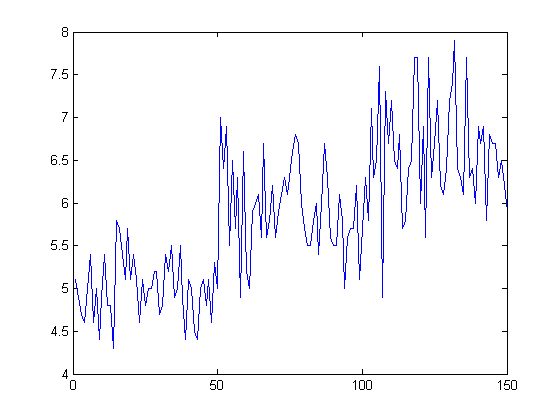
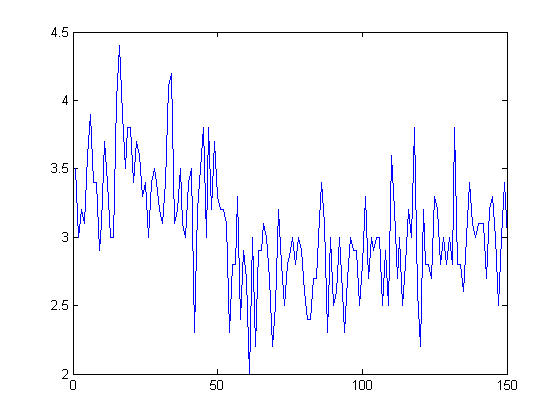
Para este ejercicio con kmeans utilizado en matlab tenemos de ejemplo una base de datos en formato csv que consiste en la clasificación de tres tipos de flores las cuales son ‘Iris Setosa’, ’iris versicolor’, ‘iris virginica’ interpretados como 1,2,3 respectivamente para este ejercicio, los atributos son la longitud y largo del tallo (tomándose ambas distancias como atributos) y longitud y largo del pétalo(también interpretados como atributos separados) como total tenemos 150 registros con 4 atributos en total los datos inicialmente están clasificados de manera equitativa siendo 50 datos para cada clase.

**Atributo 1:**



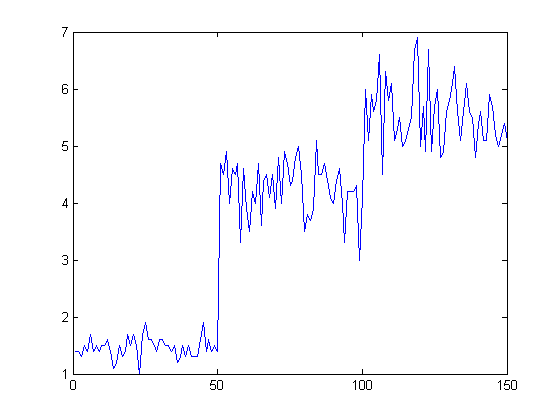
Aquí podemos apreciar el comportamiento de los 150 registros con su atributo 1 que es la longitud del tallo.

**Atributo 2.**



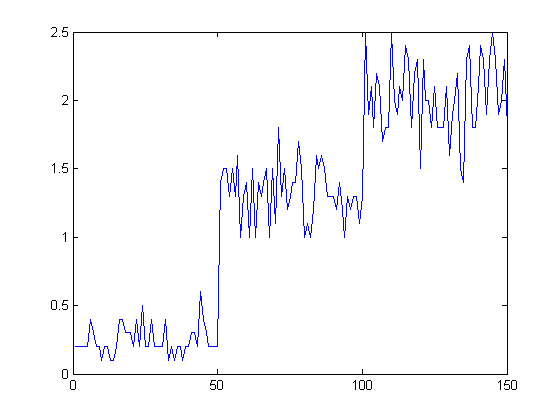
Ancho del tallo

**Atributo 3.**



Longitud del pétalo.

**Atributo 4.**



Ancho del pétalo.

Notamos que estos datos están fuertemente agrupados en intervalos de 50 para cada atributo, para de primera impresión notar un comportamiento en los datos tendiéramos que buscar una manera de graficar cada atributo de cada registro en el espacio donde se tengan 5 coordenadas, a pesar de eso con las gráficas mostradas podemos percibir un comportamiento.

En el proceso de kmeans se basa en la cercanía que tienen los datos con centros escogidos aleatoriamente o mejor dicho seleccionado en base a un proceso donde tomando los registros, en cada atributo se toma el mínimo y máximo y dentro de este rango se genera una dirección para ese atributo en especifico y procedimiento se repite para cada atributo ahora si vemos los grupos veremos que las direcciones de los centroides están lejanos de los grupos así que tenemos que hacer varias iteraciones dentro de un bucle donde tomando un promedio de las asignaciones modificamos las direcciones de centroide con esto ajustamos los grupos de manera adecuada.

Función para generar los primeros centroides aleatorios.

function [clusters]=CrearCentroides(data,k)

dimensiones = size(data); %tomamos las dimensiones de la tabla

minimo=min(data);%esto nos devuelve un vector con los valores minimos de cada columna

maximo=max(data);%valores maximos de cada columna

clusters=[k,dimensiones(2)+1];%inicializamos la tabla de clusters

for v=1:k%este for recorre renglon por renglon de los clusters pensados

for c=1:dimensiones(2)%este for recorre columna por columna

numeroAleatorio=minimo(c)+rand(1,1)\*(maximo(c)-minimo(c));%producimos un numero aleatorio entre el maximo y minimo de cada columna

clusters(v,c)=numeroAleatorio;%agregamos este valor a la matriz de clusters(dentro de este for anidado se agregan a cada celda de cada renglon(o sea a las columnas de cada renglon)

end

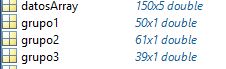
clusters(v,dimensiones(2)+1)=v;

end

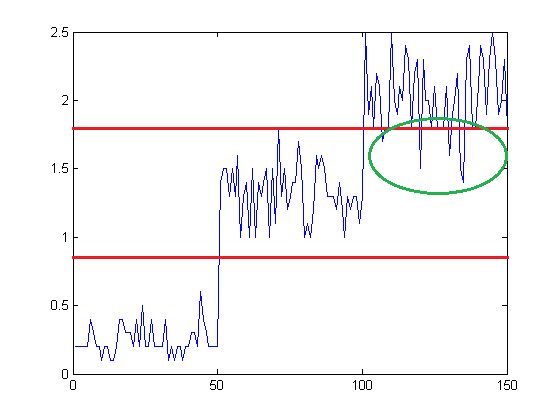
end

debido a esto en nuestro script se debe ejecutar varias veces ya que al empezar los centroides aleatorios esto cambia el resultado de clasificación, el único comportamiento que parece repetirse son las direcciones final de los centroides ajustados, siempre convergen a los mismos valores.

En el resultado más cercano al deseado podemos ver que los primeros 50 registros se logra hacer una asignación correcta pero entre los dos grupos de 2 y 3 se encuentran mezclados a pesar de eso se encuentran predominados por su correspondido grupo.



Este comportamiento podría está justificado por la siguiente gráfica, en la cual podemos ver que parte de los datos que se encuentran junto al grupo 3 comparte coordenadas con los del grupo dos obsérvese los puntos en el circulo verde.



En otras dos graficas se presente esto mismo así que eso podría explicar el por qué se mezclan los datos.