實驗物理學（二）
實驗日誌

Fundamental Python
Chi-square fitting - 3

Group 2

洪　瑜 B125090009
黃巧涵 B122030003
洪懌平 B102030019

2025/05/13

1 實驗步驟、初步結果

1.1 Practice 1

1. By using previous knowledge, choose your own function to generate data points.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.stats import chi2
```
✓ 0.0s Python

```
np.random.seed(42)
```
✓ 0.0s Python

2. Generate 1000 sets of data points with the same function with normally distributed random noise.

```
def parabolic(x, a, b, c):
    return a * x**2 + b * x + c
```
✓ 0.0s Python

```
# Generate synthetic data
true_a = 2.0
true_b = 1.0
true_c = 0.5

sigma0 = 3.0

# Generate data
n_data = 10
n_sets = 1000

x = np.linspace(-5, 5, n_data)
```
✓ 0.0s Python

3. Fit these data points with the function you used to generate the data points.

4. Calculate the χ^2 values of each fitting result.

```python
all_data    = []
fit_results = []
chi2_values = []
```
✓ 0.0s Python

```python
for i in range(n_sets):
    # Generate synthetic data with noise
    y = parabolic(x, true_a, true_b, true_c) + np.random.normal(0, sigma0, size=x.shape)
    all_data.append(y)
    # Fit the data
    popt, pcov = curve_fit(parabolic, x, y)

    # Calculate chi-squared
    residuals = y - parabolic(x, *popt)
    chi2_value = np.sum((residuals / sigma0)**2)

    fit_results.append(popt)
    chi2_values.append(chi2_value)
```
✓ 0.1s Python

5. Create your own **p-value** calculator to calculate the **p-value** of each χ^2 instead of using functions in `scipy.stats`.

```python
from math import gamma

def chi2_p_value(chi2_value, dof):
    """
    Calculate the p-value for a given chi-square value and degrees of freedom (dof).
    """
    # Compute the incomplete gamma function (upper tail)
    incomplete_gamma = gamma(dof / 2) - sum((chi2_value / 2)**k / gamma(k + 1) for k in range(dof // 2))

    # Compute the p-value
    p_value = incomplete_gamma / gamma(dof / 2)
    return p_value

# Example usage
dof = 7  # degrees of freedom
p_values = [chi2_p_value(chi2, dof) for chi2 in chi2_values]
print(p_values[:10])  # Print the first 10 p-values
```
✓ 0.0s Python

```python
# Display the results
plt.figure(figsize=(10, 6))

x_plot = np.linspace(-8, 8, 500)
plt.plot(x_plot, parabolic(x_plot, true_a, true_b, true_c), color='C4', label='True model',
         linewidth=4, linestyle='-')

sampled_sets = np.random.choice(n_sets, 200, replace=False)
for i in sampled_sets:
    plt.scatter(x, all_data[i, :], color='C1', alpha=0.3)
    plt.plot(x_plot, parabolic(x_plot, *fit_results[i]), color='plum', alpha=0.1,
             linewidth=1.5, linestyle=':')

plt.scatter([], [], color='C1', alpha=0.3, label='Sample points')
plt.plot([], [], color='plum', alpha=0.5, linewidth=1.5, label='Fitted lines', linestyle=':')

plt.title('Sample scatter plot', fontsize=16)
plt.xlim(-5.5, 5.5)
plt.ylim(-10, 70)
plt.xlabel('x', fontsize=14)
plt.xticks(fontsize=12)
plt.ylabel('y', fontsize=14)
plt.yticks(fontsize=12)
plt.legend(fontsize=14, loc='upper left', framealpha=1)
plt.savefig("output1_1.pdf", transparent=True)
plt.show()
```
✓ 1.0s Python

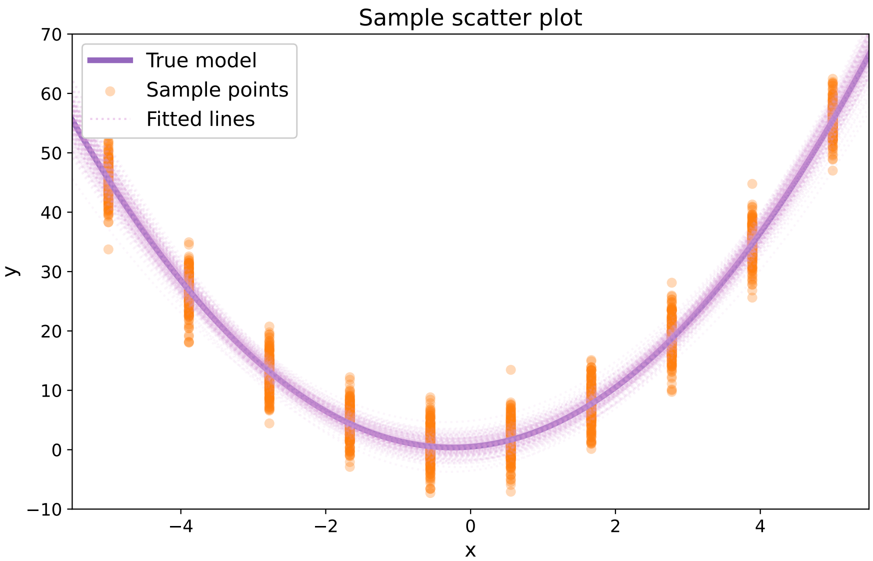

Figure 1: 使用已有的知識選擇的數學函數，並加上隨機雜訊產生之數據，及每組數據的擬合曲線

6. Plot the histogram of χ^2.

```python
#chi square直方圖
plt.figure(figsize=(10, 6))

plt.hist(chi2_values, bins=30, density=True, alpha=0.7, label='χ²', color='C6',
         histtype='stepfilled')

x_chi = np.linspace(0, 50, 500)
plt.plot(x_chi, chi2.pdf(x_chi, df=7), linestyle='-', color='maroon', label='PDF (dof=7)',
         linewidth=3.5)
plt.plot([np.mean(chi2_values), np.mean(chi2_values)], [0, 0.2], linestyle='--',
         color="purple", lw=2, label='__no_legend__')
plt.text(np.mean(chi2_values)+0.5, 0.12, f'$\mu$={np.mean(chi2_values):.3f}',
         fontsize=14, color='purple')

plt.title('Chi-square Distribution (dof=8)', fontsize=16)
plt.xlabel('Chi-square value', fontsize=14)
plt.ylabel('Probability density', fontsize=14)
plt.xlim(left=0, right=30)
plt.ylim(bottom=0, top=0.13)
plt.legend(fontsize=14, framealpha=1)
plt.savefig("output1_2.pdf", transparent=True)
plt.show()
```

✓ 0.1s Python

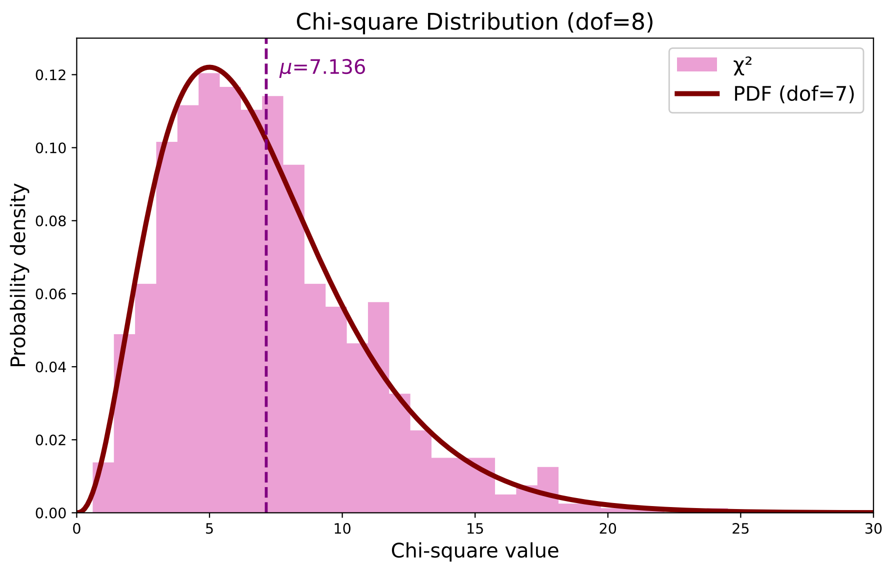

Figure 2: χ^2直方圖與$dof = 7$時的PDF

7. Mark a certain χ^2 value with its corresponding **p-value** on the histogram.

```python
# Select a specific χ² value and calculate its p-value
specific_chi2_value = chi2_values[i]  # Use the χ² value at index i
specific_p_value = chi2.sf(specific_chi2_value, df=dof)  # Calculate p-value using survival function

# Plot the histogram
plt.figure(figsize=(10, 6))
plt.hist(chi2_values, bins=30, density=True, alpha=0.7, label='χ²', color='C6', histtype='stepfilled')

# Plot the χ² PDF
x_chi = np.linspace(0, 50, 500)
plt.plot(x_chi, chi2.pdf(x_chi, df=dof), linestyle='-', color='maroon', label='PDF (dof=7)', linewidth=3.5)

# Mark the specific χ² value
plt.axvline(specific_chi2_value, color='blue', linestyle='--', linewidth=2, label=f'χ²={specific_chi2_value:.2f}')
plt.text(specific_chi2_value + 0.5, 0.1, f'p={specific_p_value:.3f}', fontsize=14, color='blue')

# Add labels, title, and legend
plt.title('Chi-square Distribution (dof=8)', fontsize=16)
plt.xlabel('Chi-square value', fontsize=14)
plt.ylabel('Probability density', fontsize=14)
plt.xlim(left=0, right=30)
plt.ylim(bottom=0, top=0.13)
plt.legend(fontsize=14, framealpha=1)

# Save and show the plot
plt.savefig("output1_3.pdf", transparent=True)
plt.show()
```
✓ 0.1s Python

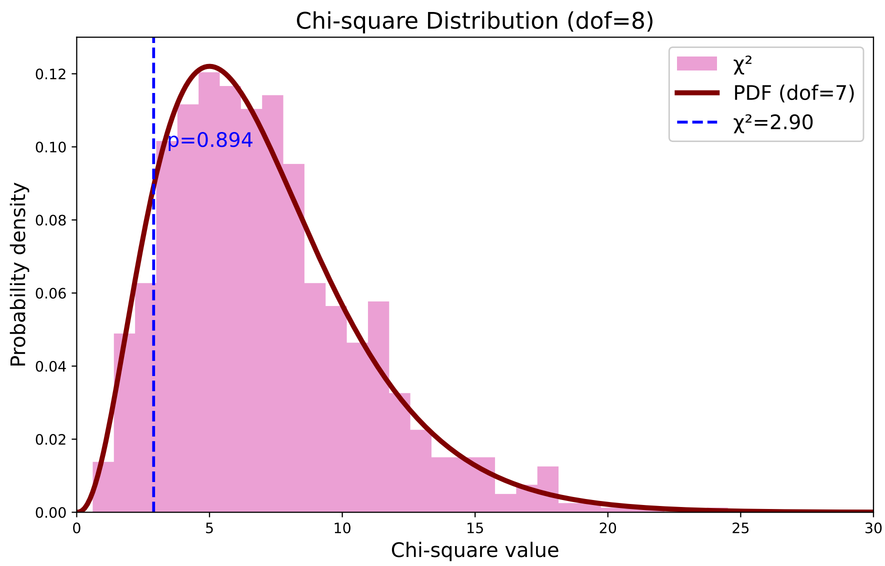

Figure 3: 標上對應之P-value

```python
# Sort the chi-squared values
sorted_chi2_values = np.sort(chi2_values)

# Compute the cumulative probabilities
cumulative_probabilities = np.arange(1, len(sorted_chi2_values) + 1) / len(sorted_chi2_values)

# Plot the CDF
plt.figure(figsize=(10, 6))
plt.plot(sorted_chi2_values, cumulative_probabilities, label='Empirical CDF', color='C0', linewidth=3.5)

# Overlay the theoretical CDF for comparison
theoretical_cdf = chi2.cdf(sorted_chi2_values, df=dof)
plt.plot(sorted_chi2_values, theoretical_cdf, label='Theoretical CDF (dof=7)', color='C3', linestyle='--', linewidth=3.5)

# Add labels, title, and legend
plt.title('Cumulative Distribution Function (CDF)', fontsize=16)
plt.xlabel('Chi-square value', fontsize=14)
plt.ylabel('Cumulative probability', fontsize=14)
plt.legend(fontsize=14, framealpha=1)

# Save and show the plot
plt.savefig("output1_4.pdf", transparent=True)
plt.show()
```

✓ 0.1s Python

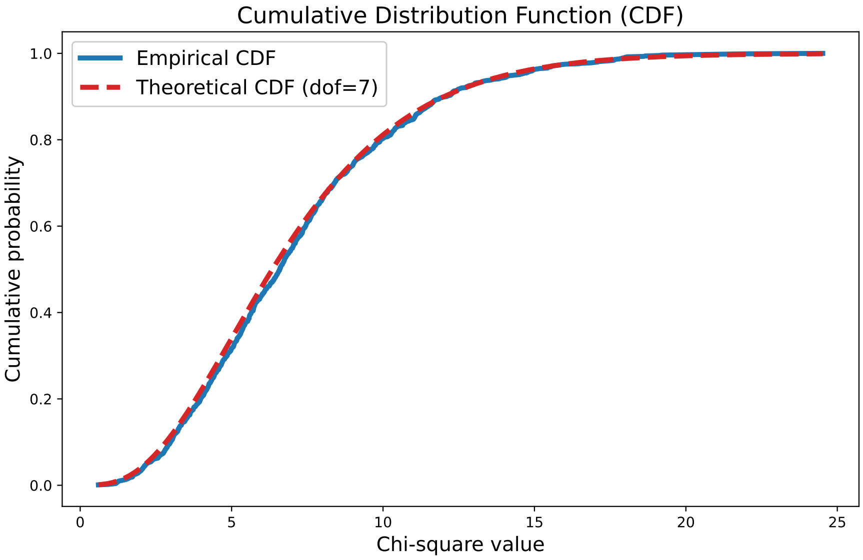

Figure 4: Cumulative Distribution Function

1.2 Practice 2

1. Using prior knowledge, select an appropriate function with noise to generate your dataset.

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.stats import chi2
```
✓ 0.0s Python

```python
np.random.seed(42)
```
✓ 0.0s Python

```python
def parabolic(x, a, b, c):
    return a * x**2 + b * x + c
```
✓ 0.0s Python

```python
# Generate synthetic data
true_a = 2.0
true_b = 1.0
true_c = 0.5

sigma0 = 3.0

# Generate data
n_data = 10
n_sets = 1000

x = np.linspace(-5, 5, n_data)
```
✓ 0.0s Python

```python
# Display the results
plt.figure(figsize=(10, 6))

x_plot = np.linspace(-8, 8, 500)
plt.plot(x_plot, parabolic(x_plot, true_a, true_b, true_c), color='C4', label='True model',
         linewidth=4, linestyle='-')

sampled_sets = np.random.choice(n_sets, 200, replace=False)
for i in sampled_sets:
    plt.scatter(x, all_data[i, :], color='C1', alpha=0.3)
    plt.plot(x_plot, parabolic(x_plot, *fit_results[i]), color='plum', alpha=0.1,
             linewidth=1.5, linestyle=':')

plt.scatter([], [], color='C1', alpha=0.3, label='Sample points')
plt.plot([], [], color='plum', alpha=0.5, linewidth=1.5, label='Fitted lines', linestyle=':')

plt.title('Sample scatter plot', fontsize=16)
plt.xlim(-5.5, 5.5)
plt.ylim(-10, 70)
plt.xlabel('x', fontsize=14)
plt.xticks(fontsize=12)
plt.ylabel('y', fontsize=14)
plt.yticks(fontsize=12)
plt.legend(fontsize=14, loc='upper left', framealpha=1)
plt.savefig("output2_1.pdf", transparent=True)
plt.show()
```

✓ 1.1s Python

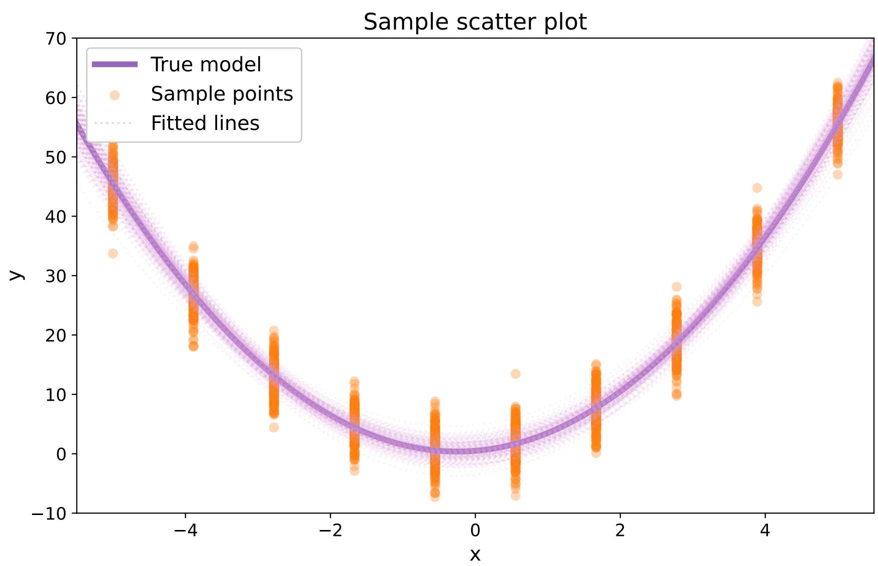

Figure 5: 使用已有的知識選擇的數學函數，並加上隨機雜訊產生之數據，及每組數據的擬合曲線

2. Plot a histogram of the χ^2 values obtained from fitting each dataset.

```python
#chi square直方圖
plt.figure(figsize=(10, 6))

plt.hist(chi2_values, bins=30, density=True, alpha=0.7, label='χ²', color='C6',
         histtype='stepfilled')

x_chi = np.linspace(0, 50, 500)
plt.plot(x_chi, chi2.pdf(x_chi, df=7), linestyle='-', color='maroon', label='PDF (dof=7)',
         linewidth=3.5)
plt.plot([np.mean(chi2_values), np.mean(chi2_values)], [0, 0.2], linestyle='--',
         color="purple", lw=2, label='__no_legend__')
plt.text(np.mean(chi2_values)+0.5, 0.12, f'$\mu$={np.mean(chi2_values):.3f}',
         fontsize=14, color='purple')

plt.title('Chi-square Distribution (dof=7)', fontsize=16)
plt.xlabel('Chi-square value', fontsize=14)
plt.ylabel('Probability density', fontsize=14)
plt.xlim(left=0, right=30)
plt.ylim(bottom=0, top=0.13)
plt.legend(fontsize=14, framealpha=1)
plt.savefig("output2_2.pdf", transparent=True)
plt.show()
```

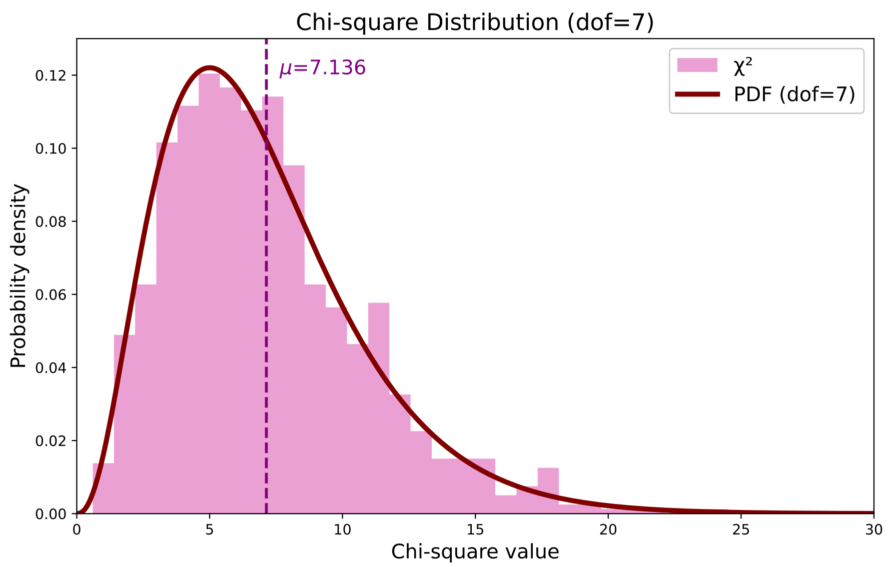

Figure 6: 每組數據對應的χ^2值之直方圖，與$dof = 7$時對應的PDF

3. Mark the **p-value** get from the fitting result of `mean(y)`.

4. You might get a χ^2 of the `mean(y)` fitting result close to zero and a **p-value** close to 1.

```
mean_y = np.mean(all_data, axis=0)
parms_mean, _ = curve_fit(parabolic, x, mean_y)
chisq_mean = np.sum((mean_y - parabolic(x, *parms_mean))**2 / sigma0**2)
p_mean = 1 - chi2.cdf(chisq_mean, df=7)

print('chi-square of mean(y)= ', chisq_mean)
print('p-value of mean(y)= ', p_mean)
```

```
chi-square of mean(y)=  0.004356074591544417
p-value of mean(y)=  0.9999999999586144
```

Figure 7: 平均數據的χ^2值和其對應的$P - value$

5. Also mark 5% significance level, and 95% significance level on the histogram.

```
# Calculate the 5% and 95% significance levels
significance_5 = chi2.ppf(0.05, df=7)
significance_95 = chi2.ppf(0.95, df=7)

# Plot the histogram with significance levels
plt.figure(figsize=(10, 6))

plt.hist(chi2_values, bins=30, density=True, alpha=0.7, label='χ²', color='C6', histtype='stepfilled')

x_chi = np.linspace(0, 50, 500)
plt.plot(x_chi, chi2.pdf(x_chi, df=7), linestyle='-', color='maroon', label='PDF (dof=7)', linewidth=3.5)

# Mark the mean chi-square value
# plt.plot([mean_chi2, mean_chi2], [0, 0.2], linestyle='--', color="purple", lw=2, label='Mean χ²')
# plt.text(mean_chi2 + 0.5, 0.11, f'$\mu$={mean_chi2:.3f}\n$p$={p_value:.3f}', fontsize=14, color='purple')

# Mark the 5% and 95% significance levels
plt.axvline(significance_5, color='blue', linestyle='--', linewidth=2, label='5% Significance Level')
plt.axvline(significance_95, color='green', linestyle='--', linewidth=2, label='95% Significance Level')
plt.plot([np.mean(chi2_values), np.mean(chi2_values)], [0, 0.2], linestyle='--',
         color="purple", lw=2, label='__no_legend__')
plt.text(np.mean(chi2_values)+0.5, 0.12, f'$\mu$={np.mean(chi2_values):.3f}',
         fontsize=14, color='purple')


plt.title('Chi-square Distribution (dof=7)', fontsize=16)
plt.xlabel('Chi-square value', fontsize=14)
plt.ylabel('Probability density', fontsize=14)
plt.xlim(left=0, right=30)
plt.ylim(bottom=0, top=0.13)
plt.legend(fontsize=14, framealpha=1)
plt.savefig("output2_4.pdf", transparent=True)
plt.show()
```

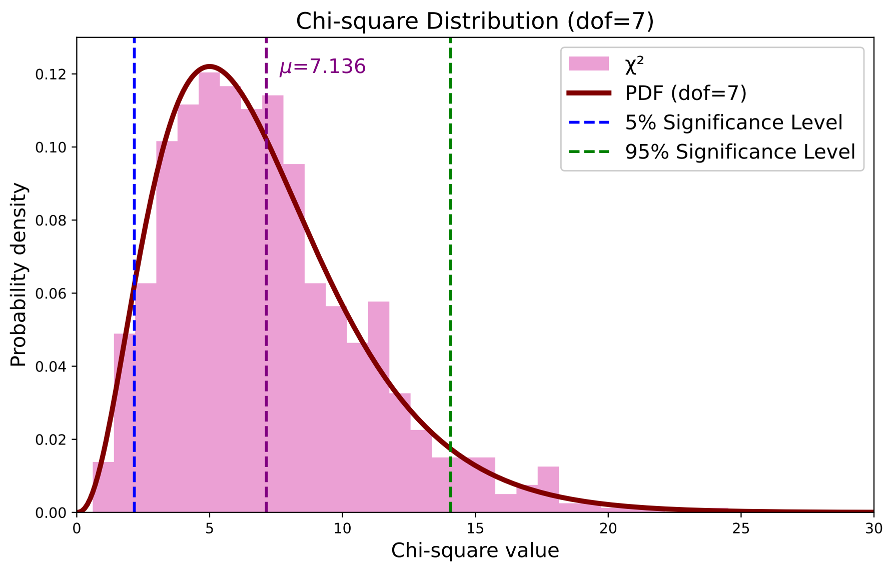

Figure 8: 由Fig.6的基礎下加入5%和95%的Significance level

2 初步分析

2.1 Practice 1

觀察Fig.1，紫色粗線爲true model，設定爲：$y = 2x^2 + x + 0.5$；擬合線爲紫色虛線。可以看出部分擬合線非常貼近眞實模型，表示這部分擬合線的資料誤差小、擬合結果準確，有些則有明顯偏離，表示該組資料雜訊大或者擬合出來的參數有誤差。擬合線與資料點大致上皆吻合，但不一定會完全與眞實曲線重合。雖然使用相同的擬合方式，但因爲受到noise的隨機數值影響，擬合出的結果會不相同，即是擬合結果的variability (變異性)。

而Fig.2是擬合結果的χ^2分布直方圖，並與理論上的卡方分布PDF進行比較，自由度爲$dof = 10`3 = 7$。從圖中可見實際χ^2值的直方圖與理論的卡方 PDF 曲線高度相符，且圖中χ^2平均值（虛線）與理論自由度相符，表示擬合過程是合理的，且高斯雜訊與誤差模型（$\sigma = 3$）和擬合模型之間的假設是一致的，也沒有明顯系統性偏誤。實際χ^2分布與理論曲線在形狀上相符，代表殘差的統計性質符合高斯假設。整體χ^2值沒有出現明顯右偏或重尾現象，代表大多數擬合皆落在合理範圍以內，沒有出現明顯異常。

Fig.3在原有χ^2分布直方圖基礎上，再多標示出一組擬合結果的$\chi^2 = 2.90$及其對應之$p-value = 0.894$。此處所選χ^2值爲X（以藍線標示），對應的 $p - value$ 爲 Y，表示該擬合結果在所有結果中的相對位置。若 $p - value$ 明顯偏小（例如 < 0.05），可視爲模型擬合不良；若 $p - value$ 適中，則該擬合與理論模型一致，未有顯著偏差。

最後，Fig.4爲$Chi - square$ 分布的CDF。藍色實線爲實際模擬資料的經驗累積分布（$Empirical\ CDF$）；紅色虛線爲理論上的卡方分布累積函數（$Theoretical\ CDF$）（自由度 $= 7$）。這張圖畫出了模擬出的χ^2值之累積分布（藍線）與理論卡方分布（自由度7）的CDF（紅線）對比結果。模擬結果與理論分布相符，顯示生成資料的統計性質與理論預期相差不大。這種比較可以當驅是擬合殘差是否爲常態分布的間接驗證方式之一，如果有明顯的差異，可能代表模型設計、誤差估計或$noise$生成中出現問題。

2.2 Practice 2

從Fig.5可以看出我們成功在$y = ax^2 + bx + c$這個模型上加入不同程度的雜訊；並可看出越接近true model時，每組雜訊所對應的擬合曲線就越密集。從Fig.6來看，可見我們所建立的不同資料組的擬合所對應是χ^2值有符合當自由爲7時，其PDF的趨勢，而χ^2 value平均值爲7.136，透過先前所學，將其除以自由度：

$$\chi^2_\nu = \frac{\chi^2}{\nu} = \frac{7.136}{7} \approx 1.019 \approx 1$$

可確認其擬合程度很好。而Fig.8可以看見我們χ^2的平均值是落在5%∼95% significance level之間，表示我們的擬合結果是在正常的範圍內，是合理的模型；最後，Fig.7得到利用平均的數據做擬合時，得到的χ^2 value趨近於0、P-value接近1，符合理論結果。

3 具體說明實驗遇到的問題，或分析可能的問題

在practice 2中，我們原先計算平均數據的χ^2 value並沒有接近0，經過檢查才發現將其寫成`parms_mean, _ = curve_fit(parabolic, x, y)`，這會導致我們使用的y值是最後一組數據的y值，並非其平方，導致結果不符預期；將其更正爲`parms_mean, _ = curve_fit(parabolic, x, mean_y)`，就獲得到理想結果了（詳情見步驟4）。