Demonstration of Kmeans after PCA

For the iris data:

```
data(iris)
iris_no_lab = iris[,1:4]
iris_feat = scale(iris_no_lab, center = TRUE, scale = TRUE)
iris_lab = iris[,5]
```

If you use SVD, do not forget to center !!!!

We can plot the data in the space of the first two score vectors:

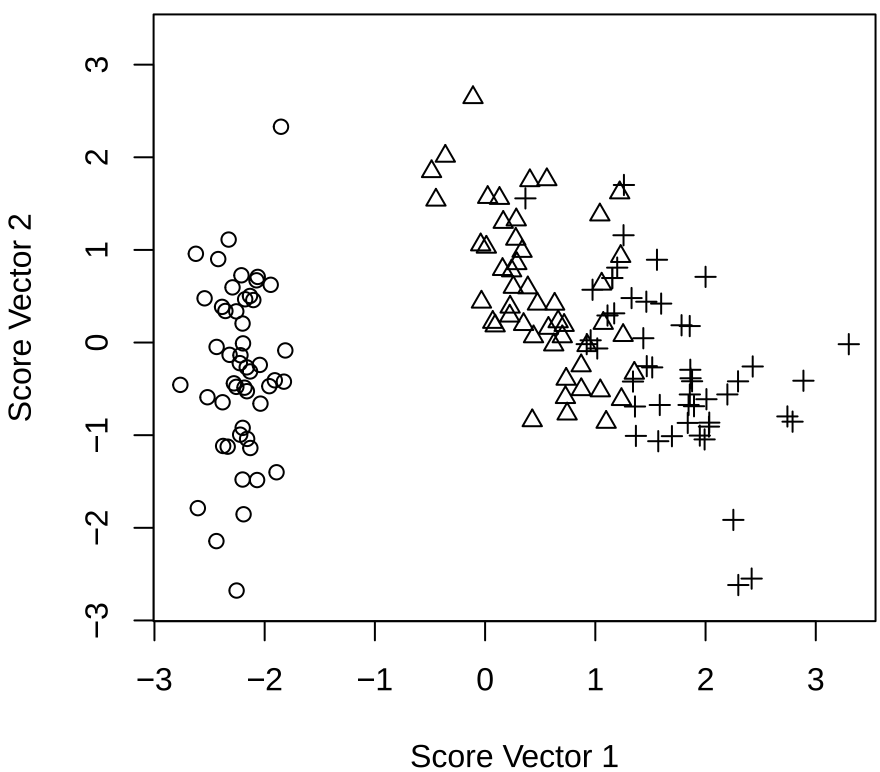
```
iris_svd = svd(iris_feat)
```

Recall that the u vectors are normalized. To get the score vectors, we need to multiple by d

```
score_vectors =iris_svd$u %*% diag(iris_svd$d)
```

Plotting in score vector space maintains the distances. I am deliberately using the same range for both axes, and making sure the plot is square (setting fig.width, fig.height in the markdown block).

```
use_range = c(min (score_vectors), max(score_vectors))
plot(score_vectors[,1],score_vectors[,2], pch = as.numeric(iris_lab),
     xlim = use_range,ylim = use_range,
     xlab = "Score Vector 1", ylab = "Score Vector 2")
```



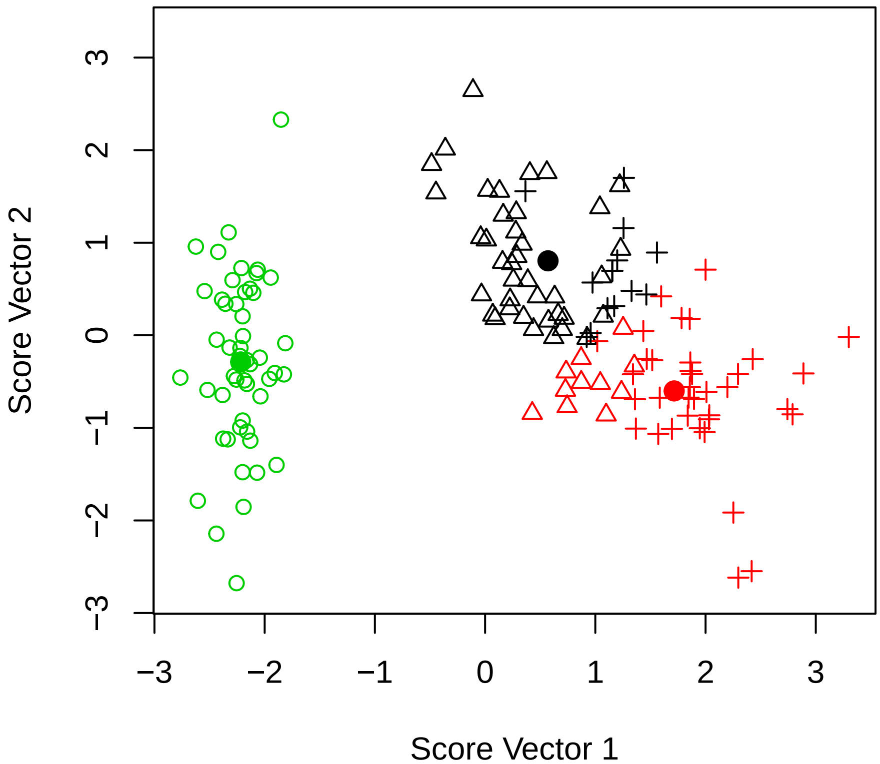
Regular K-means, displayed in PCA space

We can run k-means in the original space and show the result in the new space.

```
set.seed(100)
# Don't forget to set.seed, because starting point is randomizedd.
```

```
cl<- kmeans(iris_feat,3)
plot(score_vectors[,1],score_vectors[,2], pch = as.numeric(iris_lab),
     xlim = use_range,ylim = use_range,
     xlab = "Score Vector 1", ylab = "Score Vector 2", col = cl$cluster)

# Draw centers in the displayed space
pca_centers =cl$centers %*% iris_svd$v[,1:2]
points(pca_centers, col = 1:3, pch = 20, cex = 2)
```



K-means after PCA

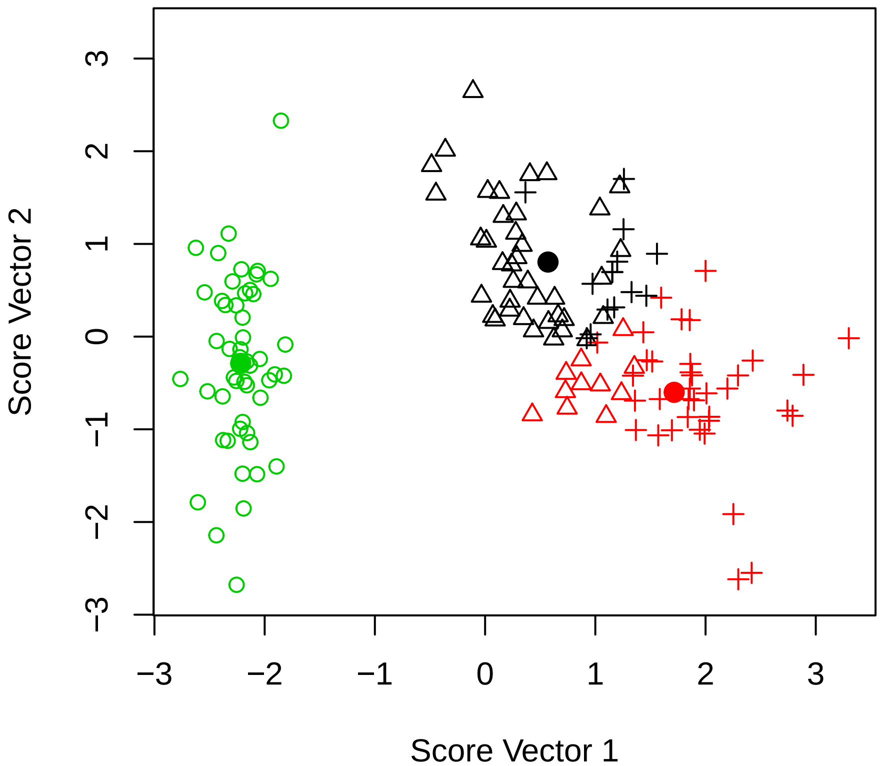
We can run k-means in the new leading-pc space. The data set we're using is either the score matrix, or u vectors in svd.

We'll use the first $q = 3$ PCs to illustrate that we don't have to run on $q = 2$.

```
set.seed(101)

first_scores = score_vectors[,1:3]
cl_pcs<- kmeans(first_scores,3)
plot(score_vectors[,1],score_vectors[,2], pch = as.numeric(iris_lab),
     xlim = use_range,ylim = use_range,
     xlab = "Score Vector 1", ylab = "Score Vector 2", col = cl_pcs$cluster)

pca_centers = cl_pcs$centers
points(pca_centers[,1:2], col = 1:3, pch = 20, cex = 2)
```



Running in the new space has several effects:

- Faster, becuase each distance computation in $q < p$ dimensions
- Sometimes reduces noise making clusters will be easier to find.
- K-means requires Euclidean space, so sometimes need to produce embedding first (MDS).

Why PCA may reduce noise?

Suppose the data comes from several clusters. Call Δ a random variable saying which cluster the example came from.

I remind you that the covariance of a random vector $\mathbf{x}$ can be composed into:

$$cov(\mathbf{x}) = cov(E[\mathbf{x}|\Delta]) + E[cov(\mathbf{x}|\Delta)].$$

- $cov(E[\mathbf{x}|\Delta])$ represents the covariance of the cluster centers weighted by points per cluster.
- $E[cov(\mathbf{x}|\Delta)]$ represents the "average" covariance around the centers of the clusters.

The large eigenvalues tend to follow the first arguument $cov(E())$ because:

- The first argument gives same direction for all examples in the same cluster.
- The second argument gives different direction for each example.