

$$\sigma_a = a\sqrt{\frac{1-R^2}{R^2(N-2)}}$$

ដោយ

- កំហុសស្តង់ដារនៃការតំរែតំរង់ (ឬការប៉ាន់ស្មានជម្រាល)

  $a = 4.267\ldots$

- មេគុណកំណត់ (ដែលវាស់វែងភាពល្អនៃការសម)

  $R^2 = 0.994\ldots$

- ទំហំគំរូ (ចំនួនចំណុចទិន្នន័យ)

  $N = 7$

```python
N = 7

# Calculating a_a
sigma_a = slope * np.sqrt((1 - r_squared) / (r_squared * (N - 2)))

print(f"sigma_a = {sigma_a}")
```

[ ] Python

... sigma_a = 0.13971011238675576

$$\sigma_a = 0.139\ldots$$

ដូច្នេះ ជម្រាលរបស់យើងតំណាងឱ្យជម្រាលពិតប្រាកដយ៉ាងជិតស្និទ្ធ។

4. Standard Error of the Mean Δa_{ale}

Δa_{ale} នេះតំណាងឱ្យភាពមិនប្រាកដប្រជាដោយសារការប្រែប្រួលចៃដន្យក្នុងការវាស់វែង — ត្រូវបានគេស្គាល់ផងដែរថាជាភាពមិនប្រាកដប្រជាអាឡេអាតូរិក (ចៃដន្យ)។

$$\Delta a_{ale} = \frac{\sigma_a}{\sqrt{N}}$$

ដោយ

- នេះគឺជាគម្លាតស្តង់ដារនៃការវាស់វែង — វាប្រាប់ថាតើចំណុចទិន្នន័យមានទំនោរខុសគ្នាពីមធ្យមប៉ុណ្ណា

  $\sigma_a = 0.139\ldots$

- ចំនួនដងនៃការវាស់ ឬចំណុចទិន្នន័យ។

  $N = 7$

```python
# Calcluate a_ale

ale_a = sigma_a / np.sqrt(N)
print(f"ale_a = {ale_a}")
```

[36] ✓ 0.0s Python

```
... ale_a = 0.052805459002320046
```

$$\Delta a_{ale} = 0.0528$$

ដូច្នេះ កំហុសចៃដន្យមានឥទ្ធិពលតិចតួចបំផុតលើការវាស់វែងរបស់យើង។ នេះបង្ហាញពី ភាពជាក់លាក់ខ្ពស់ នៅក្នុងទិន្នន័យ។

5. Uncertainty of g Δa_{ale}

- Δg **តូច** : បង្ហាញថាភាពមិនប្រាកដប្រជានៅក្នុង g (ល្បឿនបង្កើនដោយសារទំនាញផែនដី) គឺទាបខ្លាំង មានន័យថាការវាស់វែង g មានភាពជាក់លាក់ខ្ពស់។

- Δg **ធំ** : បង្ហាញពីភាពមិនប្រាកដប្រជាច្រើនជាងក្នុងការវាស់វែង g មានន័យថាទិន្នន័យមិនសូវអាចទុកចិត្តបាន ឬការវាស់វែងមានរឹមនៃកំហុសខ្ពស់ជាង។

- **ទំនាក់ទំនងផ្ទាល់ជាមួយ** Δa_{ale}: តម្លៃនៃ Δg ត្រូវបានជះឥទ្ធិពលដោយផ្ទាល់ដោយភាពមិនប្រាកដប្រជានៅក្នុង a (ចម្ងាយ ឬប្រវែង) ដូច្នេះប្រសិនបើ Δa_{ale} តូច នោះ Δg ក៏នឹងតូចដែរ ដែលបង្ហាញពីភាពជាក់លាក់ល្អជាងនៅក្នុងការពិសោធន៍។

$$\Delta g = \left| \frac{\delta g}{\delta a} \Delta a_{aleatoir} \right|$$

ដែល

$$g = \frac{4\pi^2}{a} \quad \Rightarrow \quad \frac{\delta g}{\delta a} = -\frac{4\pi^2}{a^2}$$

នាំឲ

$$\Delta g = \left| \frac{\delta g}{\delta a} \Delta a_{ale} \right|$$

$$= \left| -\frac{4\pi^2}{a^2} \cdot \Delta a_{ale} \right|$$

```python
# Calculate D_g
D_g = np.abs((-4 * np.pi**2) * (ale_a) / (slope**2))

print(f"D_g = {D_g}")
```

[37] ✓ 0.0s Python

```
D_g = 0.11448367580452858
```

នោះ

$$\Delta g = 0.114\ldots$$

6. Determine g

យើងដឹងថា

$$a = \frac{4\pi^2}{g} \quad \Rightarrow \quad g = \frac{4\pi^2}{a}$$

ដែល $a = 4.267\ldots$

```python
import math

# Calculating for g
g = (4 * math.pi ** 2) / slope

print(f"g = {g}")
```

[12] ✓ 0.0s Python

```
g = 9.251501848513252
```

ដូចនេះ តាមការពិសោធន៍របស់យើង

$$\boxed{g = 9.251\ldots \quad m/s^2}$$

7. Accuracy and Precision

$$\begin{cases} \text{Accuracy } \% = \dfrac{|g - 9.81|}{9.81} \times 100\% \\ \text{Precision } \% = \dfrac{\Delta g}{g} \times 100\% \end{cases}$$

```python
# Calculating Accuracy
accuracy = np.abs(g - 9.81) / 9.81

# Calculating Precision
precision = D_g / g

print(f"accuracy = {accuracy}")
print(f"precision = {precision}")
```

[13] ✓ 0.0s Python

```
accuracy = 0.056931513913022286
precision = 0.012374604434946581
```

$$\begin{cases} \text{Accuracy } \% = 5.693\% \\ \\ \text{Precision } \% = 1.237\% \end{cases}$$

III. សេចក្តីសន្និដ្ឋាន

នៅក្នុងការពិសោធន៍នេះ តម្លៃសំទុះទំនាញផែនដី (g) ត្រូវបានកំណត់ដោយប្រើទិន្នន័យពិសោធន៍ គឺផ្តល់លទ្ធផល $g = 9.251\, m/s^2$ ។ តម្លៃនេះខុសគ្នាពីតម្លៃដែលពិតប្រាកដ $g = 9.81\, m/s^2$ ដែលបង្ហាញពីកម្រិតនៃ ភាពមិនត្រឹមត្រូវ ក្នុងការវាស់វែង។ ភាពត្រឹមត្រូវដែលបានគណនាគឺ 5.693% បង្ហាញពីគម្លាតពីតម្លៃទ្រឹស្តី ដែលអាចបណ្តាលមកពីកំហុសពិសោធន៍ ឬដែនកំណត់ក្នុងការរៀបចំការវាស់វែង។

ទោះបីជាមានភាពមិនត្រឹមត្រូវនេះក៏ដោយ ភាពជាក់លាក់ នៃរង្វាស់គឺខ្ពស់គួរសម ដោយតម្លៃភាពជាក់លាក់គឺ 1.127% ។ នេះបង្ហាញថា ទោះបីជាទិន្នន័យមិនជិតដឹតនឹងតម្លៃពិតនៃ g ក៏ដោយ របៀបនៃការរង្វាស់ម្តងហើយម្តងទៀតគឺស្របគ្នាដោយគ្មានការលំអៀងធំពេកឡើយ ដែលបង្ហាញពីបច្ចេកទេសរង្វាស់ ដែលអាចទុកចិត្តបានទាក់ទងនឹងការប្រើប្រាស់ទិន្នន័យ។