

HW8

系級	學號	姓名
電子三乙	B1027234	林永濬
電子三乙	B1027236	蕭銘宏

1 統計資料的方法

以下是程式碼步驟：

1. 定義要分析的檔案列表 `files`，包含了四個檔案的名稱：'T12_hit_00.1.prn'、'T12_hit_01.1.prn'、'T12_hit_10.1.prn' 和 'T12_hit_11.1.prn'。
2. 使用 `for` 迴圈遍歷檔案列表，對每個檔案進行分析。在每次迴圈中，將當前的檔案名稱賦值給變數 `filename`，然後調用 `ReadData` 函數對該檔案進行讀取和分析。
3. 在 `ReadData` 函數中，使用 `dlmread` 函數讀取指定的檔案，並提取第三列到第八列的數據，存儲在 `columnData` 元胞陣列中。
4. 根據檔案名稱設置對應的變數名稱，存儲在 `variableNames` (Cell Array) 中。不同的檔案有不同的變數名稱，例如 'ch1'、'ch2'、'ch3' 等。
5. 創建一個新的圖形窗口，並使用 `subplot` 函數將圖形窗口劃分為 2 行 3 列的子圖。
6. 使用 `for` 迴圈遍歷每一列數據，對每一列數據繪製長條圖和分布曲線。在每個子圖中，使用 `histogram` 函數繪製數據的直方圖，並使用 `fitdist` 函數擬合數據的正態分布曲線。
7. 計算並顯示每一列數據的平均值和標準差。將前三列數據的統計資訊顯示在一個表格中，並將後三列數據的統計資訊顯示在另一個表格中。使用 `fprintf` 函數進行格式化輸出。
8. 在每個檔案分析完成後，輸出分隔線，以示區分不同檔案的分析結果。

透過這樣的統計分析方法，程式碼可以對多個資料檔案進行批次處理，並生成對應的統計圖表和數據。這種方法可以快速了解資料的分佈情況和統計特性。

2 結果

T12_hit_00.1.prn:

```
=====
Ch      Mean      Std dev
=====
ch1     71.97      1.12
ch2     85.91      1.78
ch3     97.64      1.89
=====
ChCa1   Mean      Std dev
=====
ch1Ca1  137.17      2.48
ch2Ca1  122.55      2.21
ch3Ca1  100.03      1.63
=====
===== 分析完成: T12_hit_00.1.prn
```

T12_hit_01.1.prn:

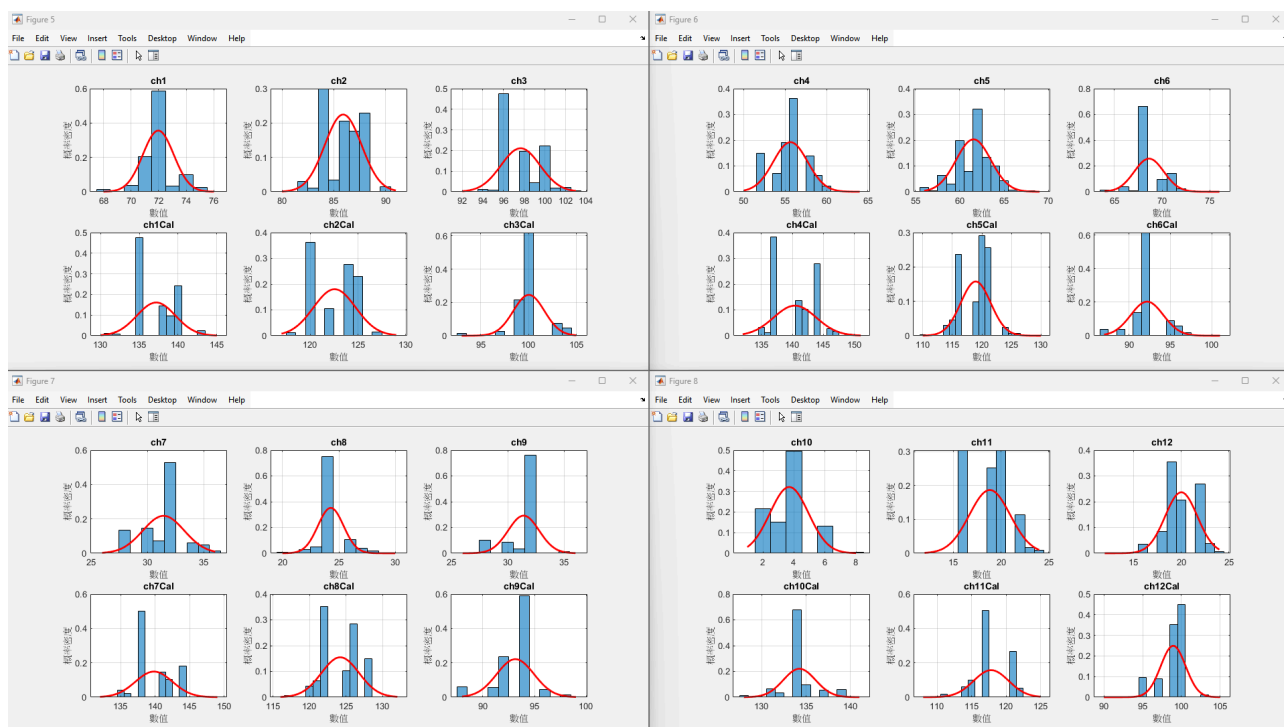
```
=====
Ch      Mean      Std dev
=====
ch4     55.64      2.07
ch5     61.56      1.96
ch6     68.64      1.55
=====
ChCal   Mean      Std dev
=====
ch4Cal  140.41      3.42
ch5Cal  118.96      2.53
ch6Cal  92.19       1.97
=====
===== 分析完成: T12_hit_01.1.prn
```

T12_hit_10.1.prn:

```
=====
Ch      Mean      Std dev
=====
ch7     31.43      1.83
ch8     24.28      1.13
ch9     31.40      1.37
=====
ChCal   Mean      Std dev
=====
ch7Cal  139.81      2.67
ch8Cal  124.19      2.58
ch9Cal  93.14       1.80
=====
===== 分析完成: T12_hit_10.1.prn
```

T12_hit_11.1.prn:

```
=====
Ch      Mean      Std dev
=====
ch10    3.70       1.24
ch11    18.83      2.14
ch12    20.01      1.69
=====
ChCal   Mean      Std dev
=====
ch10Cal 134.24      1.81
ch11Cal 117.84      2.54
ch12Cal 98.94      1.60
=====
===== 分析完成: T12_hit_11.1.prn
```



2.1 分析

下表比較了每個 channel 的數據分布範圍、數據集中趨勢和常態分佈曲線。觀察發現除了 [channel 3](#) 與 [channel 12](#) 之外，**校正過後的數據有較分散的趨勢**。

Channel	數據分布範圍	數據集中趨勢	常態分佈曲線
ch1	70-75	集中在 72-73	較集中
ch1Cal	130-140	集中在 134-136	較分散
ch2	80-90	集中在 84-87	較集中
ch2Cal	120-130	集中在 122-125	較分散
ch3	92-104	集中在 95-99	較分散
ch3Cal	95-105	集中在 99-101	較集中
ch4	50-60	集中在 53-57	較集中
ch4Cal	135-150	集中在 137-142	較分散
ch5	55-70	集中在 58-64	較集中
ch5Cal	110-130	集中在 118-122	較分散
ch6	65-75	集中在 68-71	較集中
ch6Cal	90-100	集中在 94-96	較分散
ch7	如圖	如圖	較集中
ch7Cal	如圖	如圖	較分散
ch8	如圖	如圖	較集中
ch8Cal	如圖	如圖	較分散
ch9	如圖	如圖	較集中
ch9Cal	如圖	如圖	較分散
ch10	如圖	如圖	較集中
ch10Cal	如圖	如圖	較分散
ch11	如圖	如圖	較集中
ch11Cal	如圖	如圖	較分散
ch12	如圖	如圖	較分散
ch12Cal	如圖	如圖	較集中

3 心得

通過編寫和研究這份程式碼，我們對於使用 MATLAB 進行資料分析和統計處理有了更深入的認識。以下是我們的一些心得：

1. MATLAB 提供了豐富的函數和工具箱，可以方便地進行資料讀取、處理和可視化。使用 `dlmread` 函數可以輕鬆地讀取檔案，而 `histogram` 和 `fitdist` 函數則可以快速繪製直方圖和擬合分布曲線。
2. 透過使用 **Cell Array**，可以方便地存儲和管理不同的數據列和變數名稱。這種方式使得程式碼更加靈活和擴展性，能夠處理不同數量和類型的資料列。
3. 在進行批次處理時，使用 `for` 迴圈可以遍歷多個檔案，對每個檔案進行相同的分析和處理。這種方式可以提高效率，減少重複的程式碼。
4. 在實際應用中，可以根據具體的需求和資料特點，進一步優化和擴展程式碼。例如，可以新增更多的統計指標、繪製其他類型的圖表、處理缺失值或異常值等。

這份作業讓我們練習如何使用 MATLAB 進行資料分析和統計處理的基本流程和技巧，這對於日後處理類似的資料分析任務將非常有幫助。