Coding Assignment

Xiaoyu Qiao

3/26/2019

```
library(mclust)
library(cluster)
library(ggplot2)
library(ggfortify)
library(cowplot)
```

We firstly write a function `getA1` for simulating high-dimensional data (p=1000) with three groups of observations where the number of observations is n=100:

```
getA1 <- function(){
n_rows = 1000
n_cols = 100

k=3
x_mus = c(0,5,5)
x_sds = c(1,0.1,1)
y_mus = c(5,5,0)
y_sds = c(1,0.1,1)
prop1 = c(0.3,0.5,0.2)

comp1 <- sample(seq_len(k), prob=prop1, size=n_cols, replace=TRUE)
samples1 <- cbind(rnorm(n=n_cols, mean=x_mus[comp1],sd=x_sds[comp1]),
                  rnorm(n=n_cols, mean=y_mus[comp1],sd=y_sds[comp1]))

proj <- matrix(rnorm(n_rows* n_cols), nrow=n_rows, ncol=2)
A1 <- samples1 %*% t(proj)
A1 <- A1 + rnorm(n_rows* n_cols)
return (list("data" = A1, "labels" = comp1))
}
```

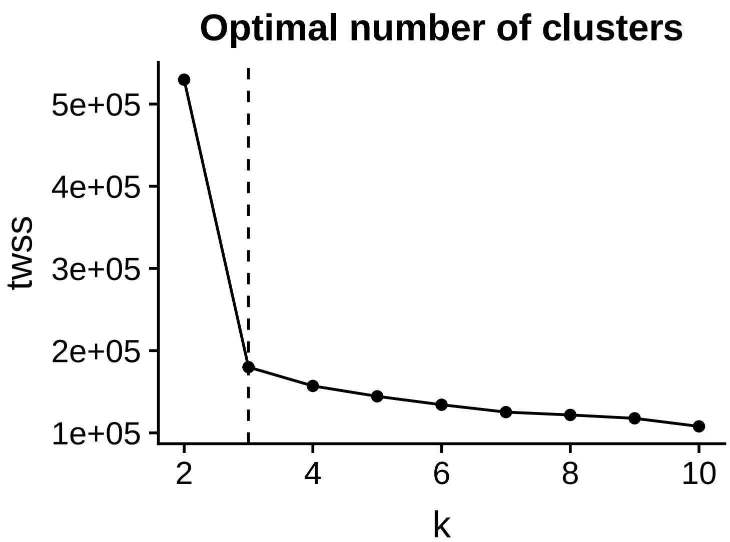
We firstly take a look at a single run, and find out the optimal number of clustering; we plot total within groups sum of squares against values of k, we pick k to be the elbow point, which corresponding to $k = 3$.

```
set.seed(100)

sample = getA1()
A1=sample$data

#function for calculating total within groups sum of squares
twss <- function(fit){
return(fit$tot.withinss)
}

result = data.frame(k=c(2:10),twss=sapply(2:10,function(k){twss(kmeans(A1, k,nstart = 25))}))
ggplot(data=result, aes(x=k, y=twss)) + geom_line()+geom_point()+ geom_vline(xintercept = 3,linetype = 
```



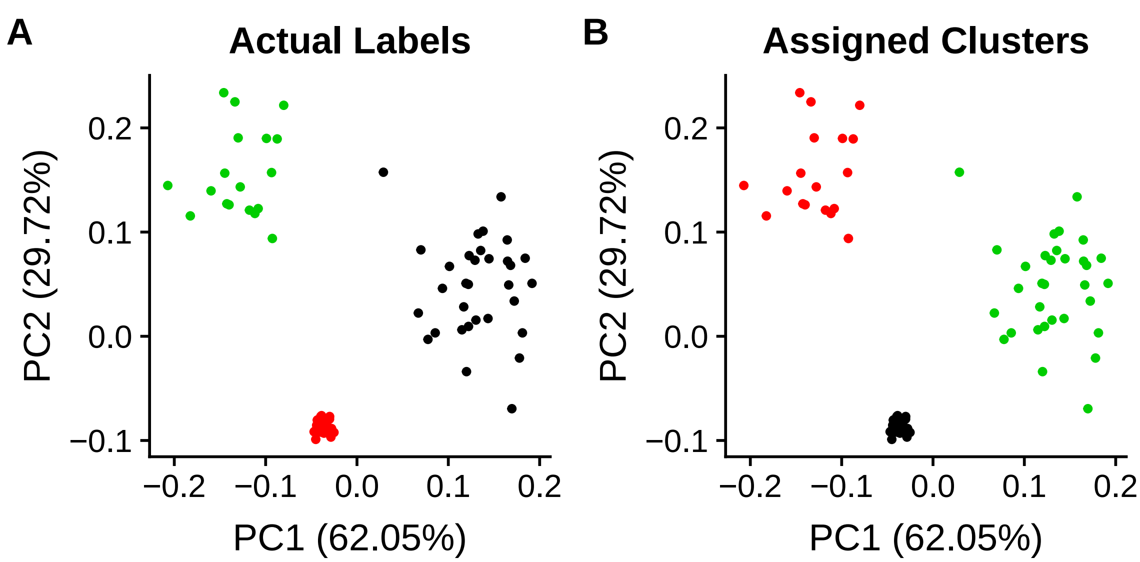

```
k.opt=3
```

To visualizing our cluster result on the sample data, we plot the first two principal components with both coloring on its original labels and on its k-means clustering results.

```
data <- sample$data
labels <- sample$labels


a<-autoplot(prcomp(data),size=1,colour = labels,main = "Actual Labels")
b<-autoplot(prcomp(data),size=1,colour = kmeans(data, k.opt,nstart = 25)$cluster,main ="Assigned Cluste

plot_grid(a,b, labels = "AUTO")
```



We see that using K-means gives a good clustering result.

Repeat the process 100 times

Now, we generate simulated high-dimensional data and perform K-means 100 times; and to access the accuracy, we calculate the adjusted rand index and the total within clusters sum of squares for each run:

```
metrics <- data.frame(ARI=numeric(0),WSS=numeric(0))

for (i in 1:100) {
result = getA1()
A1 = result$data
lbs = result$labels
KM = kmeans(A1, k.opt,nstart = 25)
clusters <- KM$cluster
new <- data.frame(adjustedRandIndex(clusters, lbs), twss(KM))
names(new)<-c("ARI","WSS")
metrics <- rbind(metrics,new)
}
metrics
```

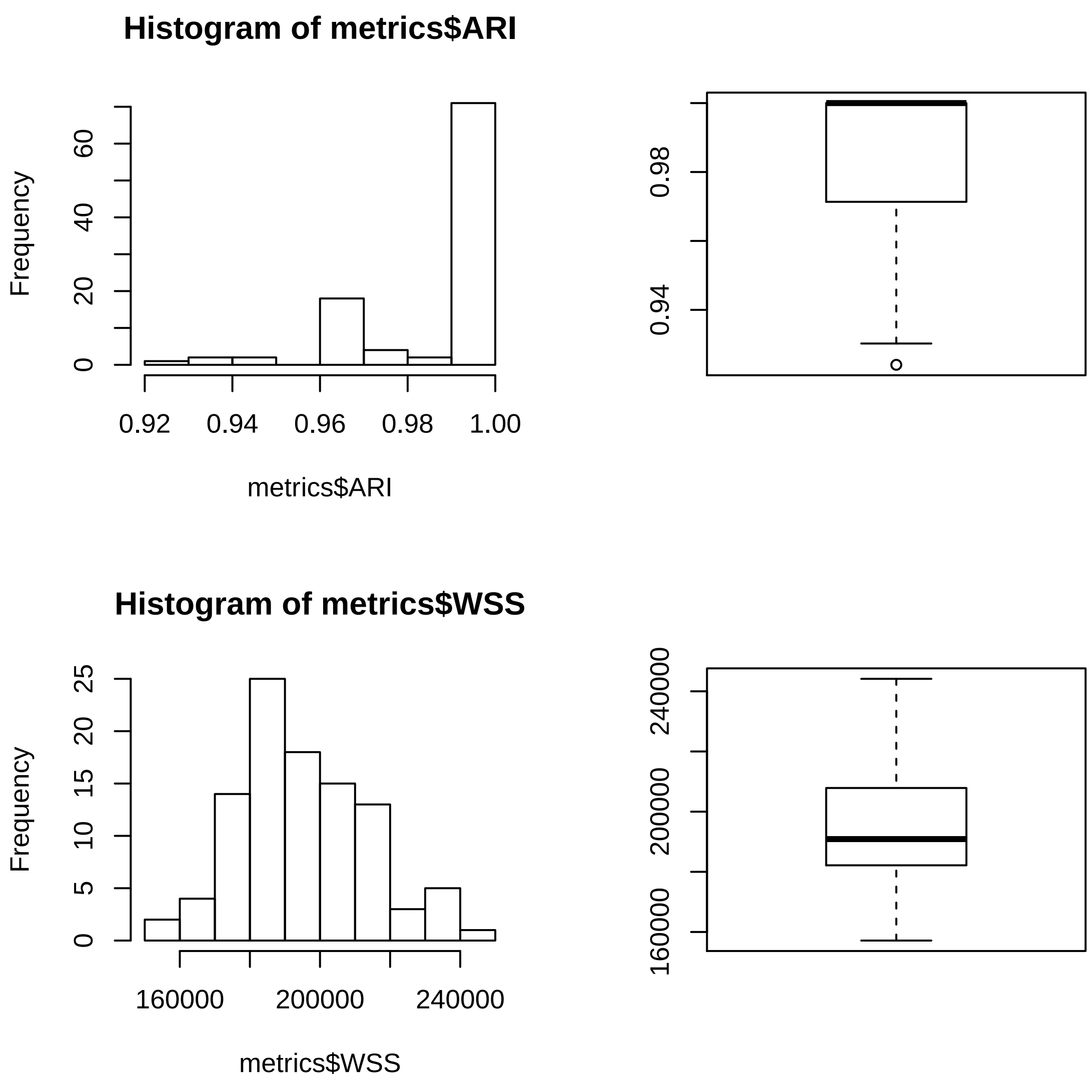
```
##        ARI    WSS
## 1   0.9688 192610
## 2   1.0000 200326
## 3   1.0000 199096
## 4   1.0000 184866
## 5   0.9653 190721
## 6   1.0000 179817
## 7   1.0000 199952
## 8   1.0000 175358
## 9   1.0000 211081
## 10  1.0000 231726
## 11  1.0000 218854
## 12  1.0000 204280
## 13  1.0000 175716
## 14  1.0000 189611
## 15  0.9435 189547
## 16  0.9699 206312
## 17  1.0000 158101
## 18  1.0000 171117
## 19  1.0000 201270
## 20  1.0000 190339
## 21  0.9712 176934
## 22  1.0000 157150
## 23  1.0000 232717
## 24  1.0000 189256
## 25  1.0000 191737
## 26  0.9656 186921
## 27  1.0000 175415
## 28  1.0000 190627
## 29  0.9302 204489
## 30  0.9811 165756
## 31  1.0000 197220
## 32  1.0000 197459
## 33  0.9241 203076
## 34  0.9657 218901
```

```
## 35  1.0000 210271
## 36  1.0000 206299
## 37  1.0000 184596
## 38  1.0000 188136
## 39  1.0000 190200
## 40  1.0000 209627
## 41  1.0000 221193
## 42  1.0000 184069
## 43  1.0000 179731
## 44  1.0000 177863
## 45  0.9656 213521
## 46  1.0000 179927
## 47  1.0000 217364
## 48  0.9413 210453
## 49  0.9699 179977
## 50  1.0000 163896
## 51  1.0000 238984
## 52  1.0000 196650
## 53  1.0000 184743
## 54  1.0000 188507
## 55  1.0000 197344
## 56  1.0000 194441
## 57  1.0000 213726
## 58  1.0000 209300
## 59  1.0000 238490
## 60  1.0000 181363
## 61  1.0000 213458
## 62  1.0000 185399
## 63  0.9715 184617
## 64  0.9803 192250
## 65  0.9666 218573
## 66  1.0000 183029
## 67  0.9676 190611
## 68  1.0000 191031
## 69  1.0000 220026
## 70  1.0000 218826
## 71  0.9670 201328
## 72  1.0000 182010
## 73  1.0000 186045
## 74  1.0000 200292
## 75  1.0000 210329
## 76  0.9689 181691
## 77  1.0000 200600
## 78  1.0000 221717
## 79  1.0000 192631
## 80  1.0000 218823
## 81  1.0000 181624
## 82  0.9707 244147
## 83  0.9698 178641
## 84  1.0000 182314
## 85  0.9655 235531
## 86  1.0000 209315
## 87  1.0000 169641
## 88  0.9661 188461
```

```
## 89   0.9358 177095
## 90   0.9697 201747
## 91   1.0000 181809
## 92   1.0000 191919
## 93   0.9719 206433
## 94   0.9659 188033
## 95   1.0000 187841
## 96   1.0000 160782
## 97   0.9682 183169
## 98   0.9664 175580
## 99   1.0000 179349
## 100 1.0000 187211
```

Now we use boxplot and histogram to view the result of adjusted rand index and the total within clusters sum of squares:

```r
par(mfrow=c(2,2))
hist(metrics$ARI)
boxplot(metrics$ARI)
hist(metrics$WSS)
boxplot(metrics$WSS)
```



By the result of adjusted rand index, we know our K-means model has great accuracy.