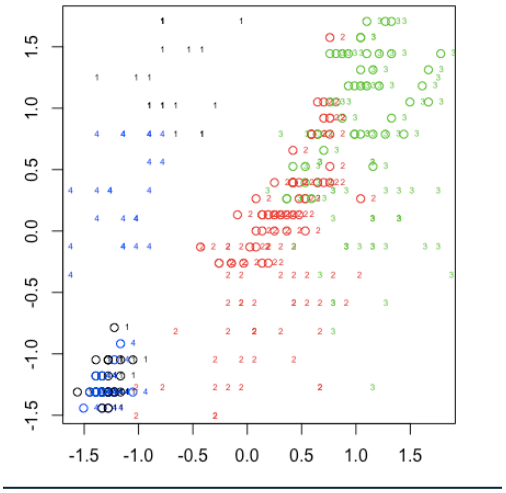
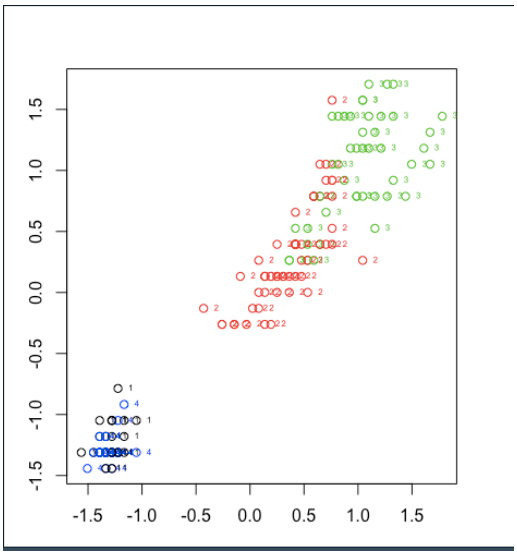
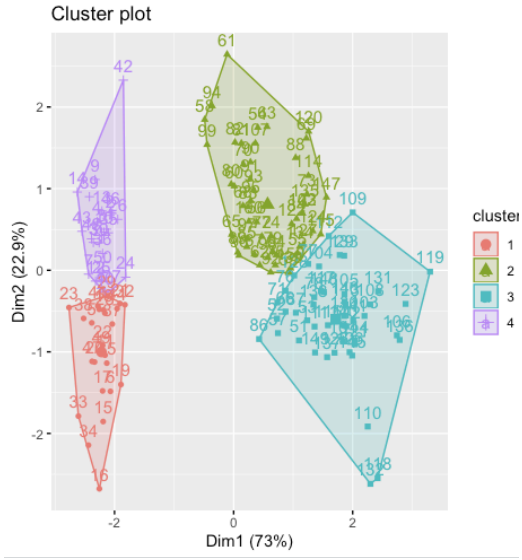
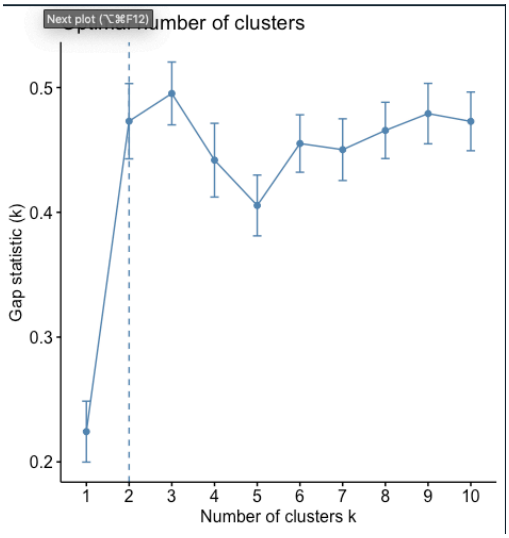
Swati Chauhan
MSTM ADV

Find out the best "k" for k means, and use the best "k" to run k-means clustering. Plot 2-d plots with all pairs of features, use the clustering results as color.

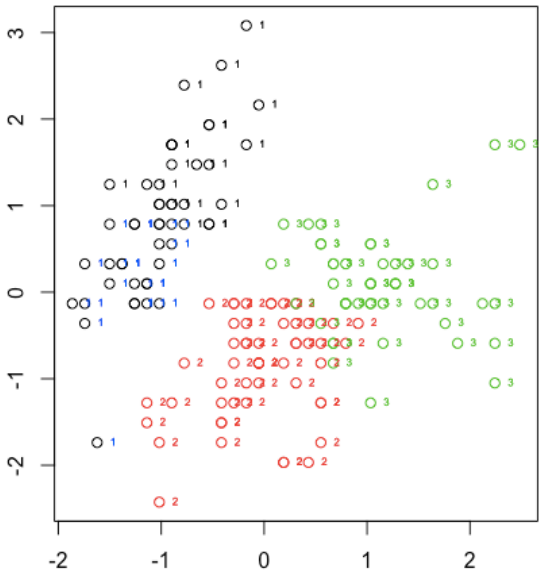


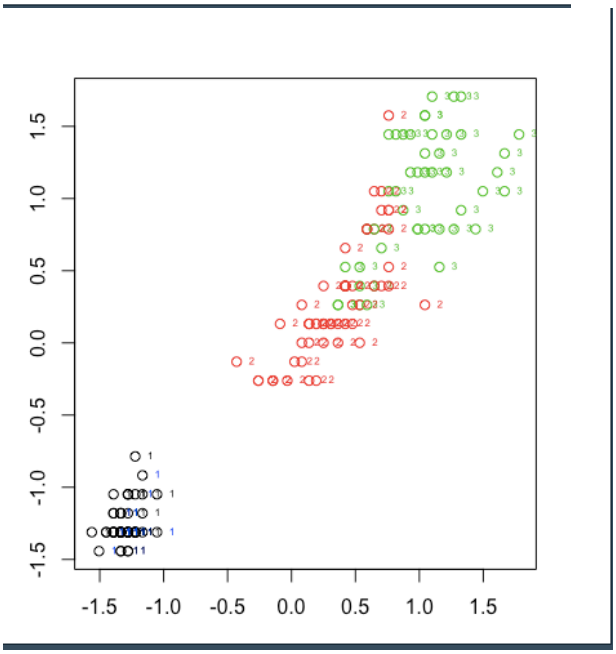






Swati Chauhan
MSTM ADV



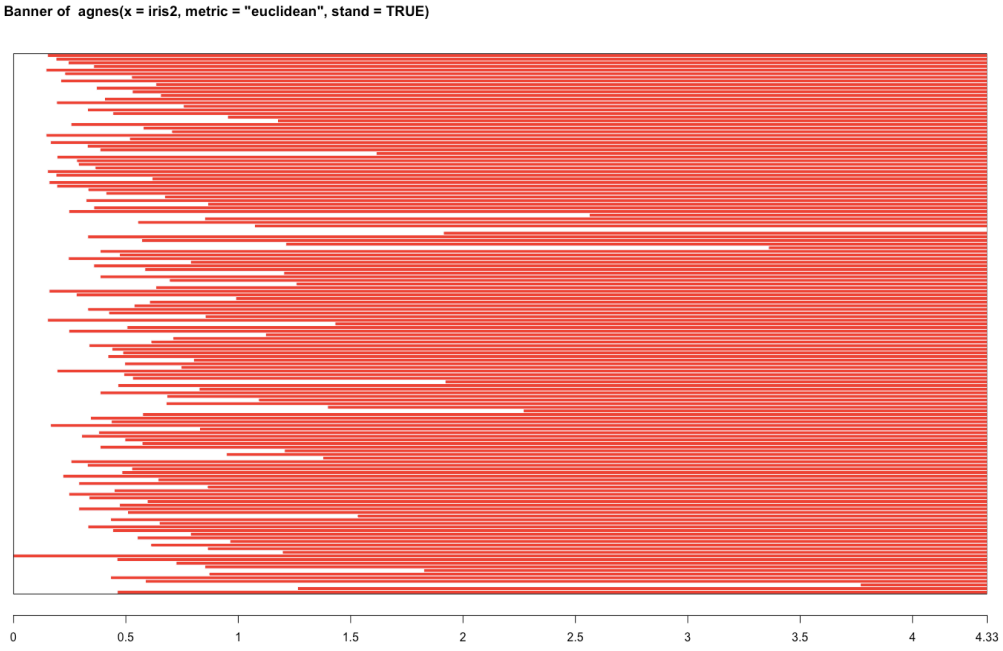


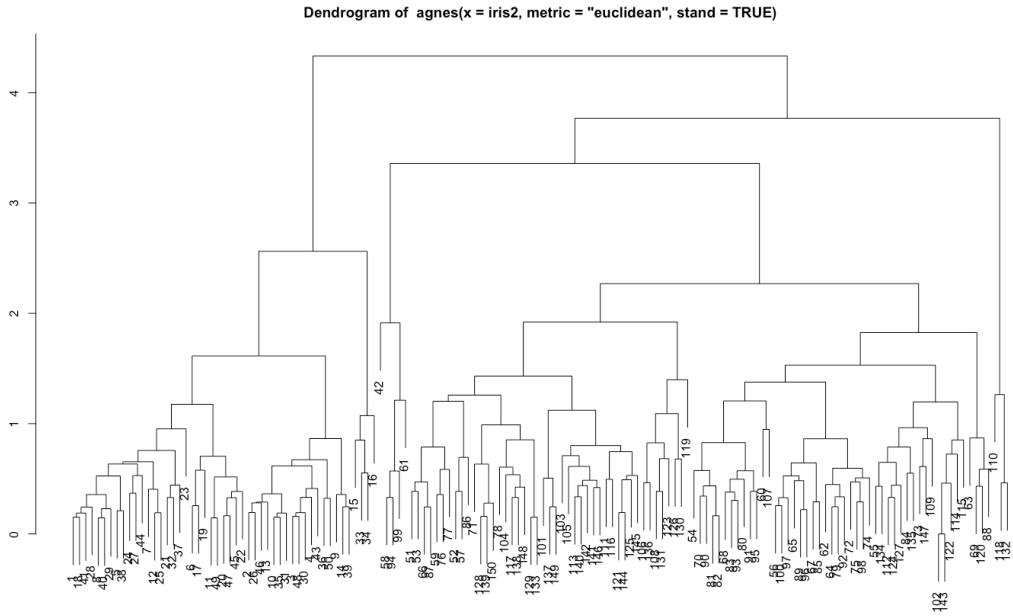
Swati Chauhan
MSTM ADV

Use agn() to run hierarchical clustering on iris dataset.

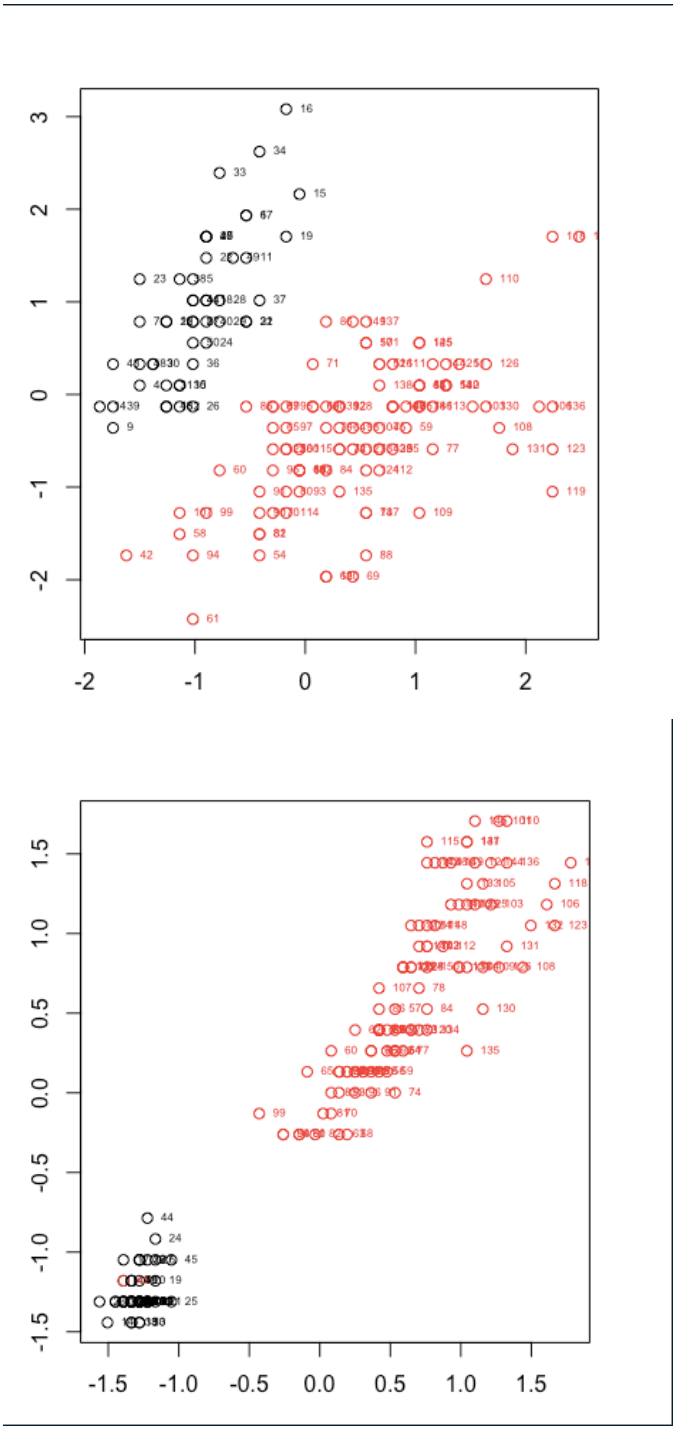
Present the dendrogram

Plot 2-d plots with all pairs of features, use the clustering results when we preserve 2 clusters as color.



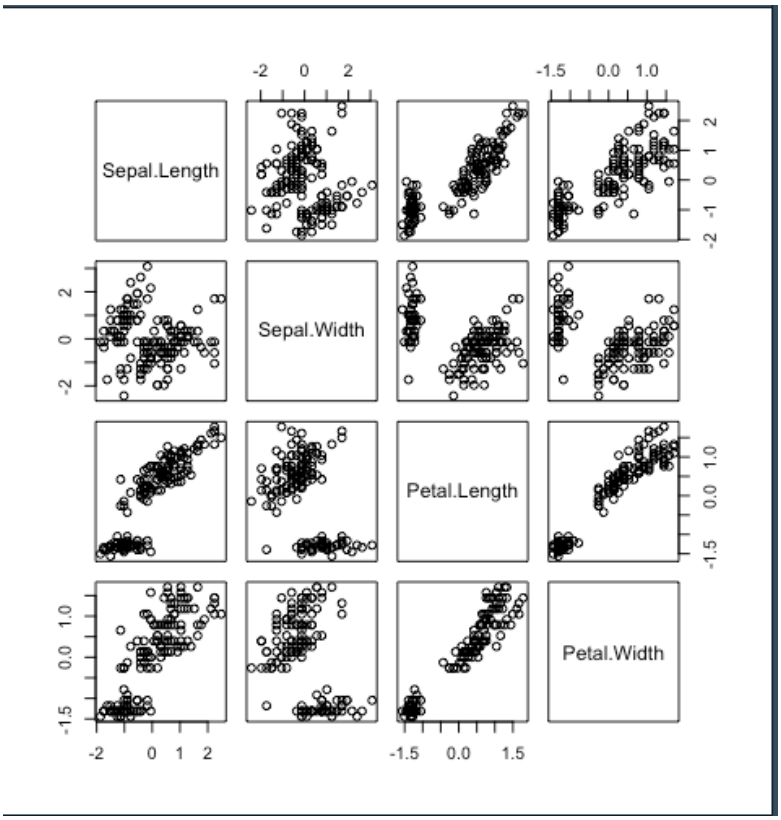


Swati Chauhan
MSTM ADV



Swati Chauhan
MSTM ADV

Use dbscan to cluster the dataset. Tune the parameters so that there will be 2-5 clusters. Plot 2-d plots with all pairs of features, use the clustering results as color.



Swati Chauhan
MSTM ADV

```
data("iris")
View(iris)
temp=scale (iris,center=TRUE, scale =TRUE)
iris2<-data.frame(scale (iris[-5], center=TRUE, scale =TRUE))
k4 = kmeans (iris2, centers=4, nstart = 25)
k4
result = cbind (iris2 , k4$cluster)
View(result)
plot(result$Sepal.Length , result$Sepal.Width,col = k4$cluster)
text(result$Sepal.Length , result$Sepal.Width,k4$cluster, cex=0.5, col=k4$cluster, pos=4)
plot(result$Petal.Length , result$Petal.Width,col = k4$cluster)
text(result$Petal.Length , result$Petal.Width,k4$cluster, cex=0.5, col=k4$cluster, pos=4)
install.packages("factoextra")
library(factoextra)
fviz_cluster (k4, data = iris2)
fviz_nbclust(iris2, kmeans, method = "gap_stat")
k3=kmeans(iris2,centers=3, nstart = 25)
fviz_cluster(k3, data = iris2)
plot(result$Sepal.Length , result$Sepal.Width,col = k3$cluster)
text(result$Sepal.Length , result$Sepal.Width,k3$cluster, cex=0.5, col=k4$cluster, pos=4)
plot(result$Petal.Length , result$Petal.Width,col = k3$cluster)
text(result$Petal.Length , result$Petal.Width,k3$cluster, cex=0.5, col=k4$cluster, pos=4)
install.packages(cluster)
library(cluster)
k.medois.fit <- pam(iris2, 3, diss=FALSE)
k.medois.fit
fviz_cluster(k.medois.fit, data = iris2)
library(cluster)
agn = agnes (iris2, metric = "euclidean", stand = TRUE)
View(agn)
plot(agn)
group.2<-cutree(agn,2)
group.2
group.10<-cutree(agn,10)
group.10
result.agnes.2<-cbind(iris2,group.2)
View(result.agnes.2)
plot(result.agnes.2$Sepal.Length , result.agnes.2$Sepal.Width,col = group.2)
text(result.agnes.2$Sepal.Length , result.agnes.2$Sepal.Width,  cex=0.5, col=group.2, pos=4)
plot(result.agnes.2$Petal.Length , result.agnes.2$Petal.Width,col = group.2)
text(result.agnes.2$Petal.Length , result.agnes.2$Petal.Width,  cex=0.5, col=group.2, pos=4)
install.packages("dbscan")
library(dbscan)
dbresult=dbscan(iris2, 30, minPts = 5)
dbresult
dbresult$cluster
plot (iris2,col=dbresult$cluster)
iris4<-cbind(iris2,dbresult$cluster)
par("mar")
par(mar=c(3,3,3,3))
```

Swati Chauhan
MSTM ADV

Outputs

```
> View(iris)
> iris2<-data.frame(scale (iris[-5], center=TRUE, scale =TRUE))
> View(iris2)
> View(iris2)
> k3 = kmeans (iris2, centers=3, nstart = 25)
> k3
K-means clustering with 3 clusters of sizes 50, 53, 47

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1  -1.01119138  0.85041372   -1.3006301  -1.2507035
2  -0.05005221 -0.88042696    0.3465767   0.2805873
3   1.13217737  0.08812645    0.9928284   1.0141287

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3
2 2 2 3 2 2
 [60] 2 2 2 2 2 2 3 2 2 2 2 3 2 2 2 2 3 3 3 2 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3
3 2 2 3 3 3
[119] 3 2 3 2 3 2 3 3 2 3 3 3 3 3 3 2 2 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2

Within cluster sum of squares by cluster:
[1] 47.35062 44.08754 47.45019
 (between_SS / total_SS =  76.7 %)
Available components:
[1] "cluster"     "centers"     "totss"       "withinss"    "tot.withinss" "betweenss"   "size"        "iter"
[9] "ifault"

K-means clustering with 4 clusters of sizes 25, 53, 47, 25

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1  -0.71894419  1.50198969   -1.2972312  -1.2165934
2  -0.05005221 -0.88042696    0.3465767   0.2805873
3   1.13217737  0.08812645    0.9928284   1.0141287
4  -1.30343857  0.19883774   -1.3040289  -1.2848136

Clustering vector:
  [1] 1 4 4 4 1 1 4 4 4 4 1 4 4 4 1 1 1 1 1 1 1 1 4 4 4 4 1 1 4 4 1 1 1 4 4 1 1 4 4 1 4 4 1 1 4 1 4 1 4 3 3 3
2 2 2 3 2 2
 [60] 2 2 2 2 2 2 3 2 2 2 2 3 2 2 2 2 3 3 3 2 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3
3 2 2 3 3 3
[119] 3 2 3 2 3 2 3 3 2 3 3 3 3 3 3 2 2 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2

Within cluster sum of squares by cluster:
[1] 12.147537 44.087545 47.450194  9.646348
 (between_SS / total_SS =  81.0 %)
Available components:
[1] "cluster"     "centers"     "totss"       "withinss"    "tot.withinss" "betweenss"   "size"        "iter"
```

```
[9] "ifault"

Medoids:
     ID Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]   8   -1.0184372   0.7861738   -1.2791040  -1.3110521
[2,] 113    1.1553023  -0.1315388    0.9868021   1.1816087
[3,]  56   -0.1730941  -0.5903951    0.4203256   0.1320673
Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
3 3 3 2 3 3
 [60] 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 2 2 2 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 2 3 2 2 2 2 3 2 2 2 2
2 3 2 2 2 2
[119] 2 3 2 3 2 3 2 2 3 3 2 2 2 2 2 3 3 2 2 2 3 2 2 2 3 2 2 2 3 2 2 2 3 2 2 3
Objective function:
   build     swap
0.9205107 0.8757051

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"  "clusinfo"   "silinfo"    "diss"       "call"
[10] "data"

> group.2<-cutree(agn,2)
> group.2
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 2
2 2 2 2 2 2
 [60] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2
[119] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
> group.10<-cutree(agn,10)
> group.10
  [1]  1  2  2  2  1  1  1  1  2  2  1  1  2  2  3  3  1  1  1  1  1  1  1  1  2  1  1  1  2  2  1  3  3  2  2  1  1  2
 [40]  1  1  4  2  1  1  2  1  2  1  2  5  5  5  6  6  6  5  7  5  6  7  6  8  6  6  5  6  6  8  6  5  6  6  6  6  5  5  5
 [79]  6  6  6  6  6  6  6  5  5  8  6  6  6  6  6  7  6  6  6  6  7  6  5  6  5  5  5  9  6  9  6 10  5  6  5  6  6  5  5
[118] 10  9  8  5  6  9  6  5  9  6  5  5  9  9 10  5  6  6  9  5  5  5  5  5  5  6  5  5  5  6  5  5  5

DBSCAN clustering for 150 objects.
Parameters: eps = 30, minPts = 5
The clustering contains 1 cluster(s) and 0 noise points.

  1
150

Available fields: cluster, eps, minPts
> dbresult$cluster
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 [60] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
[119] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```