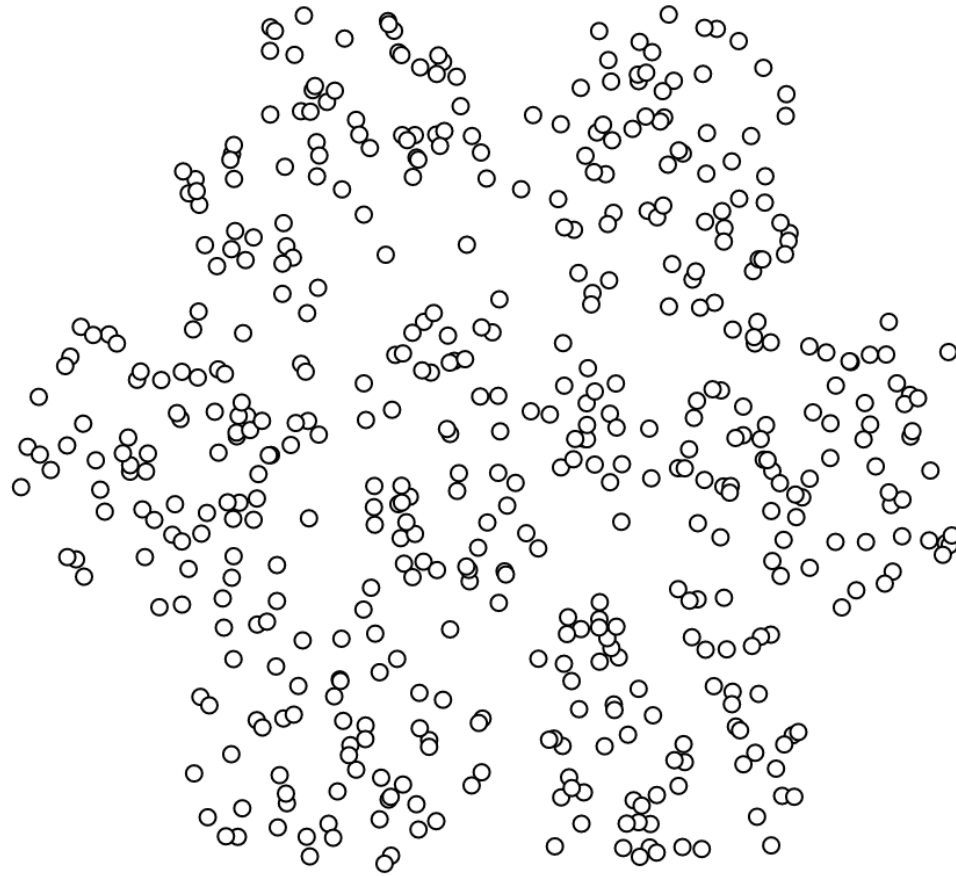


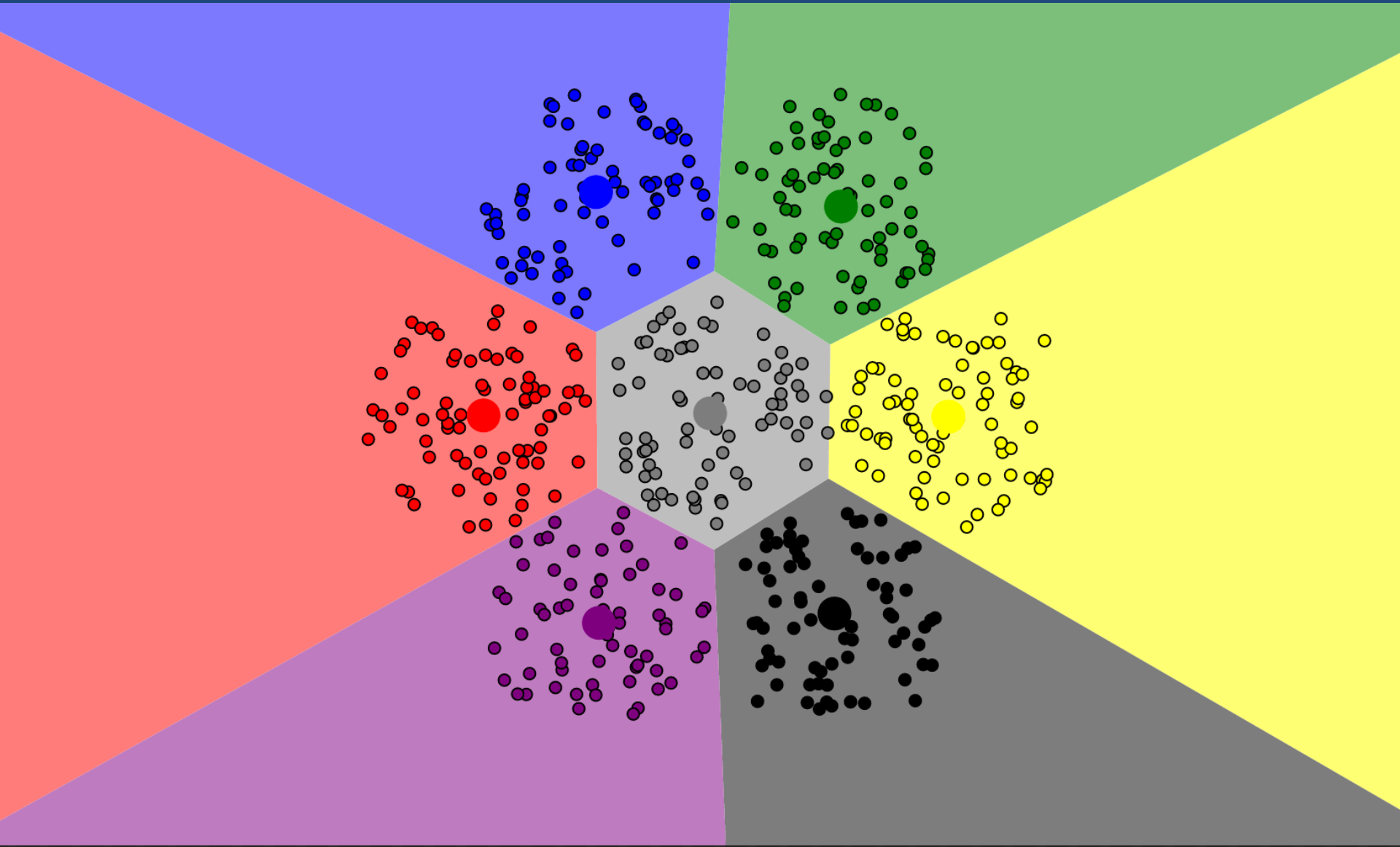
k-means clustering

Method to automatically separate data sets into distinct groups.

Clustering example



Clustering example

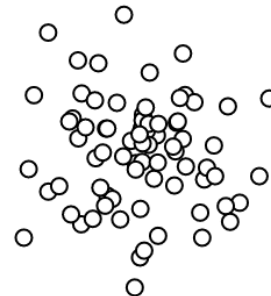
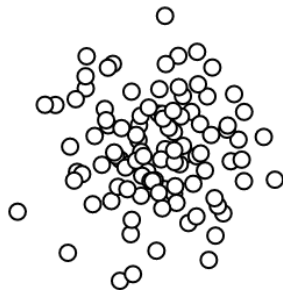
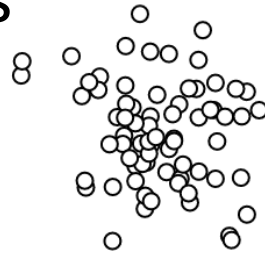


k -means clustering algorithm

1. Start with k randomly chosen means
2. Color data points by the shortest distance to any mean
3. Move means to centroid position of each group of points
4. Repeat from step 2 until convergence

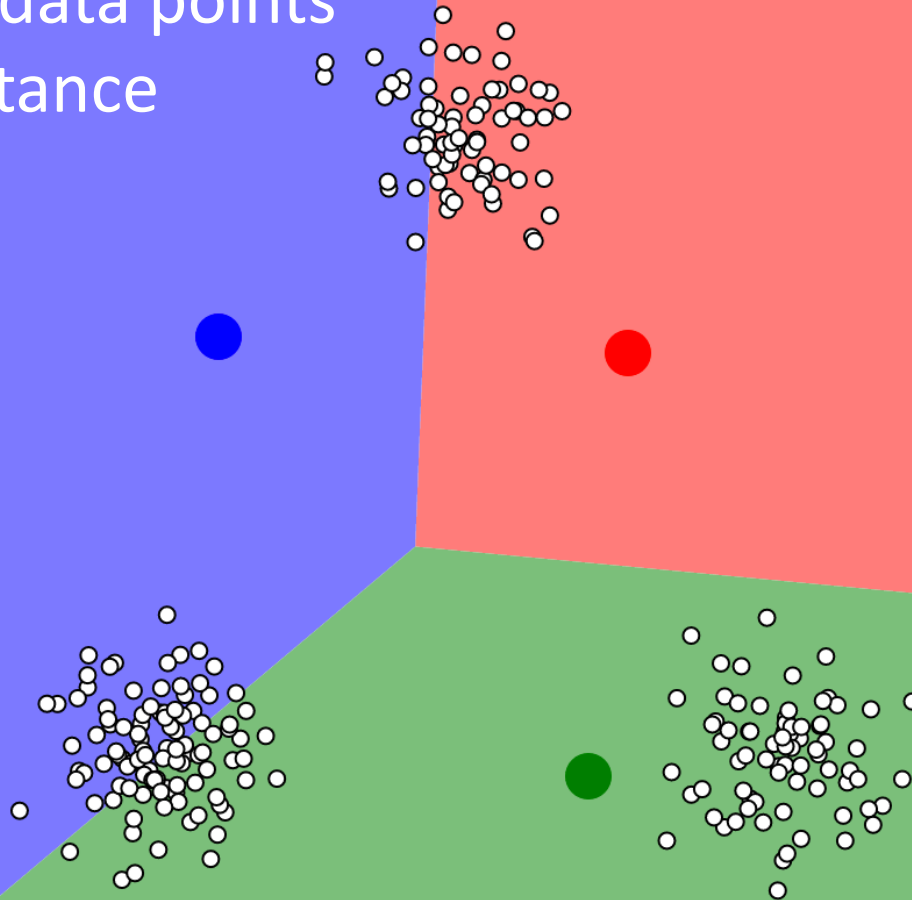
Algorithm example ($k = 3$)

Step 1: Choose 3 means
at random



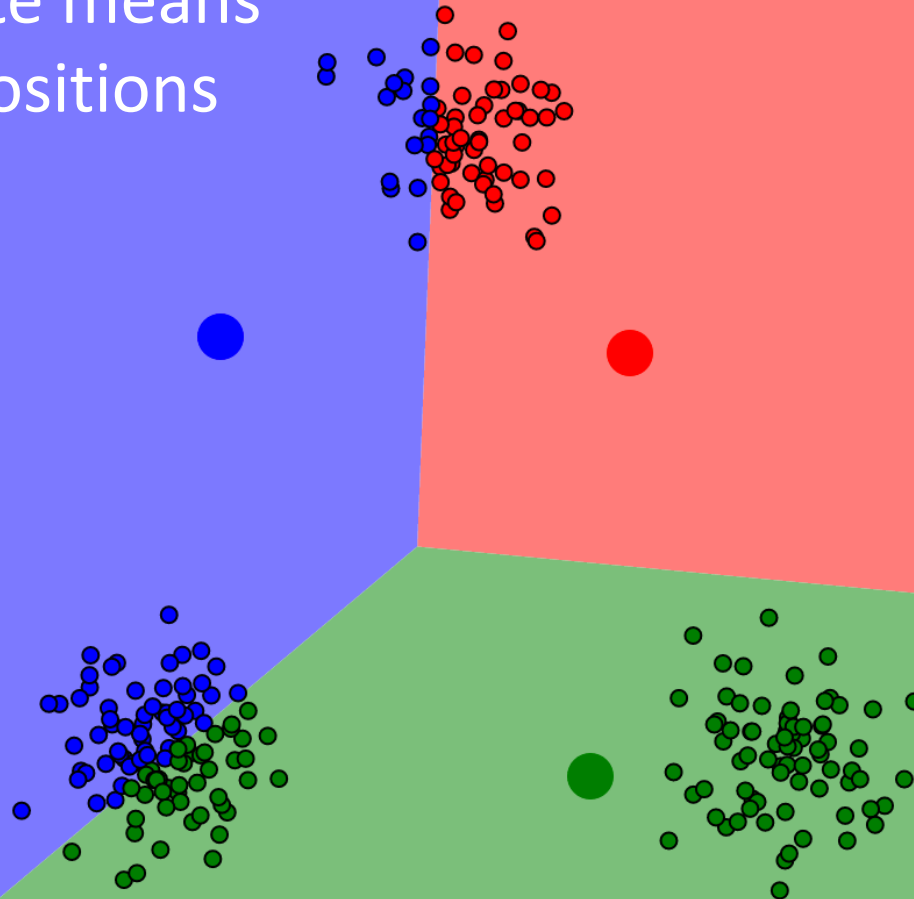
Algorithm example ($k = 3$)

Step 2: Color data points
by closest distance
to any mean



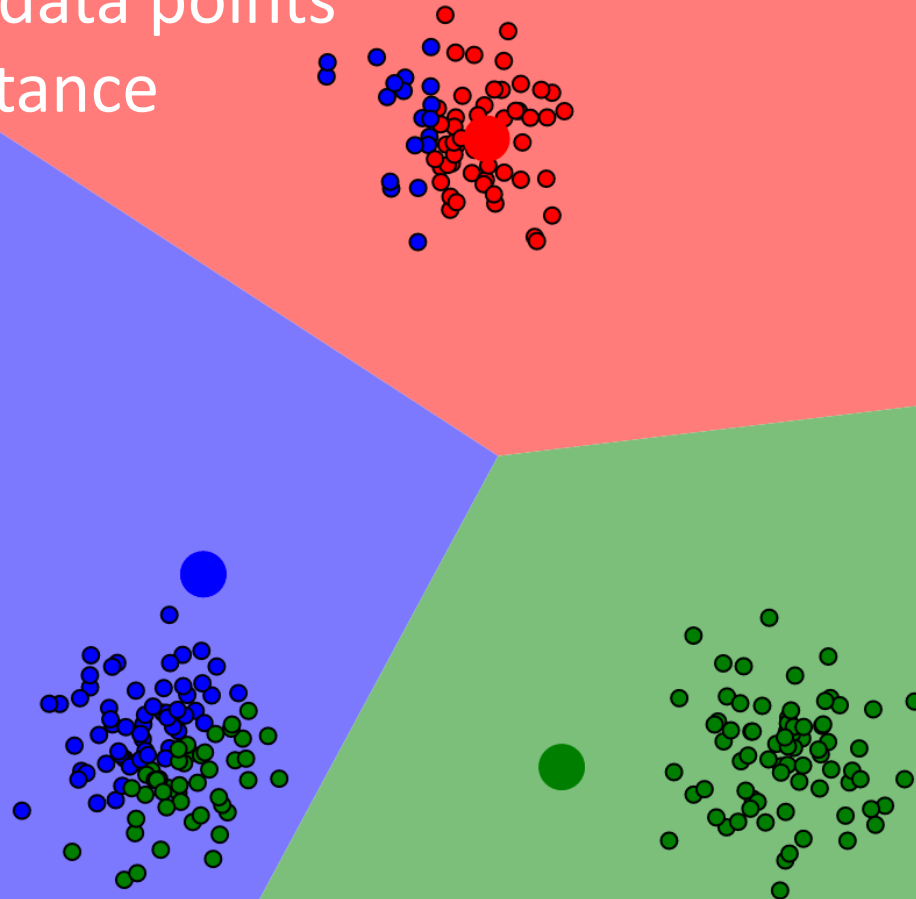
Algorithm example ($k = 3$)

Step 3: Update means to centroid positions



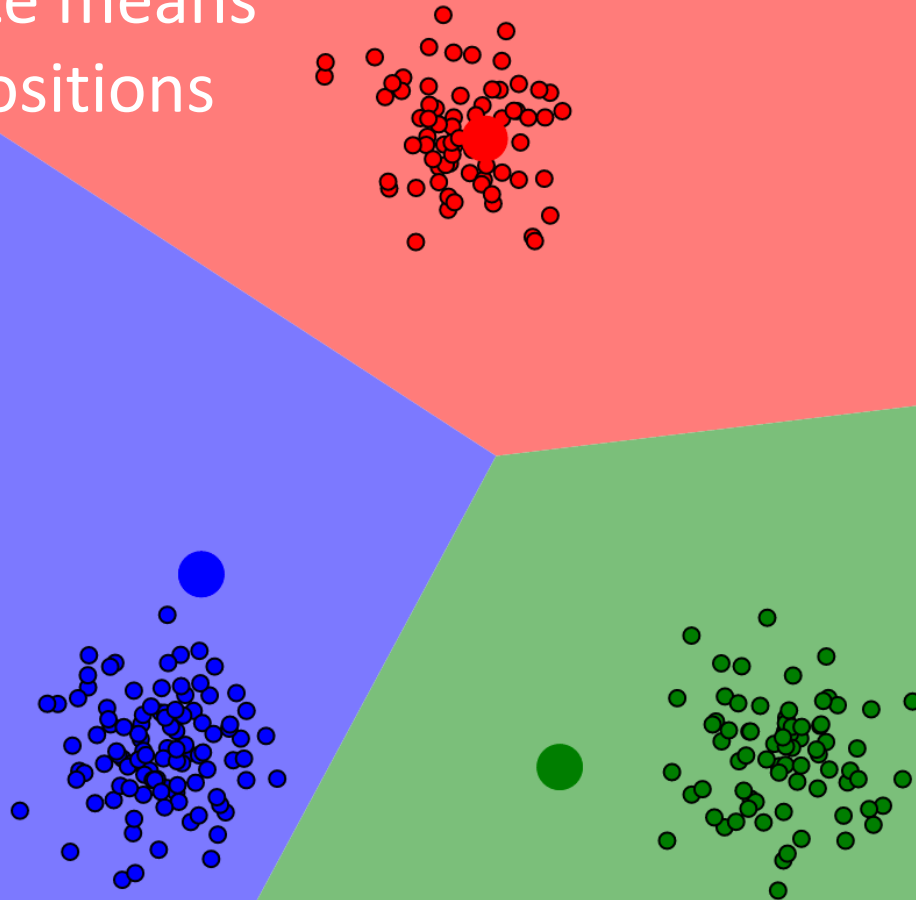
Algorithm example ($k = 3$)

Step 2: Color data points
by closest distance
to any mean



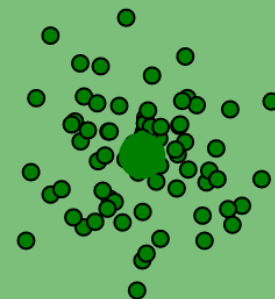
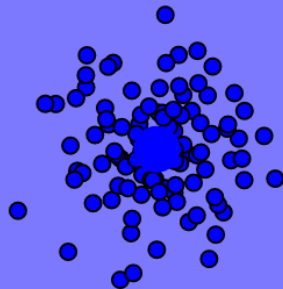
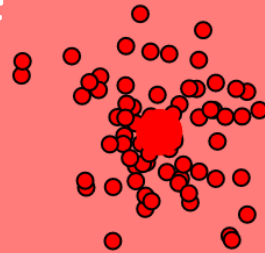
Algorithm example ($k = 3$)

Step 3: Update means
to centroid positions



Algorithm example ($k = 3$)

Stop: no further change occurs



Now try it yourself

<http://www.naftaliharris.com/blog/visualizing-k-means-clustering/>

k-means in R

(example: iris data set)

```
iris %>% select(-Species) %>% # remove Species column
  kmeans(centers=3) ->        # do k-means clustering
                               # with 3 centers
km                             # store result as "km"
```

k-means in R (example: iris data set)

> km

K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	6.850000	3.073684	5.742105	2.071053
2	5.901613	2.748387	4.393548	1.433871
3	5.006000	3.428000	1.462000	0.246000

Clustering vector:

[1]	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3					
[38]	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2				
[75]	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	1	1	1	1	2	1	1	1	1	
[112]	1	1	2	2	1	1	1	1	2	1	2	1	2	1	1	2	2	1	1	1	1	1	2	1	1	1	1	2	1	1	1	2	1	1	2	1
[149]	1	2																																		

Within cluster sum of squares by cluster:

```
[1] 23.87947 39.82097 15.15100
      (between SS / total SS = 88.4 %)
```

k-means in R (example: iris data set)

> km

K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	6.850000	3.073684	5.742105	2.071053
2	5.901613	2.748387	4.393548	1.433871
3	5.006000	3.428000	1.462000	0.246000

Cluster means:
the location of the
final centroids

Clustering vector:

```
[1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
[38] 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
[75] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1  
[112] 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1  
[149] 1 2
```

Within cluster sum of squares by cluster:

[1] 23.87947 39.82097 15.15100
(between SS / total SS = 88.4 %)

k-means in R (example: iris data set)

> km

K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	6.850000	3.073684	5.742105	2.071053
2	5.901613	2.748387	4.393548	1.433871
3	5.006000	3.428000	1.462000	0.246000

Clustering vector:

[1]	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3		
[38]	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
[75]	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	1	1	1	1	2	
[112]	1	1	2	2	1	1	1	1	2	1	2	1	2	1	1	2	2	1	1	1	1	2	1	1	1	1	2	1	1	1	2
[149]	1	2																													

Clustering vector: provides the cluster to which each

Clustering vector: provides the cluster to which each observation belongs

Within cluster sum of squares by cluster:

```
[1] 23.87947 39.82097 15.15100
```

(between SS / total SS = 88.4 %)

k-means in R (example: iris data set)

> km

K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	6.850000	3.073684	5.742105	2.071053
2	5.901613	2.748387	4.393548	1.433871
3	5.006000	3.428000	1.462000	0.246000

Clustering vector:

[illegible]

Within cluster sum of squares: measures quality of

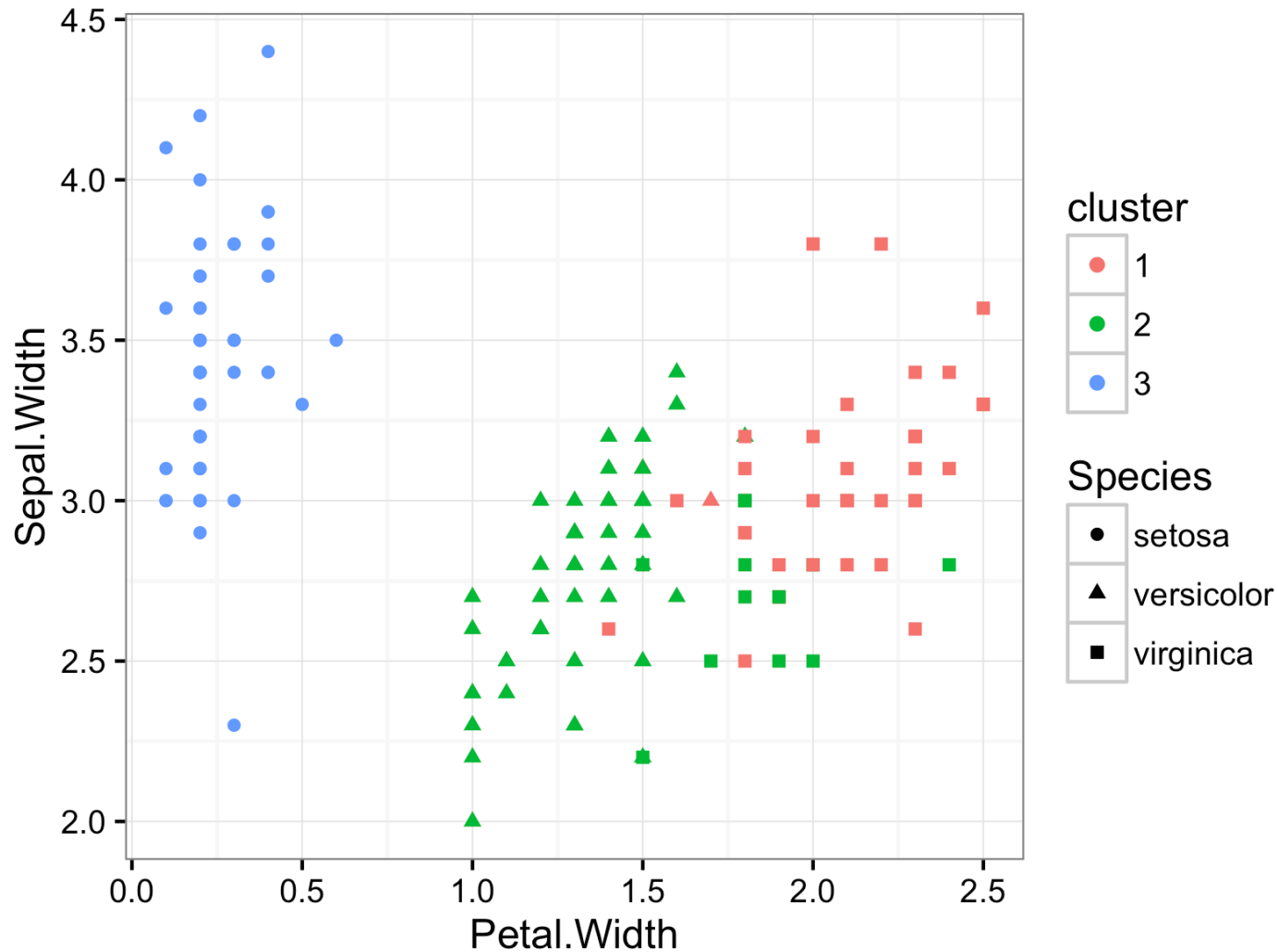
Within cluster sum of squares by cluster:

```
[1] 23.87947 39.82097 15.15100
```

(between_SS / total_SS = 88.4 %)

the clustering (lower is better)

The clusters mostly but not exactly recapitulate the species assignments



How do we determine the right number of means k ?

- Many different methods, see e.g.:
<http://stackoverflow.com/a/15376462/4975218>
- Simplest: plot within-sum-of-squares against k

A bend in within-sum-of-squares indicates the ideal number of clusters

