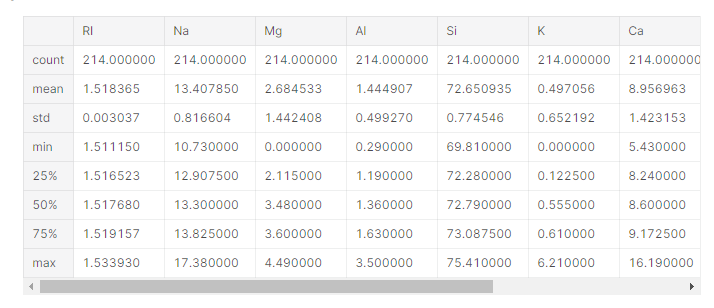
首先是導入 dataset，獲取其所有的column name (用作cleaning或者分析)

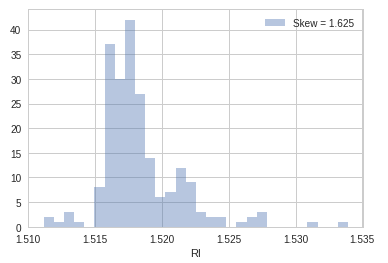
df.describe來看數據的資料，例如mean, standard deviation等，用來判斷是否和如果處理data

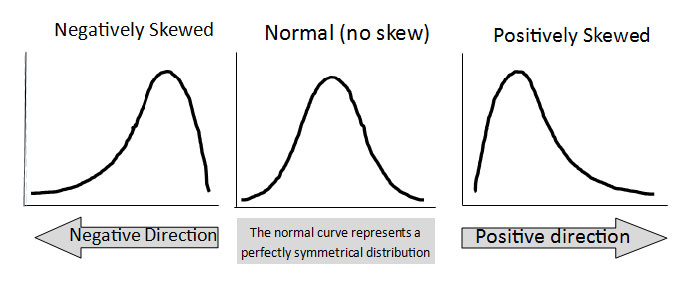


從description中能夠看到，Si和Fe的mean相差很遠，所以如果我們使用某些算法例如 logistic regression, 我們就需要進行處理令他們在same scale.

而輪到label，我們也需要看看label的分佈，例如這個dataset裡面，type 1 and type 2佔了整個dataset的67% label, 可知很unbalance.

然後是data visualization, 將每個data column的data都plot出來，從而更容易地分析。這裡首先使用的是skew. Skew其實就是一種measure distribution的東西，if no skew, data的distribution應該是normal distribution那樣。用





然後是一個detect outlier的function，for loop 每一個column，計算整個column所有data的所在的distribution的位置，如果大於75%或者少於25%都記下來。遍歷所有column後，每row數一下有多少個column是被記下來的，如果有多於兩個column被記下，那就移除整個row.

為了更好地理解，所以在執行detect outlier前，我們先畫一個boxplot來看每個column的distribution。

然後畫pairplot (heatmap) 去看每個data column之間，數據的關聯。從畫出的圖可以得出，column RI and Ca有強關聯，或許需要進行pca來decorrelate (pca是什麼後面再說)

用df.info() 來查找na數據量，有的話要做處理（但是沒有）

然後便是remove outlier，上面已經說過了。移除outlier後，再每個column plot一次data，去看skew.

然後是將dataset分為input feature和label,再使用train\_test\_split來分為train set和test set.

進行box cox而令dataset 符合normal distribution (box cox就是一個用於data 正態化 skewness -> 0 的方法)，任何transformation都只能在training set進行，不可以在test set做 (這不是preprocessing)

Fit data with xgboost (don’t know what this means; just one machine learning type)

perform a PCA on the features to decorrelate the ones that are linearly dependent (don’t know how; pca is an statistic analysis method，找出最關聯的principle component,然後我們只用這些component來進行modeling而將其他捨棄，簡化了dataset，例如workbook裡就發現前5個element佔據99%的variance,只是在用其他model train時只用principal compoenet和全用有區別罷了)

然後用一堆模型去fit data, 畫graph去determine which is better

(模型包括 svm, knn, random forest, adaboost, extratree, gradientboost, logistic regression) (don’t know how some of them work, check later)

最後得出random forest的結果最好，tune random forest (don’t know what happening)

基於rfc結果畫圖（add later）