Reports of Data Mining Project 1

What do you observe in the below 4 scenarios? What could be the reason? Include the Runtime statistics, # of frequent patterns, and # of results under 4 scenarios.

在這項實驗中，關於 low boundary 與 high boundary 我在 sup 分別設置成 0.05 與 0.2，而在 conf 分別設置成 0.1 與 0.7，並為了凸顯出不同設置的差異，選擇以 apriori algorithm 去進行實驗。以下表格為實驗結果：

Apriori	Runtime(s)	# of Freq_patterns	# of results
Low sup, Low conf	56.88	5354	57193
Low sup, High conf	57.84	5354	2589
High sup, Low conf	1.37	117	326
High sup, High conf	1.43	117	42

可以看到在 runtime 方面 High sup 遠快於 Low sup，因為 min_sup 的閥值高所以能篩選掉許多不符合的 pattern，使得後續遞迴的情況減少，而 min_conf 的調整沒有什麼影響到 runtime，這是因為 conf 的計算需要先有 freq_patterns 才能做運算，而真正影響到計算時間的正是找 freq_patterns 的過程，而此過程只有與 min_sup 相關，可以看到 freq_patterns 數目不會因 conf 不同而不同。
conf 所影響的就是後續根據 freq_patterns 計算關聯法則時，才會在這裡篩掉低於 min_conf 的 results，所以相比 low conf，high conf 的 results 數目會大幅減少。

FP_growth	Runtime(s)
Low sup, Low conf	1.92
Low sup, High conf	1.72
High sup, Low conf	0.41
High sup, High conf	0.42

FP_growth 明顯減少時間複雜度的問題，但若 min_sup 調得更高，可能導致 apriori 因為簡單架構反而較快，而 FP_growth 因基礎架構複雜因而較慢。

Bonus: kaggle-groceries

可以明顯感受到 apriori 與 fp_growth 兩者時間差異性，而關於 min_sup, min_conf 的設置，到了(0.001, 0.2)還只有兩筆 results，能了解若要運用於實際場合可能需要設置較小 min_sup, min_conf 或是更大的資料集才能有效截取出結果。