# Group 1: Multivariate analysis of australian climate data

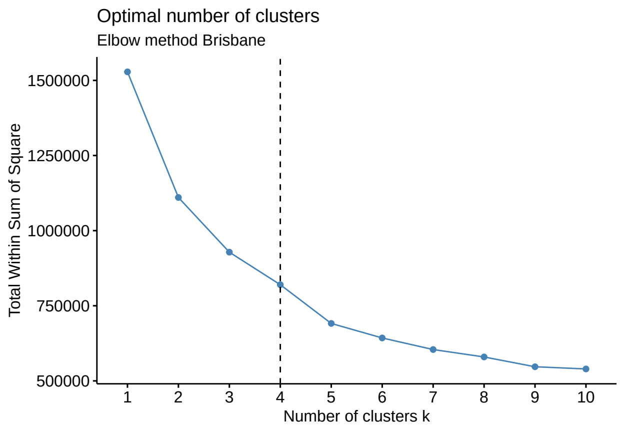
## Data input

To perform the clustering analysis are used the original datasets (numeric variables) for Brisbane, Perth and Cairns.

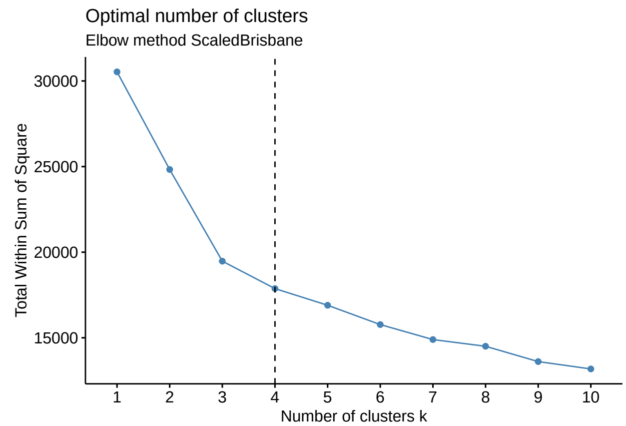
### Clustering Analysis

In order to analyze if data presents patterns of association are it is performed a clustering analysis. For this purpose, all incomplete cases remaining are removed and as a first step, the optimal number of clusters are estimated through direct methods: elbow, average silhouette and ASM to choose the most common value of optimal clusters.

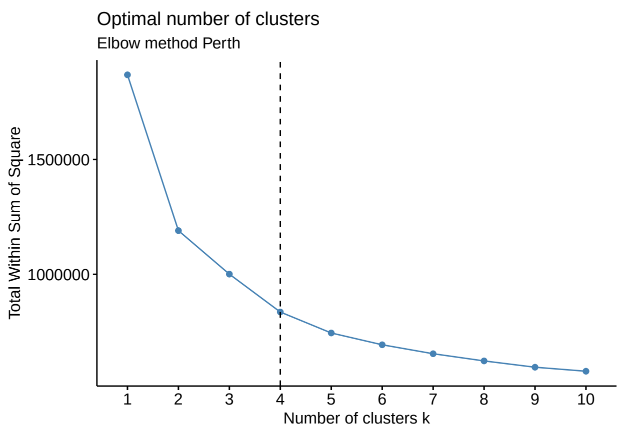
```
par(mar = c(4,4,.1,.1))
fun01<-function(x){ tmp_df = listall[[x]]
                    tmp_name = names(listall)[x]
                    fviz_nbclust(tmp_df, kmeans, method = "wss") +
                    geom_vline(xintercept = 4, linetype = 2) +
                    labs(subtitle = paste("Elbow method",tmp_name))}
fun02<-function(x){ tmp_df = listall[[x]]
                    tmp_name = names(listall)[x]
                    fviz_nbclust(tmp_df, kmeans, method = "silhouette") +
                    geom_vline(xintercept = 4, linetype = 2) +
                    labs(subtitle = paste("Elbow method",tmp_name))}
fun03<-function(x){ tmp_df = listall[[x]]
                    tmp_name = names(listall)[x]
                    fviz_nbclust(tmp_df, kmeans, method = "gap_stat") +
                    geom_vline(xintercept = 4, linetype = 2) +
                    labs(subtitle = paste("Elbow method",tmp_name))}

wss<-lapply(1:length(listall),fun01)
wss

silhouette<-lapply(1:length(listall),fun02)
silhouette

#Gaps<-lapply(1:length(listall),fun03)
#Gaps
```



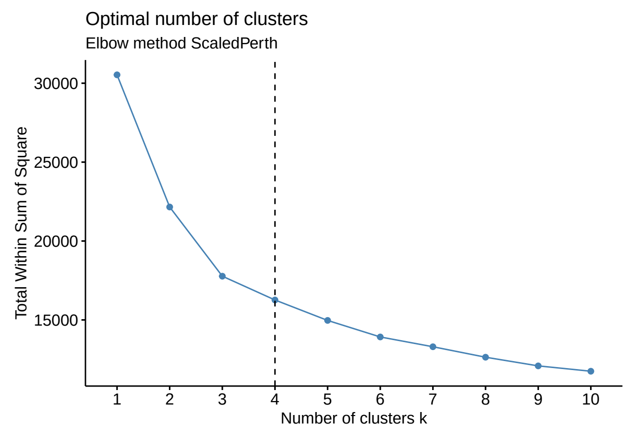
Optimal number of clusters
Elbow method Brisbane
Total Within Sum of Square
1500000
1250000
1000000
750000
500000
1 2 3 4 5 6 7 8 9 10
Number of clusters k

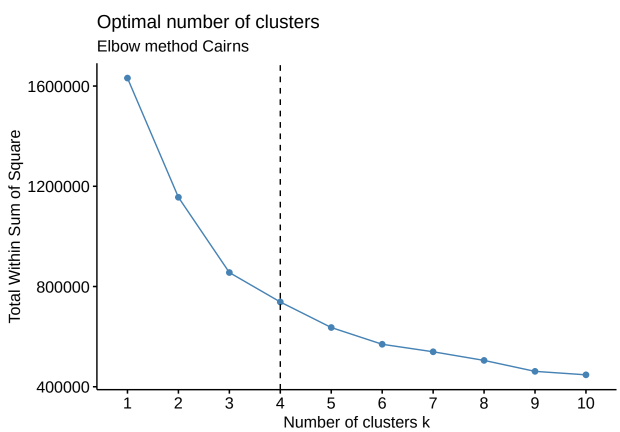
Optimal number of clusters
Elbow method ScaledBrisbane
Total Within Sum of Square
30000
25000
20000
15000
1 2 3 4 5 6 7 8 9 10
Number of clusters k

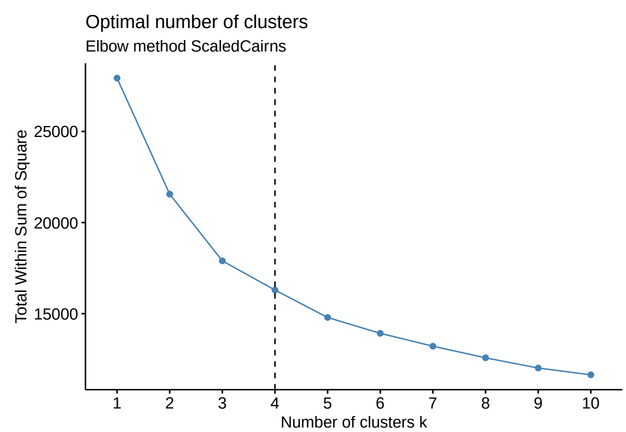
Optimal number of clusters
Elbow method Perth
Total Within Sum of Square
1500000
1000000
1 2 3 4 5 6 7 8 9 10
Number of clusters k

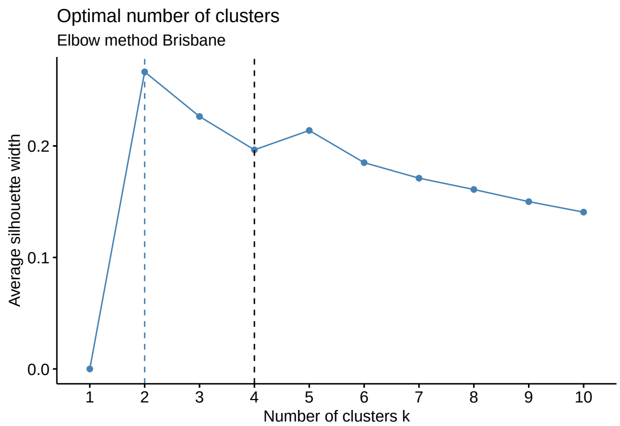
Optimal number of clusters
Elbow method ScaledPerth
Total Within Sum of Square
30000
25000
20000
15000
1 2 3 4 5 6 7 8 9 10
Number of clusters k

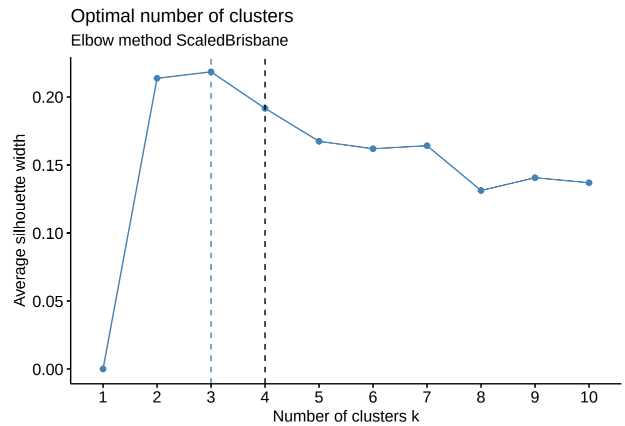



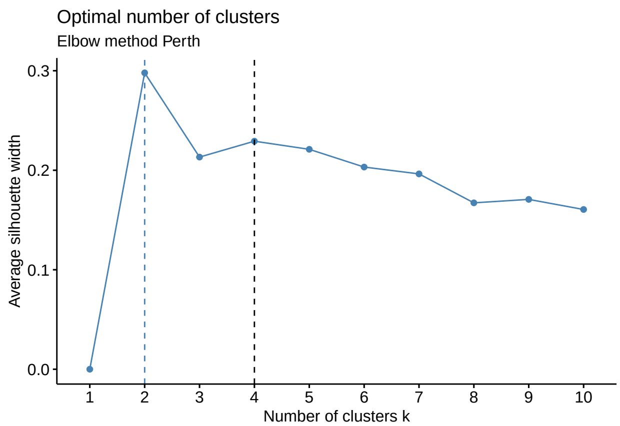
Optimal number of clusters
Elbow method Cairns
Total Within Sum of Square
1600000
1200000
800000
400000
1 2 3 4 5 6 7 8 9 10
Number of clusters k

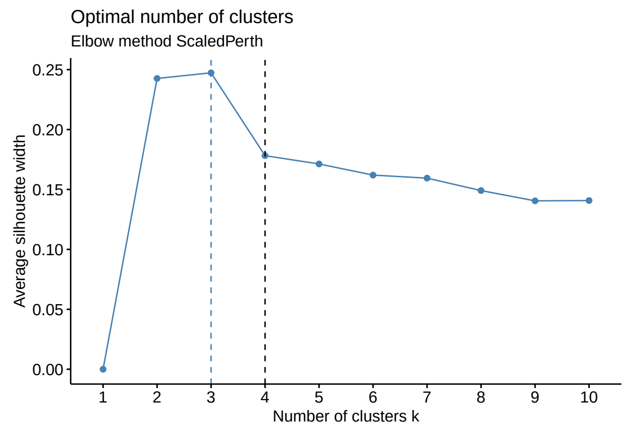



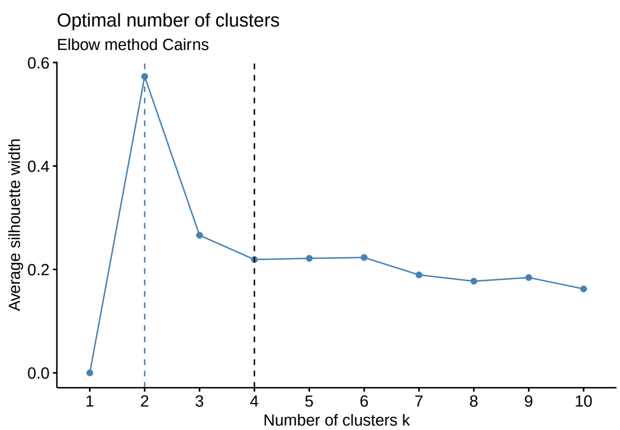
Optimal number of clusters
Elbow method ScaledCairns
Total Within Sum of Square
25000
20000
15000
1 2 3 4 5 6 7 8 9 10
Number of clusters k

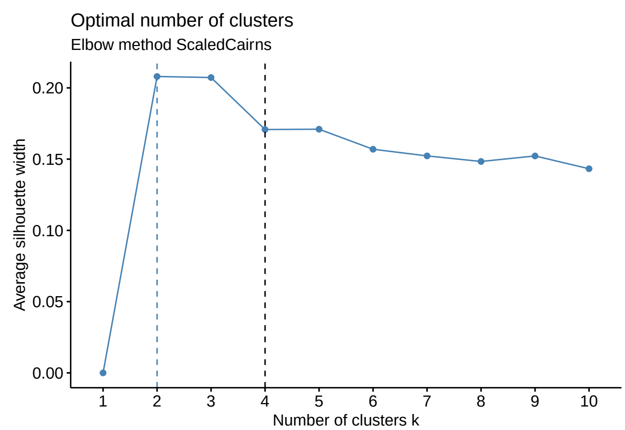


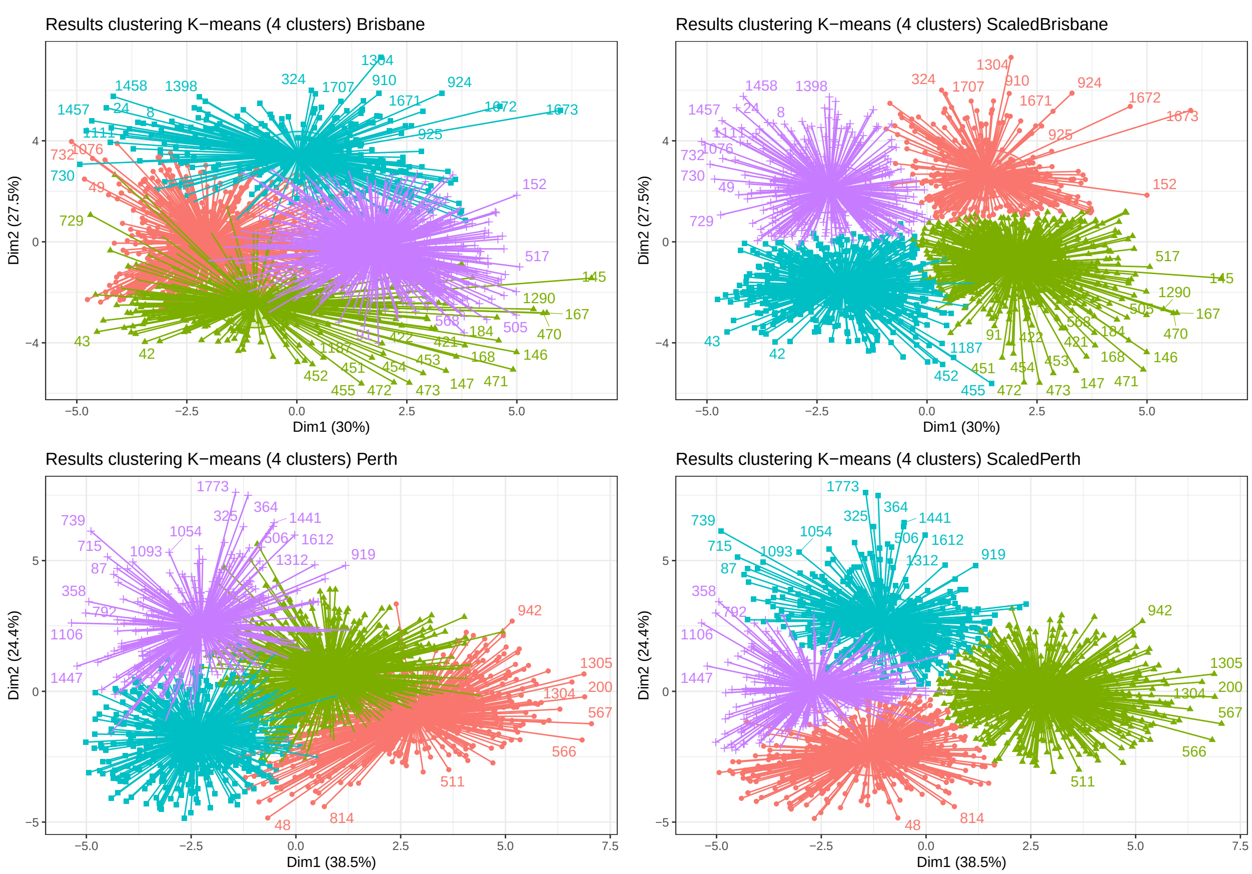



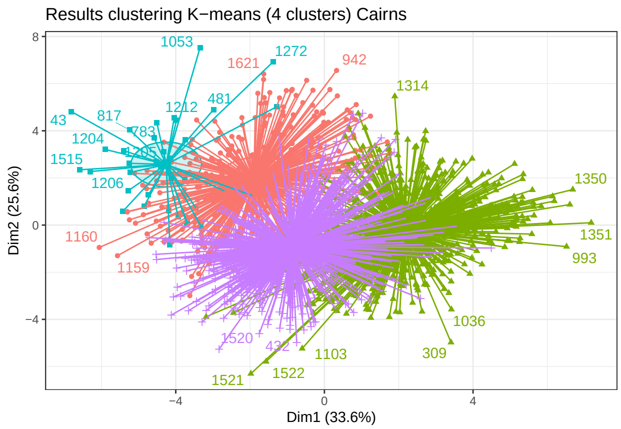

Given the results provided by the methods, it can be concluded the clustering can be performed with 4 cluster for all the dataset, the original numerical variables and the coordinates of the performed MCA.

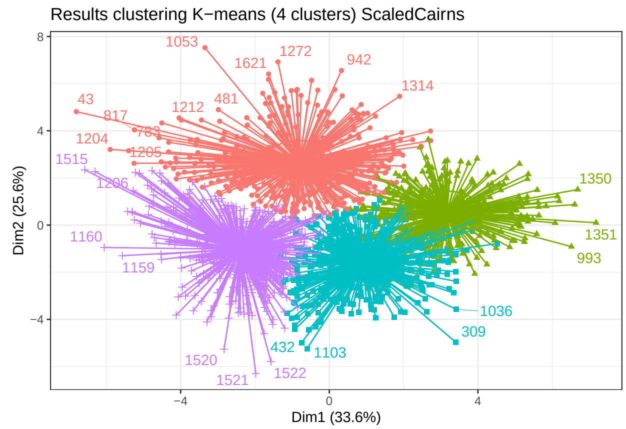
```
funVizKm<- function(i){ tmp_df = listall[[i]];
                        tmp_kmeans = kmeans(x = listall[[i]], centers = 4)
                        tmp_name = names(listall)[i]
                        fviz_cluster(object = tmp_kmeans, data = listall[[i]],
                                     show.clust.cent = TRUE, ellipse.type = "euclid",
                                     star.plot = TRUE, repel = TRUE) +
                     theme_bw() + theme(legend.position = "none") +
                          labs(title = paste("Results clustering K-means (4 clusters)",
                                             tmp_name))}
```

```
VizKmeans<-lapply(1:length(listall),funVizKm)
VizKmeans
```

```
## Warning: ggrepel: 1751 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning: ggrepel: 1751 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning: ggrepel: 1771 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning: ggrepel: 1771 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning: ggrepel: 1618 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning: ggrepel: 1618 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



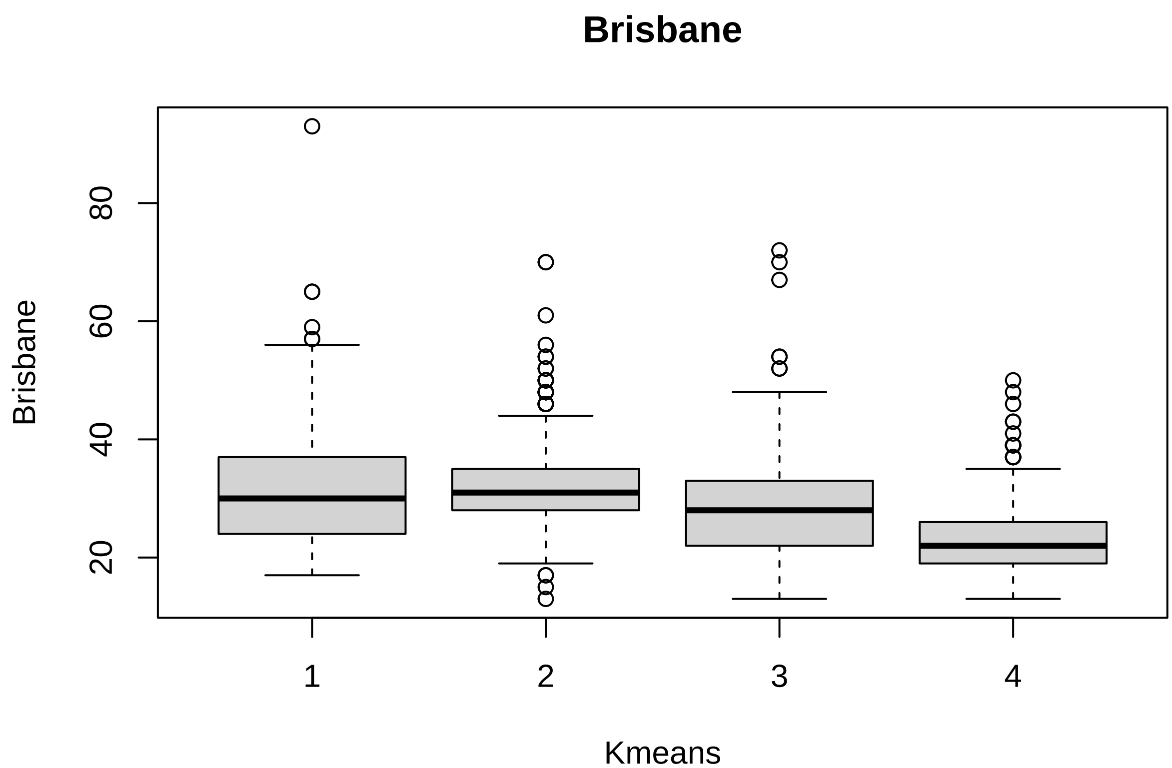


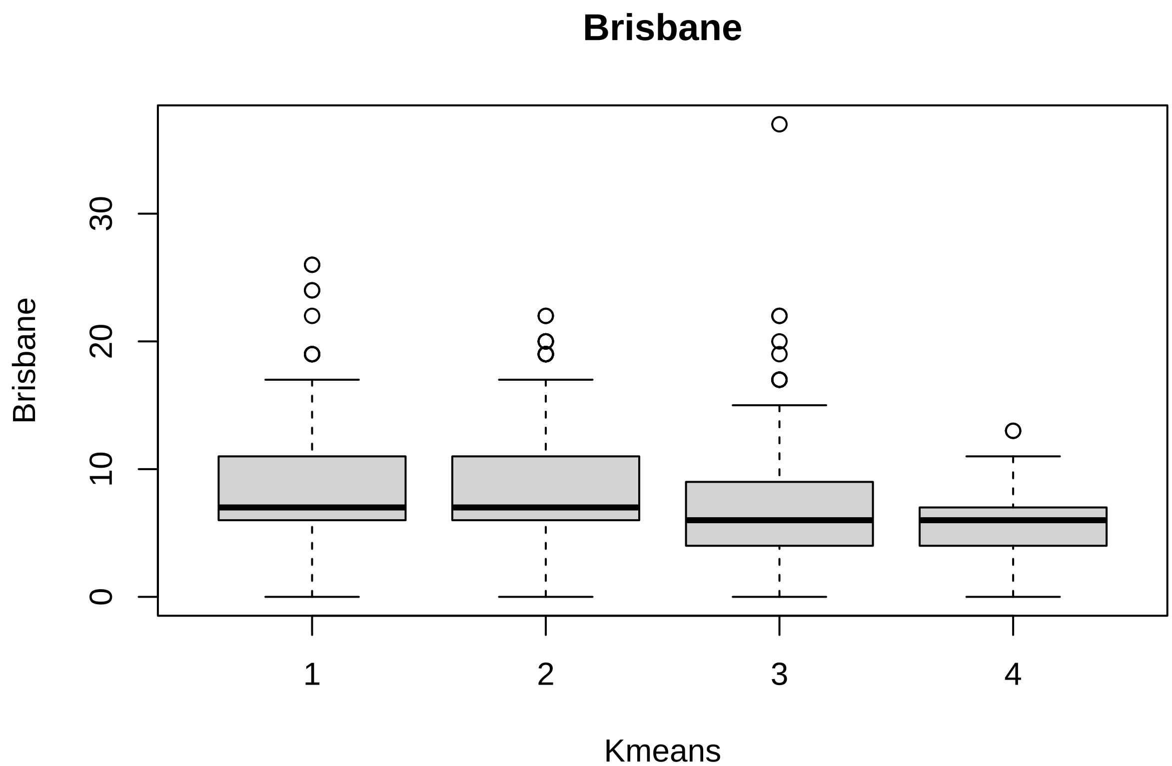



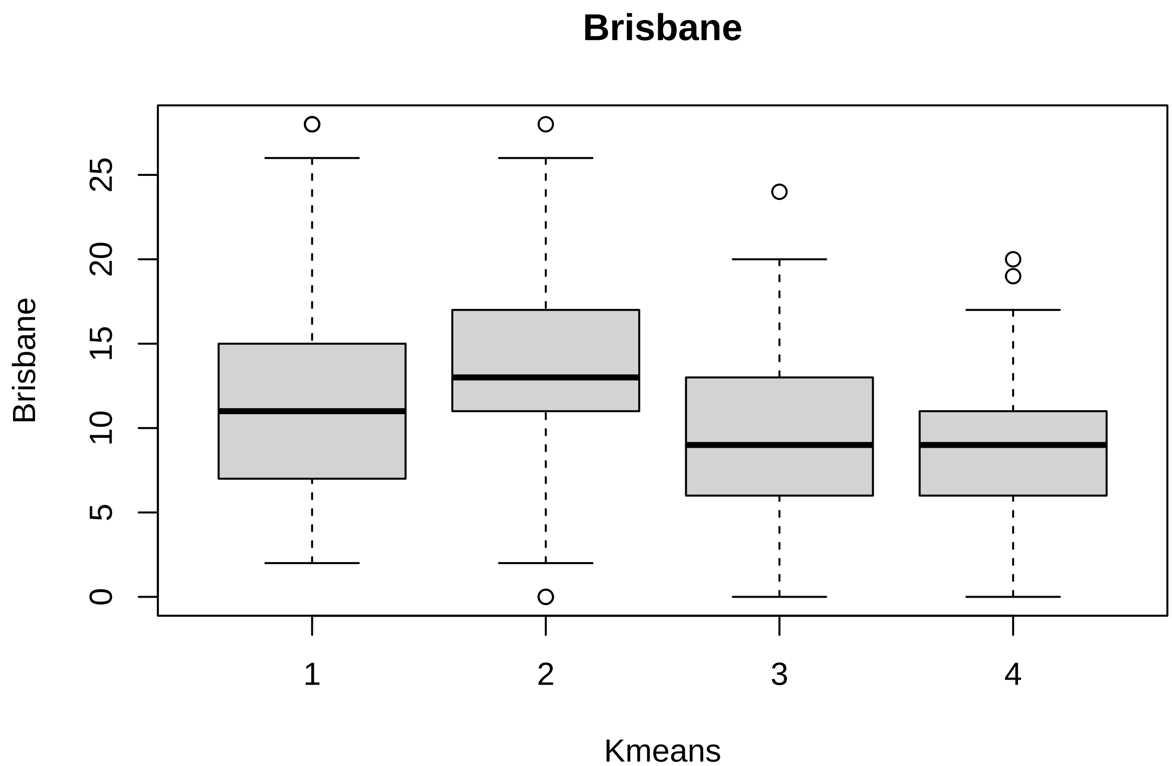

```r
funKm<- function(i){ tmp_df = listall[[i]];
                     tmp_kmeans = kmeans(x = listall[[i]], centers = 4)
                     listall[[i]]<-add_column(listall[[i]], KmeansCluster =
                                                        tmp_kmeans$cluster)}

Kmeans<-lapply(1:length(listall),funKm)
names(Kmeans)<-c("Brisbane","ScaledBrisbane","Perth","ScaledPerth",
                 "Cairns","ScaledCairns")

#a<-Kmeans[[1]]

#fun08<-function(a,i) {
#  for (i in 1:17) {boxplot(a[,i] ~ a[,18], xlab = 'Kmeans', ylab = names(a)[i])}}

#lapply(Kmeans,fun08)

#boxplot(a[,1] ~ a[,18])
#boxplot(a[,2] ~ a[,18])

#boxplot(a)

#plot(formula = KmeansCluster ~ ., data = a)


lapply(1:length(Kmeans), function(x){

  # Get the dataframe and the name
  tmp_df = Kmeans[[x]]
  tmp_name = names(Kmeans)[x]

  for (i in 1:17) {boxplot(tmp_df[,i] ~ tmp_df[,18], xlab = 'Kmeans', ylab = tmp_name, main = tmp_name)
  })
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
```

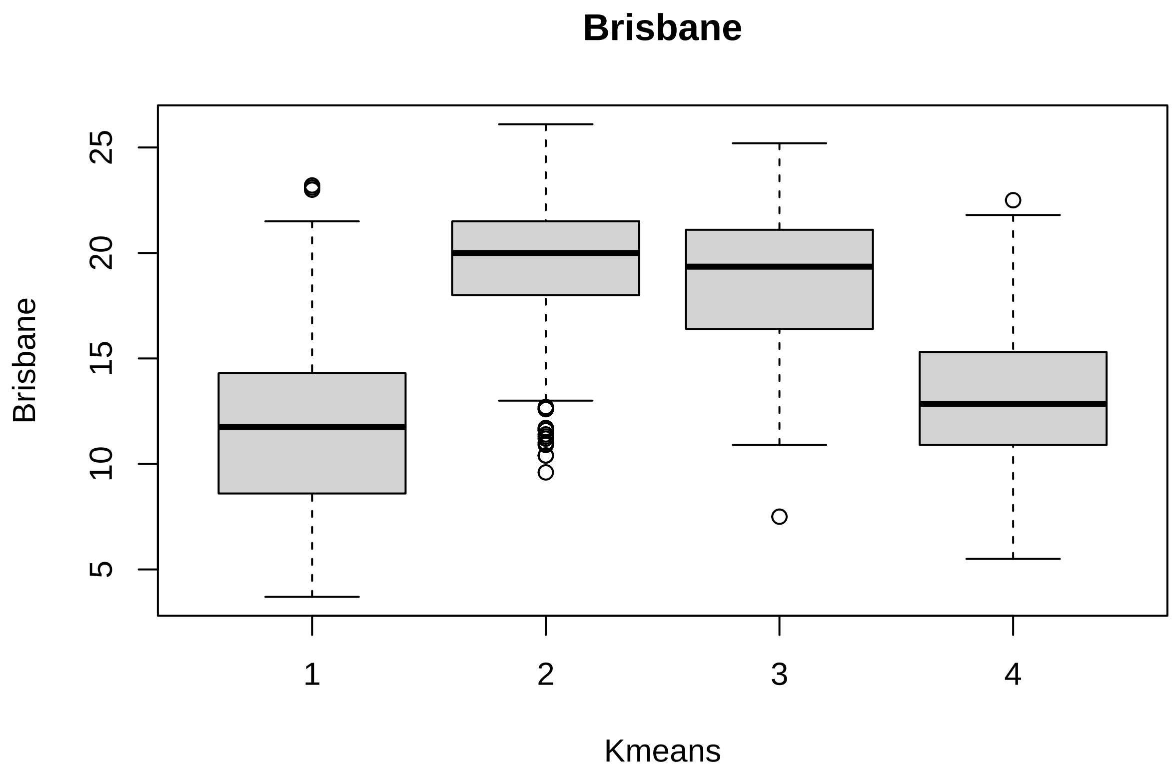
```
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
```

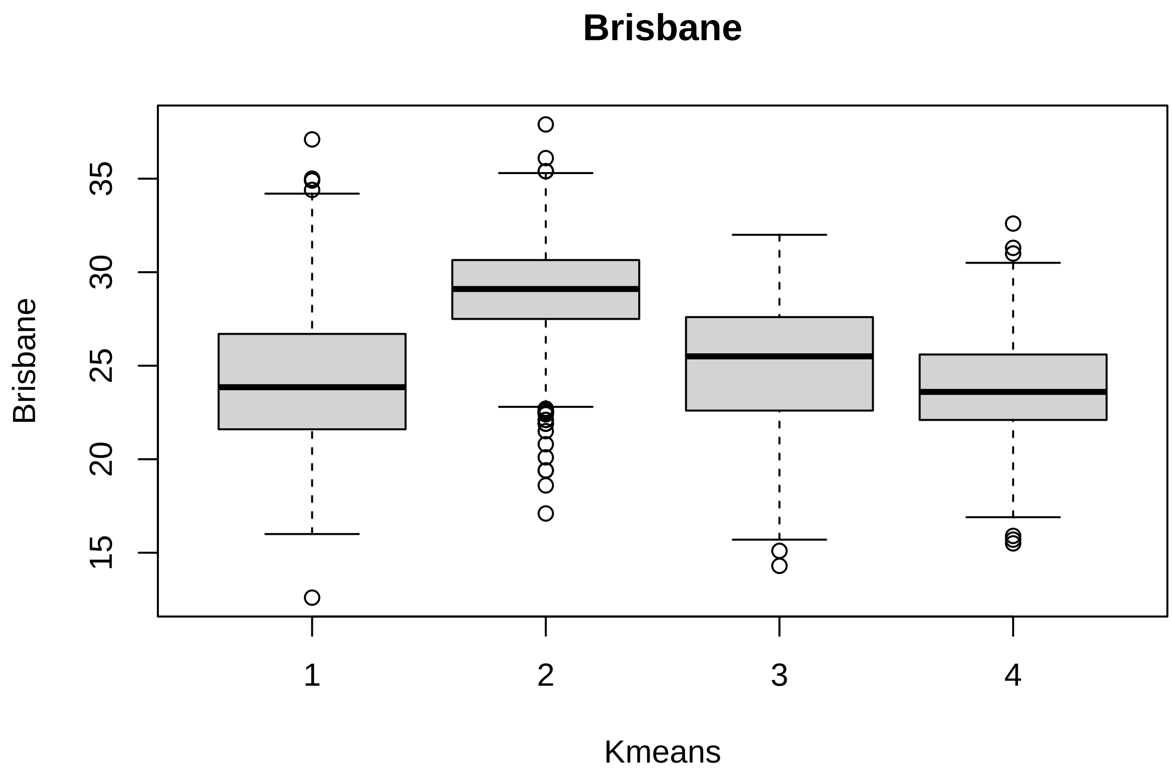


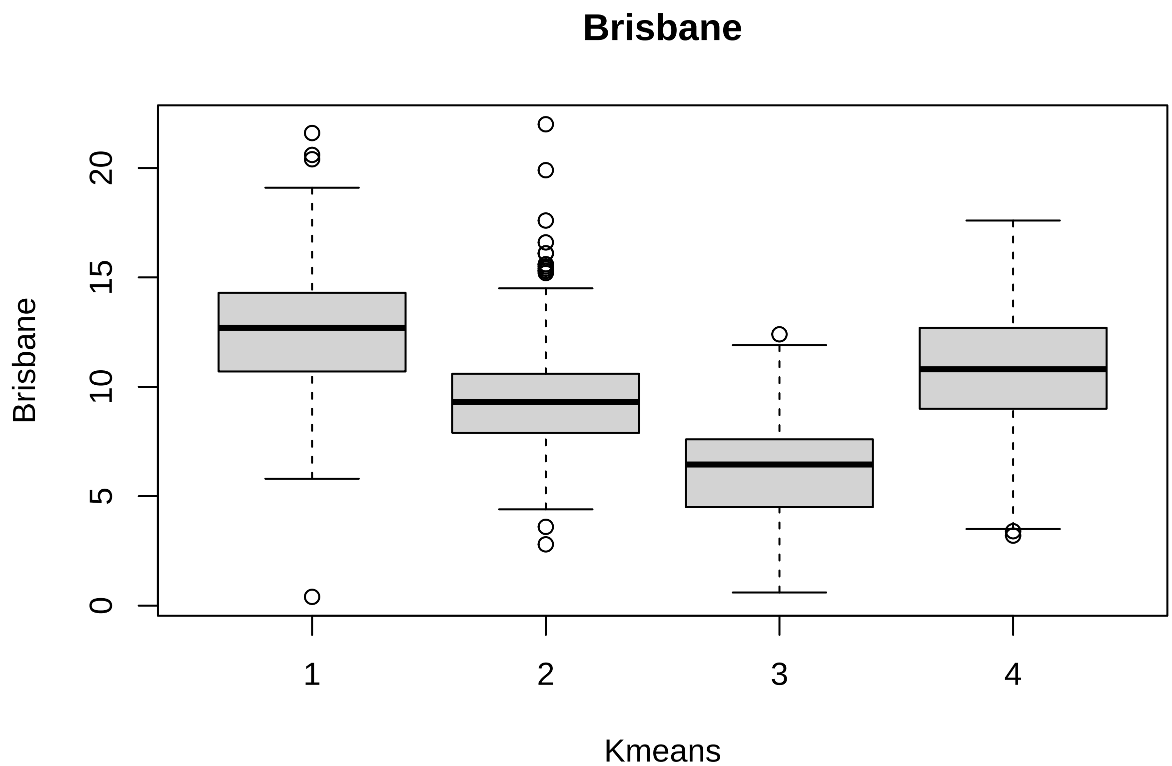

Brisbane
Brisbane
0
10
20
30
1
2
3
4
Kmeans



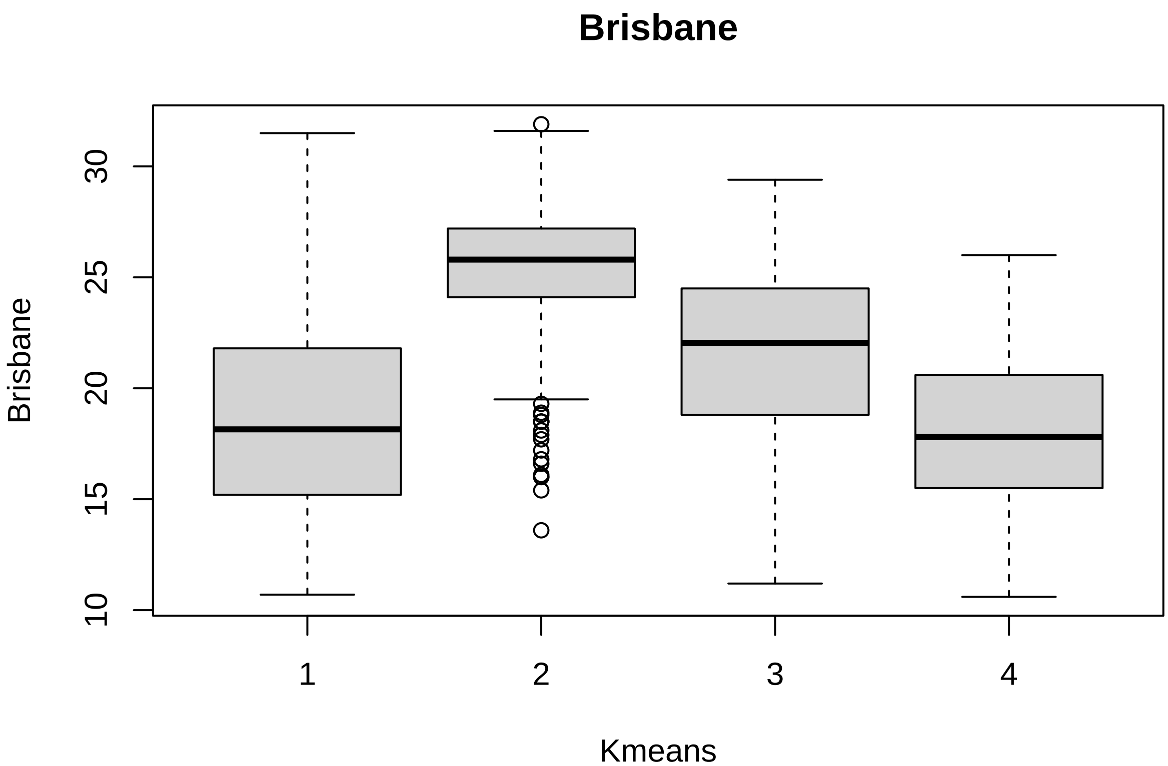
Brisbane
Brisbane
0
5
10
15
20
25
1
2
3
4
Kmeans

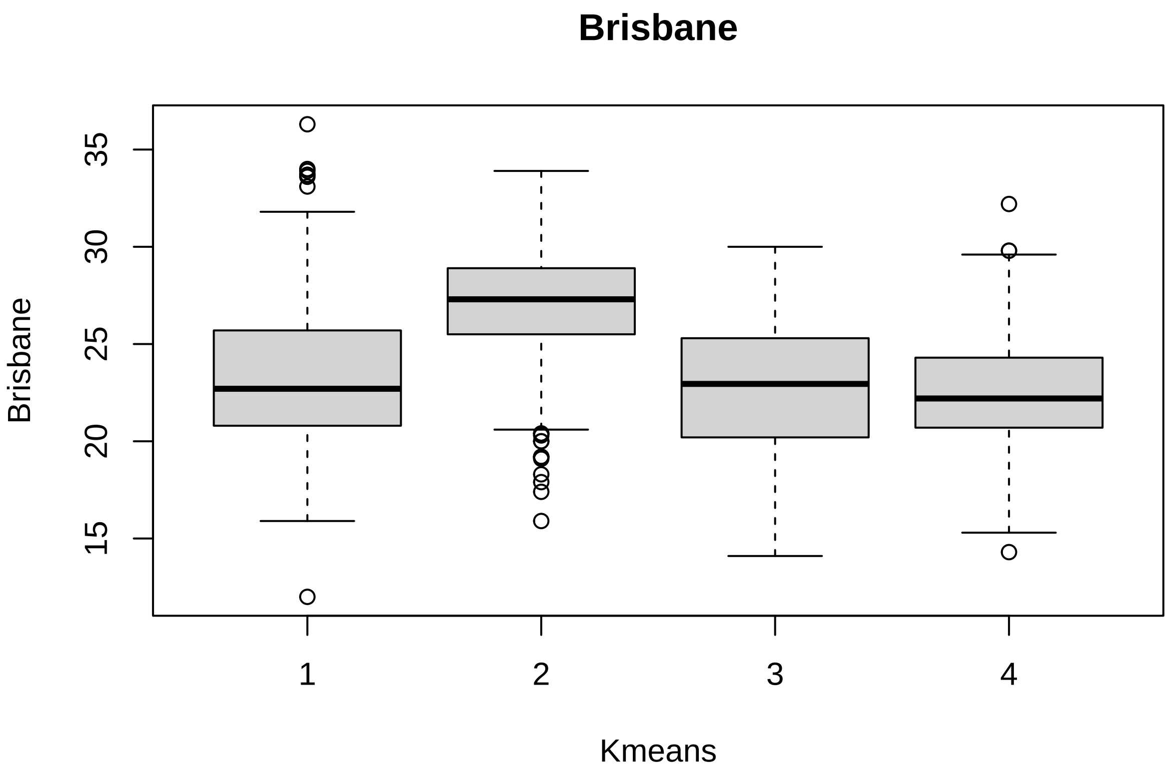

Brisbane
Brisbane
5
10
15
20
25
1
2
3
4
Kmeans


Brisbane
Brisbane
15
20
25
30
35
1
2
3
4
Kmeans

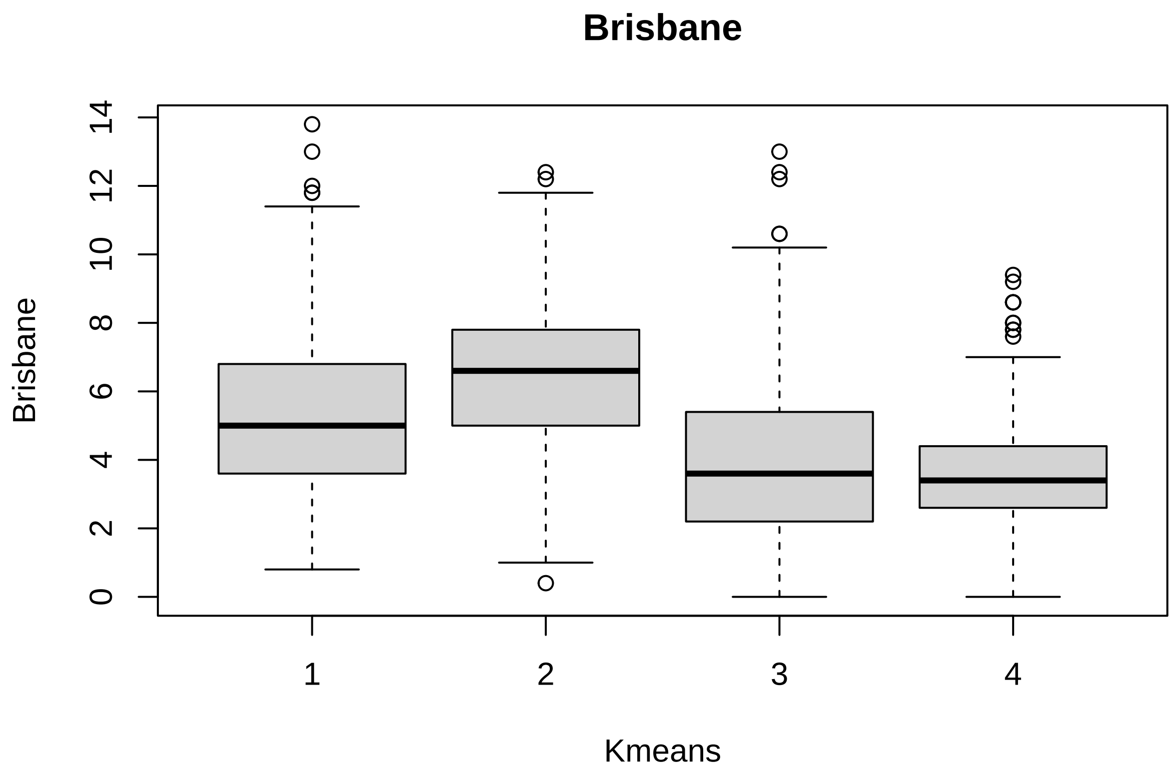


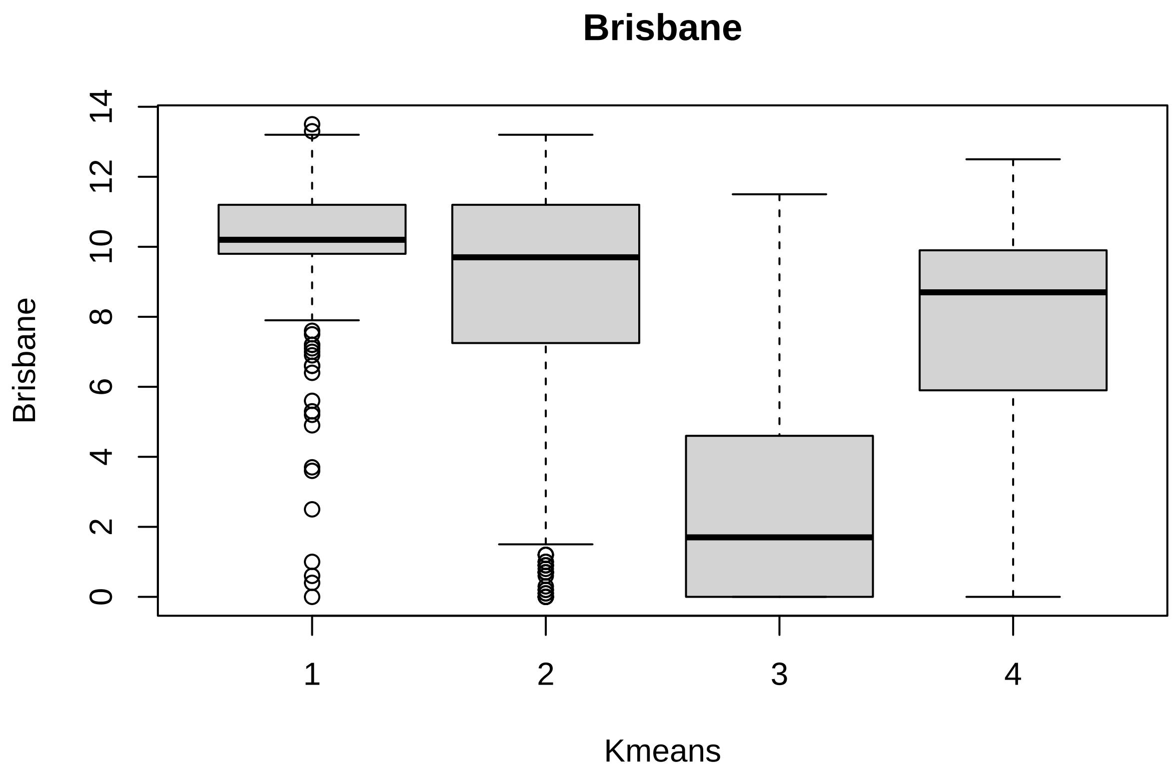
Brisbane
Brisbane
0
5
10
15
20
1
2
3
4
Kmeans

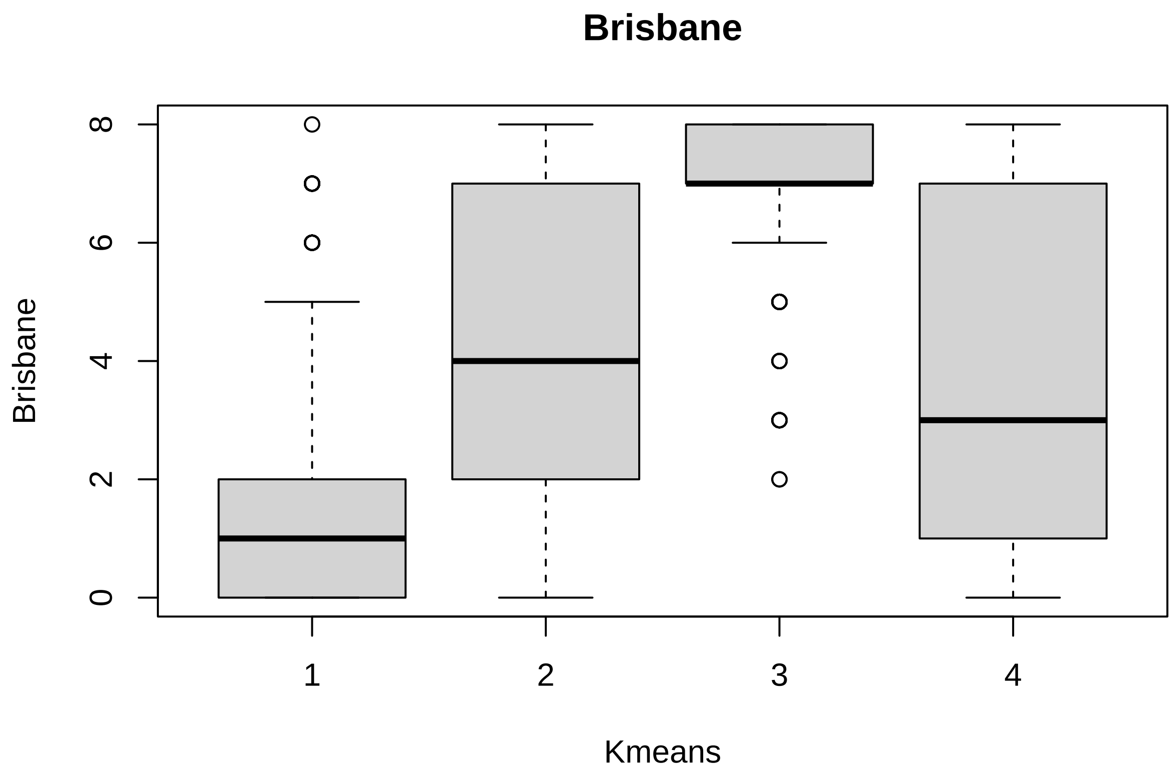

Brisbane
Brisbane
10
15
20
25
30
1
2
3
4
Kmeans

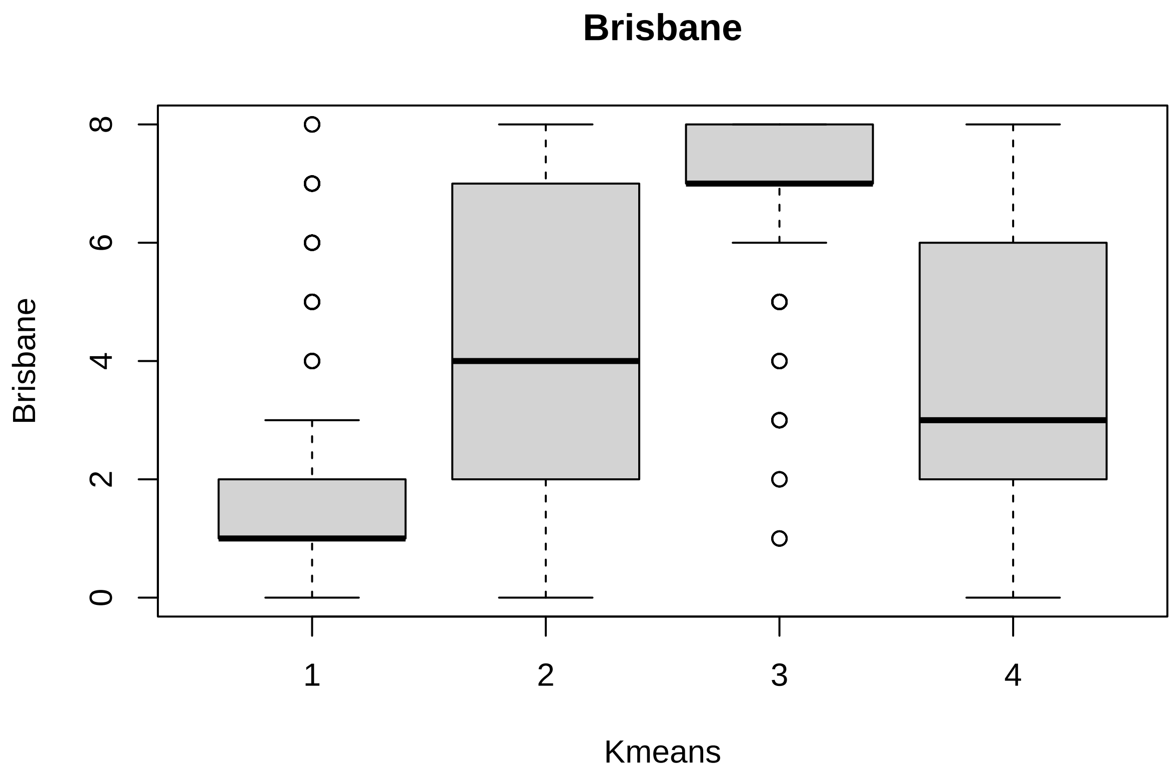


Brisbane
35
30
25
20
15
Brisbane
1
2
3
4
Kmeans

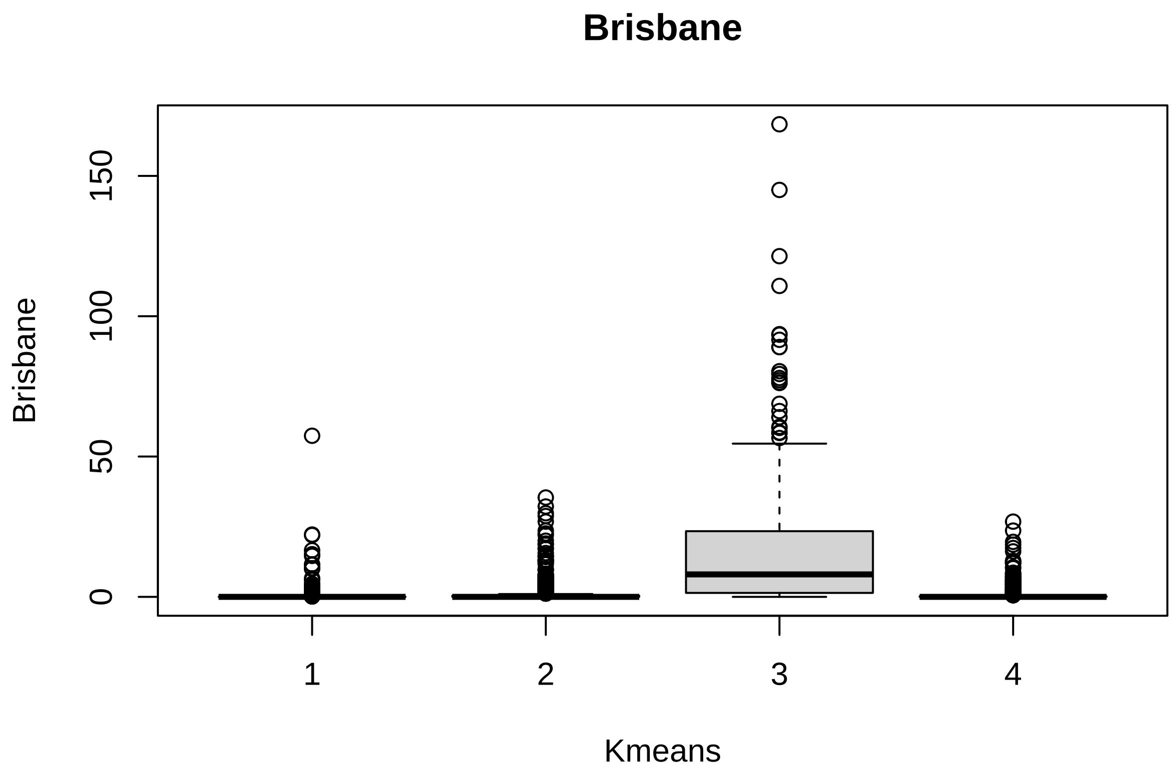

Brisbane
Brisbane
0
2
4
6
8
10
12
14
1
2
3
4
Kmeans

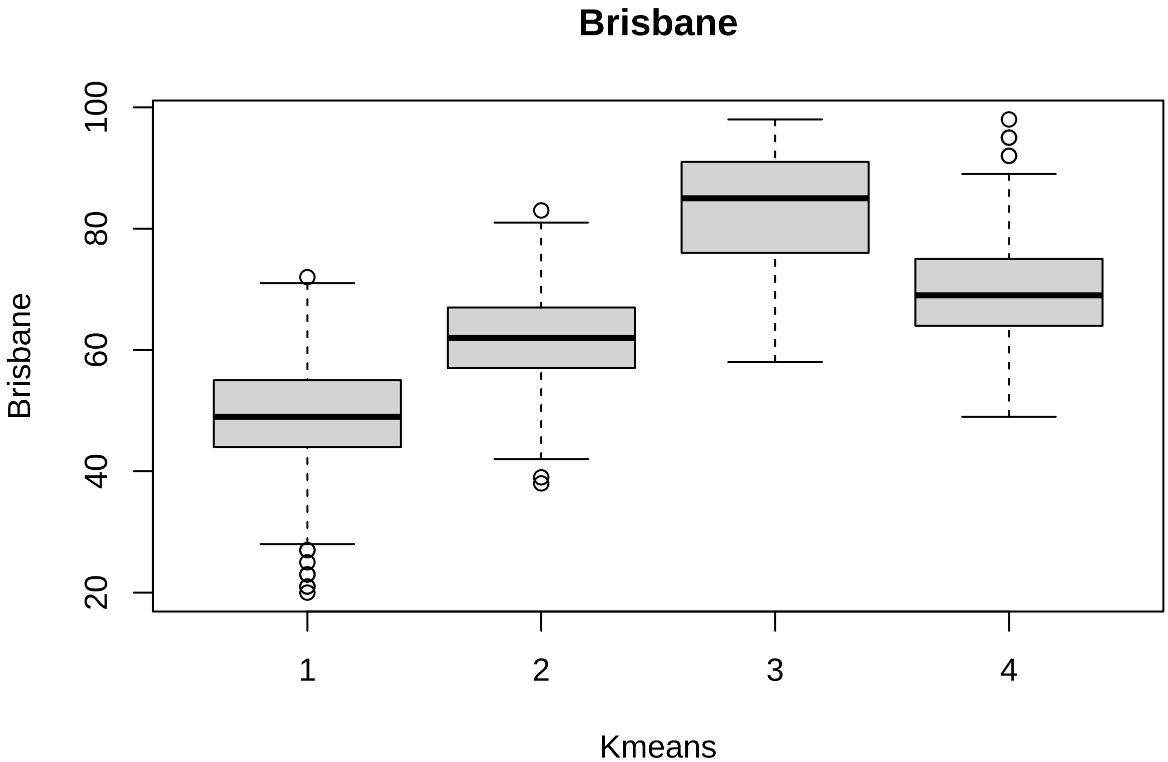

Brisbane
Brisbane
0
2
4
6
8
10
12
14
1
2
3
4
Kmeans

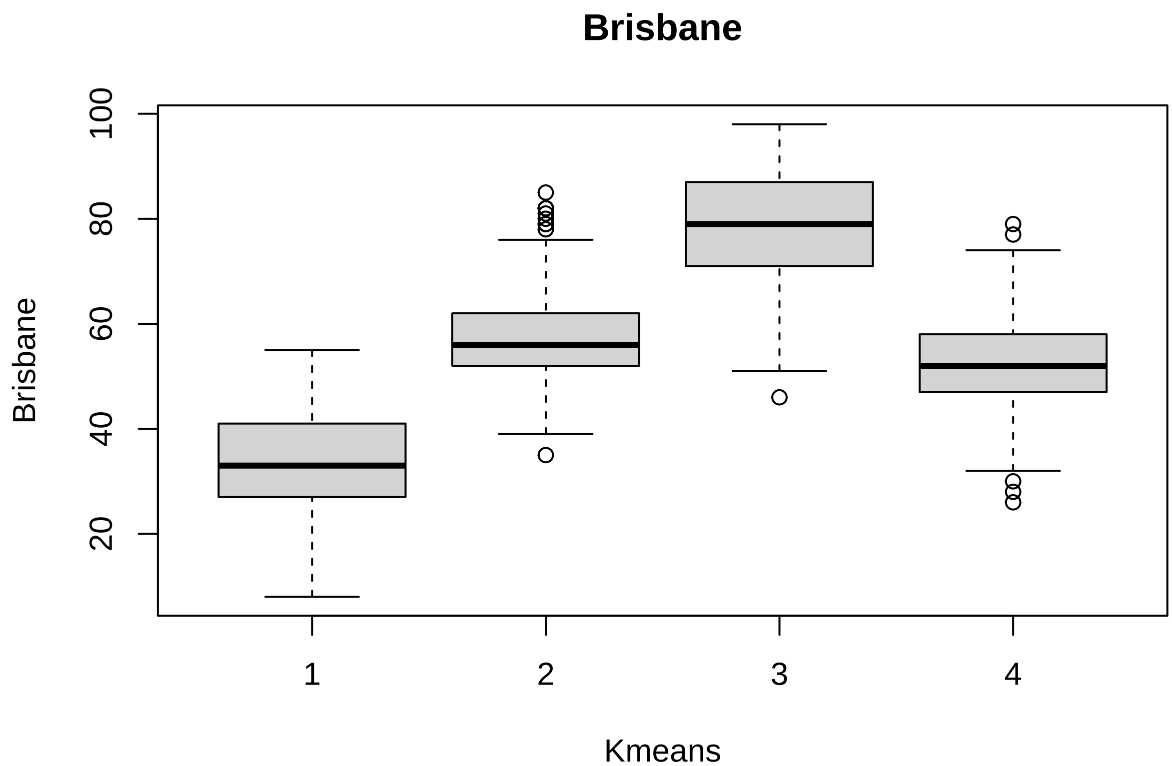

Brisbane
Brisbane
0
2
4
6
8
1
2
3
4
Kmeans

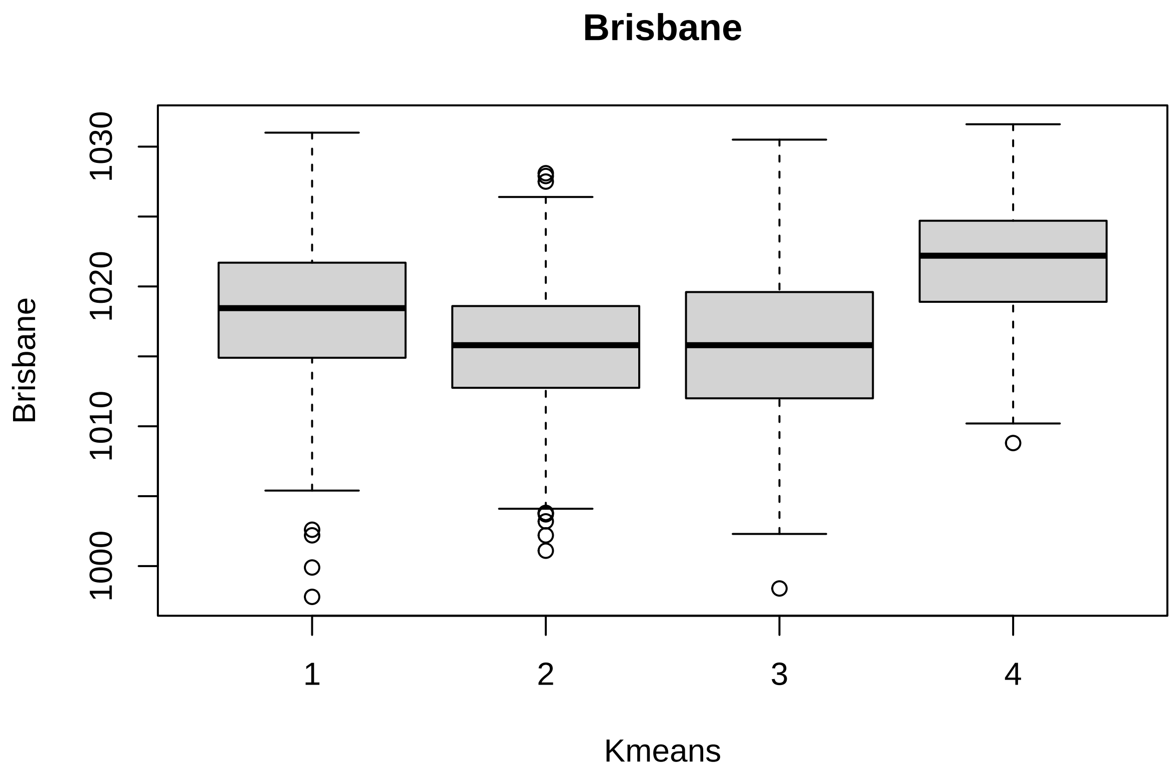


Brisbane
Brisbane
0
2
4
6
8
1
2
3
4
Kmeans

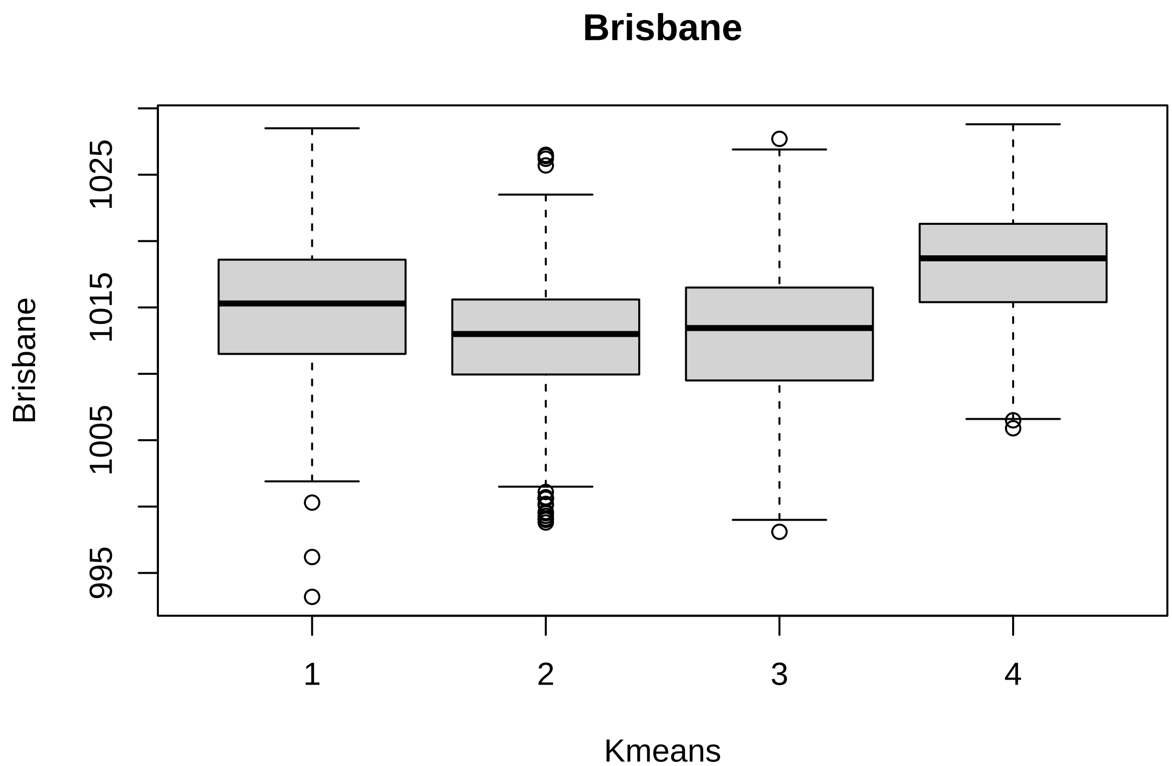

Brisbane
Brisbane
0
50
100
150
1
2
3
4
Kmeans


Brisbane
100
80
60
40
20
Brisbane
1
2
3
4
Kmeans

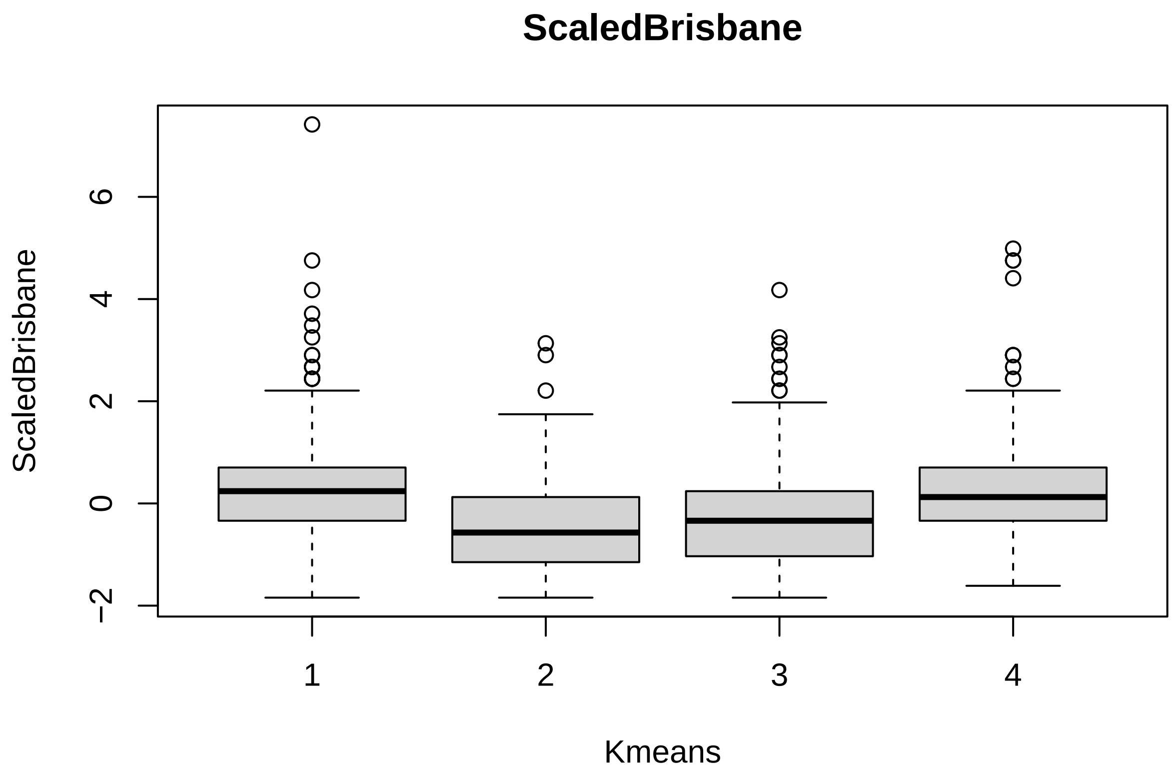

Brisbane
100
80
60
40
20
Brisbane
1
2
3
4
Kmeans

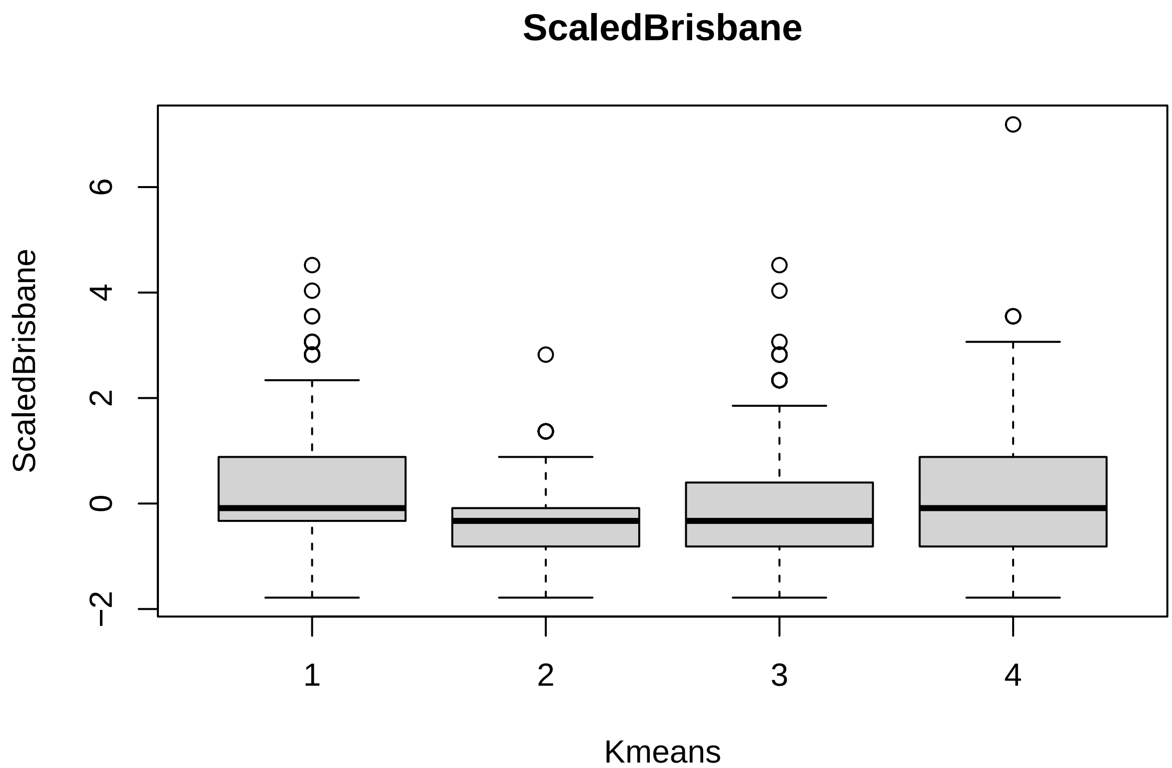


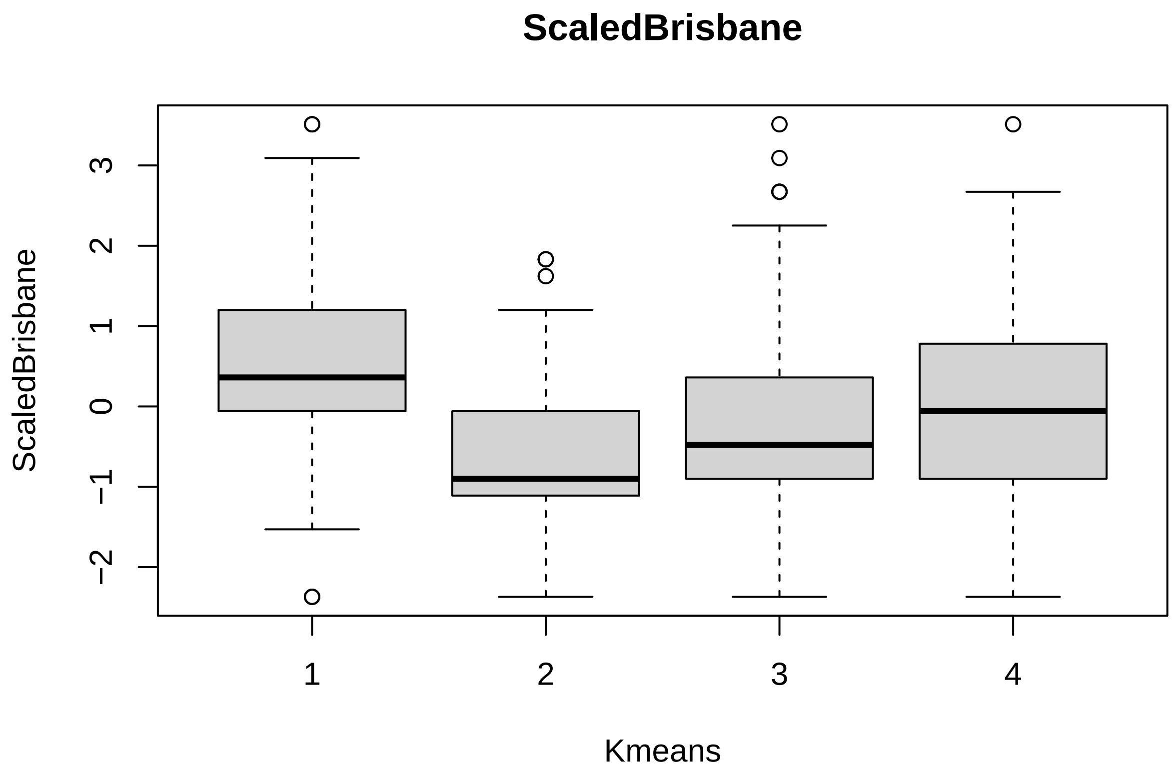
Brisbane
Brisbane
1000
1010
1020
1030
1
2
3
4
Kmeans

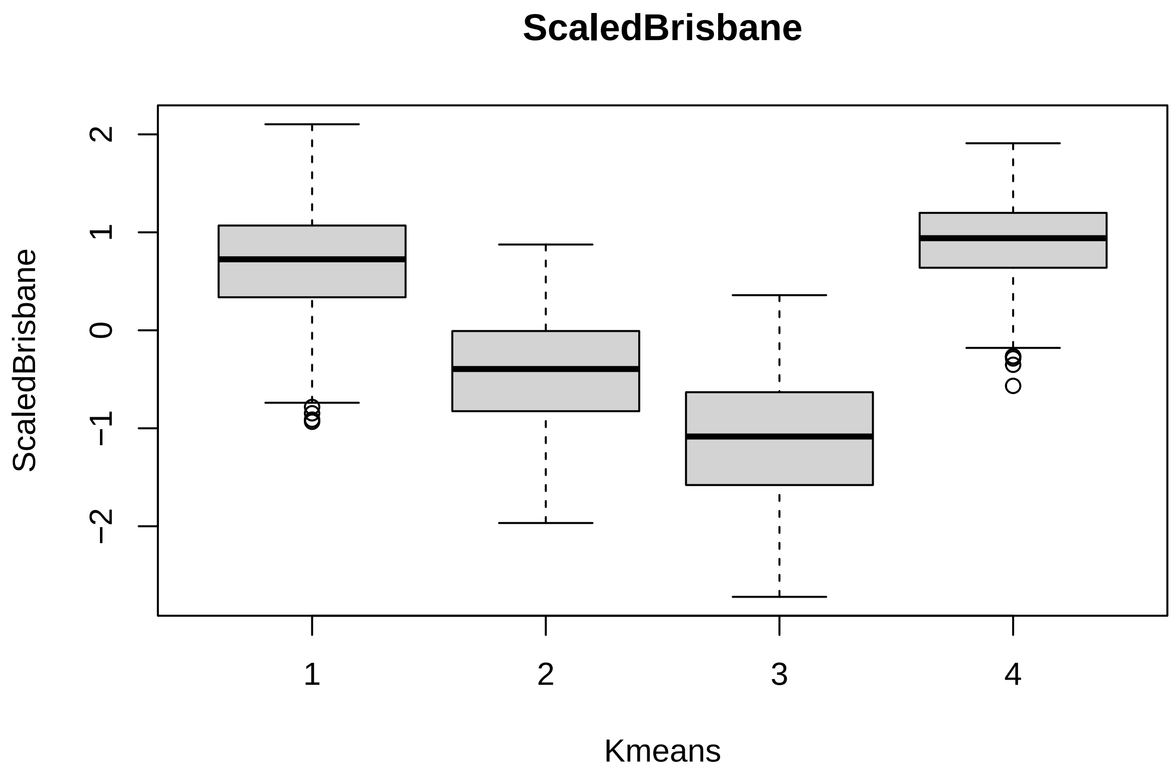


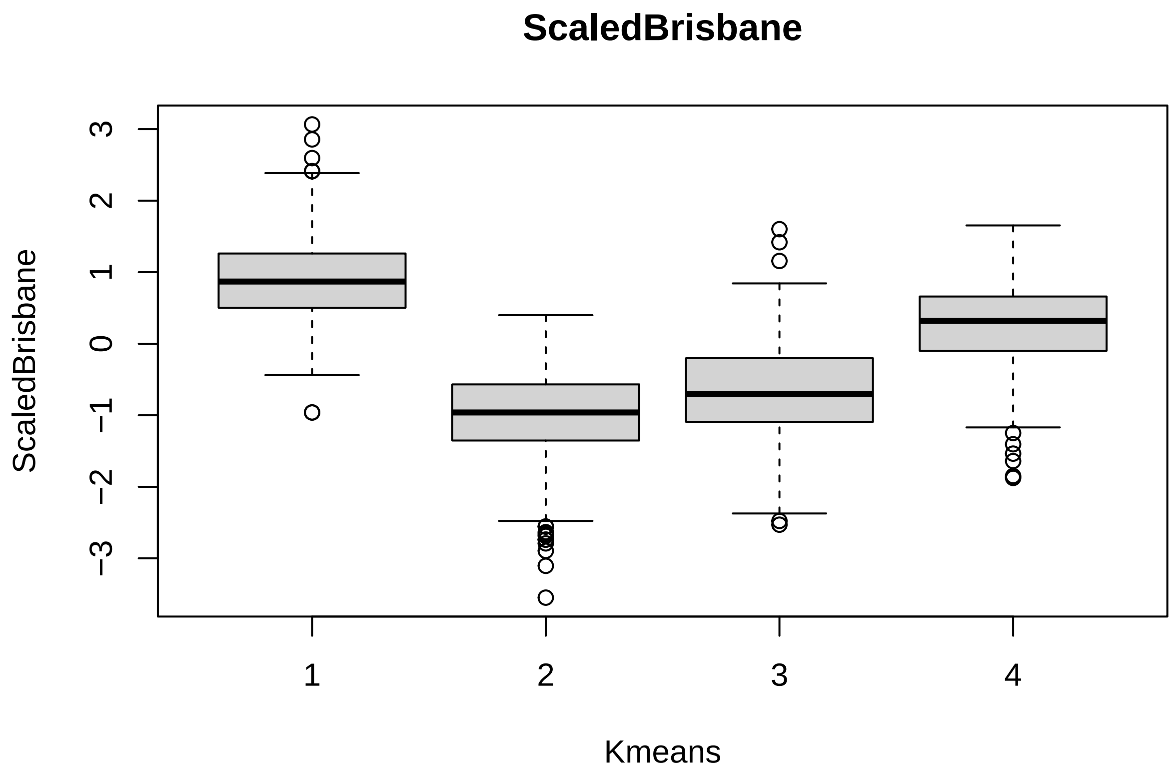
Brisbane
Brisbane
995
1005
1015
1025
1
2
3
4
Kmeans

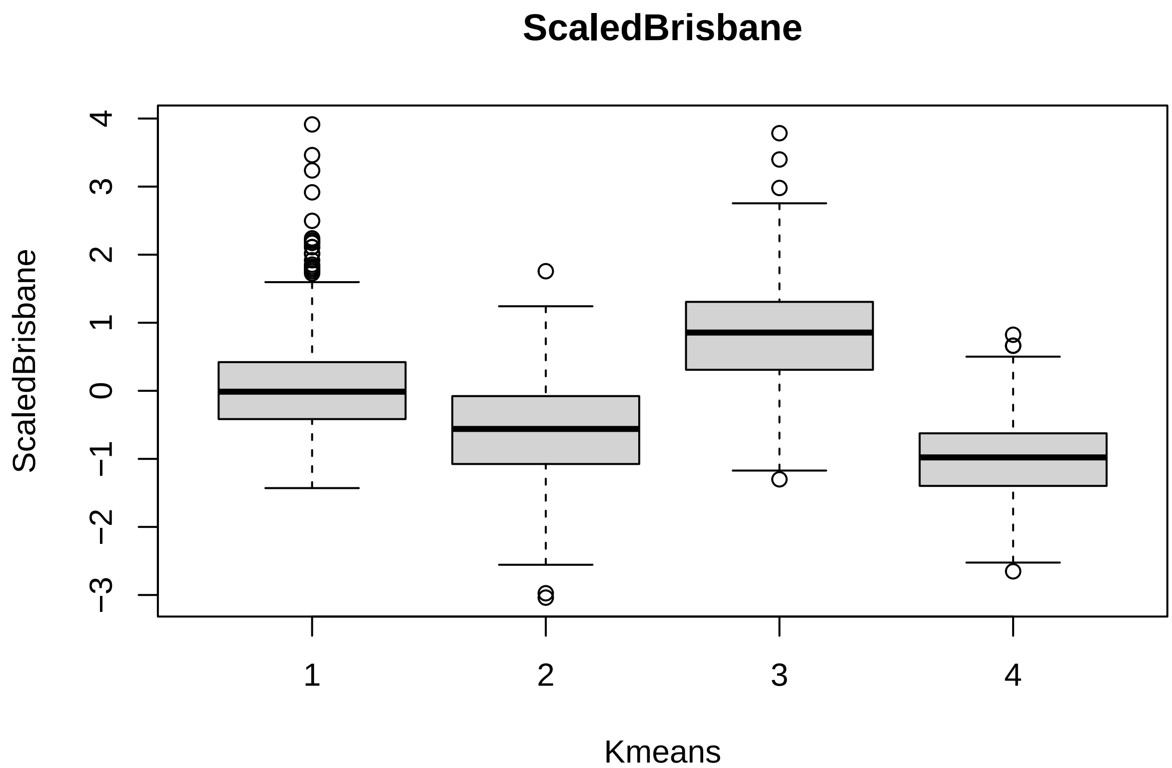

ScaledBrisbane
ScaledBrisbane
Kmeans
−2
0
2
4
6
1
2
3
4

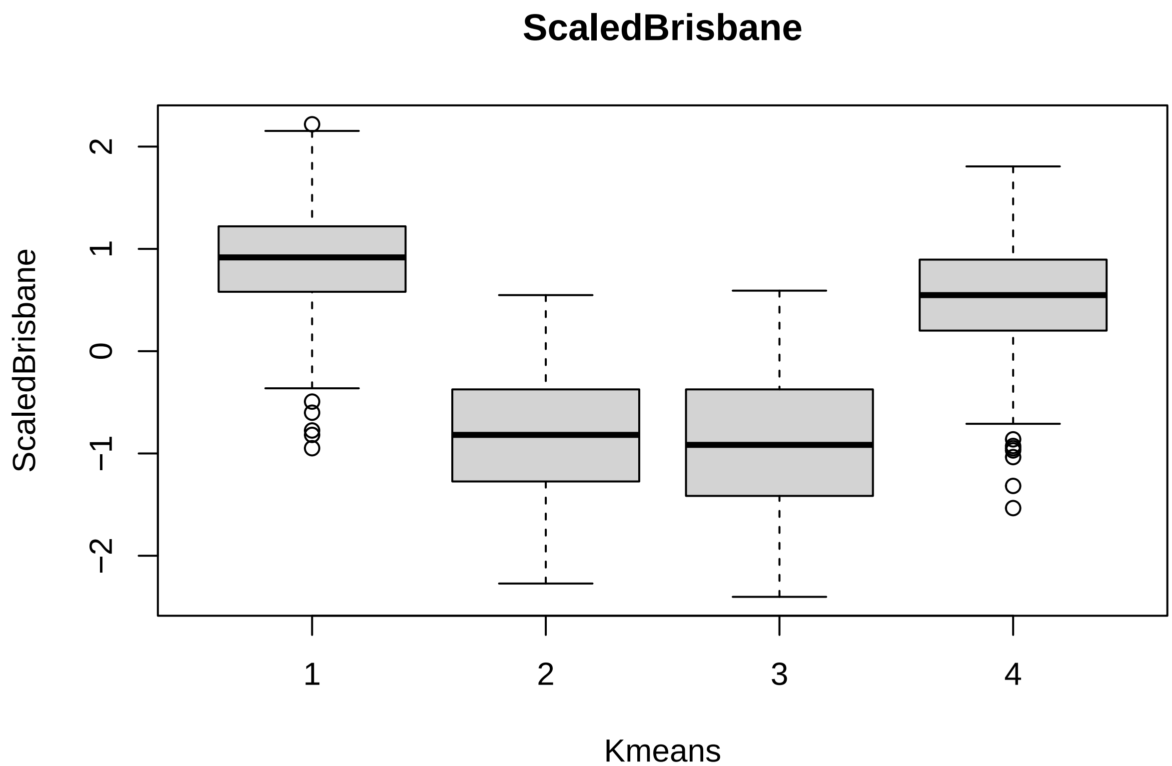

ScaledBrisbane
ScaledBrisbane
Kmeans
−2
0
2
4
6
1
2
3
4


ScaledBrisbane
ScaledBrisbane
3
2
1
0
−1
−2
1
2
3
4
Kmeans

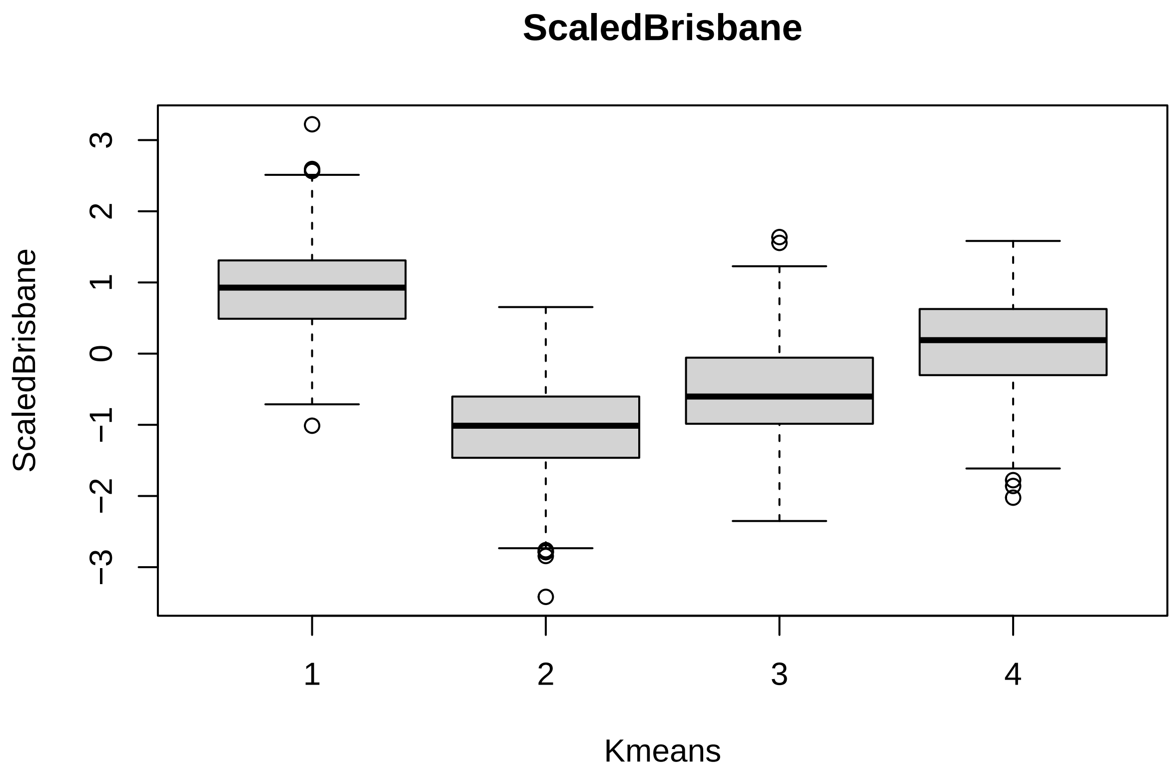

ScaledBrisbane
ScaledBrisbane
Kmeans
1
2
3
4
2
1
0
−1
−2

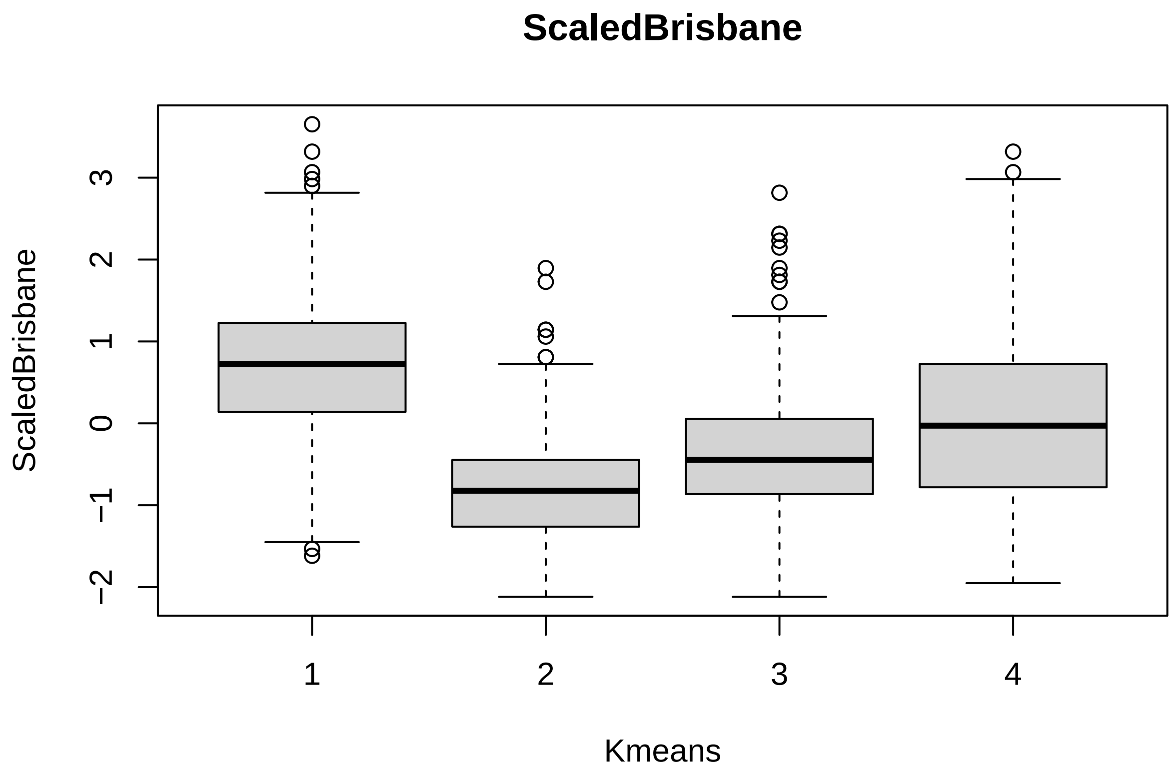

ScaledBrisbane
ScaledBrisbane
3
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

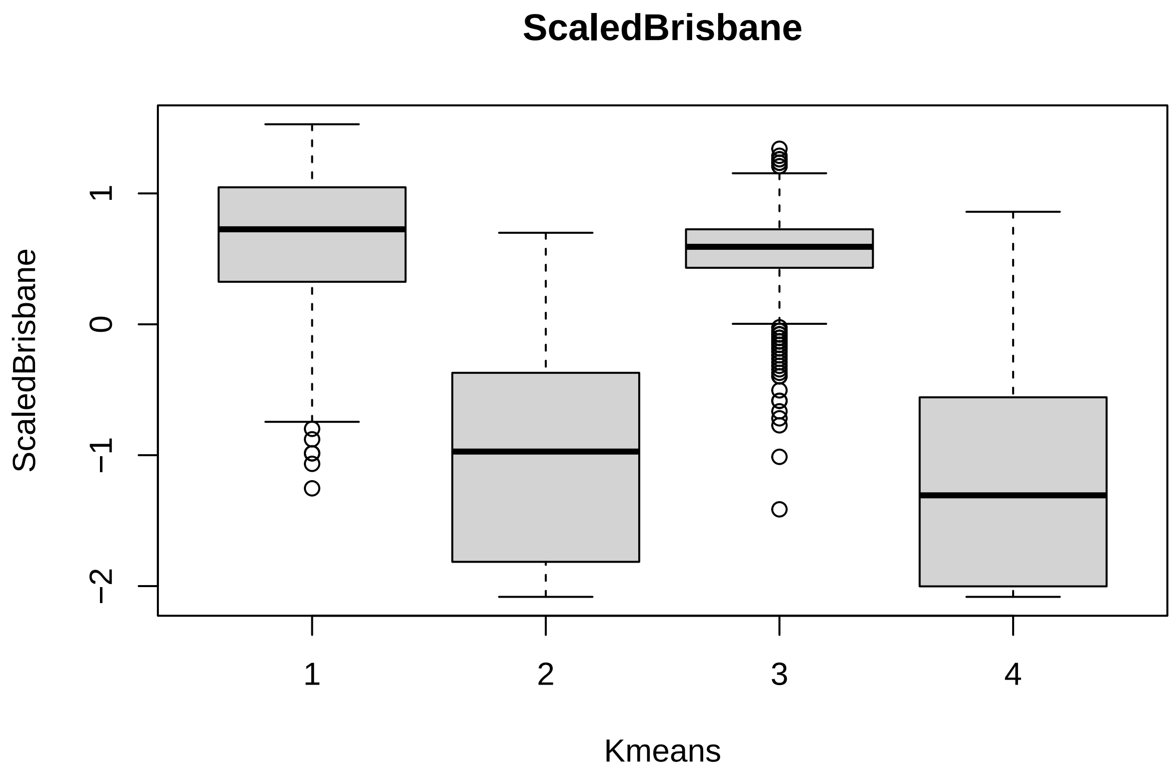

ScaledBrisbane
ScaledBrisbane
Kmeans
1
2
3
4
−3
−2
−1
0
1
2
3
4

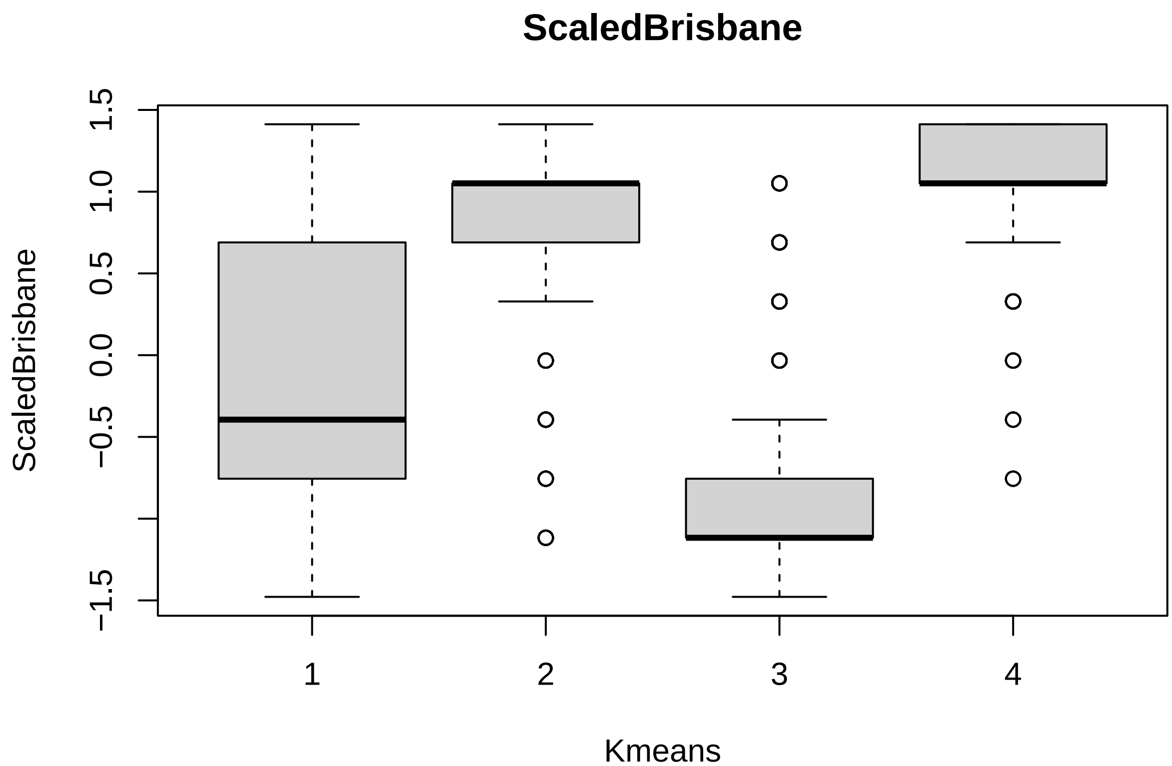


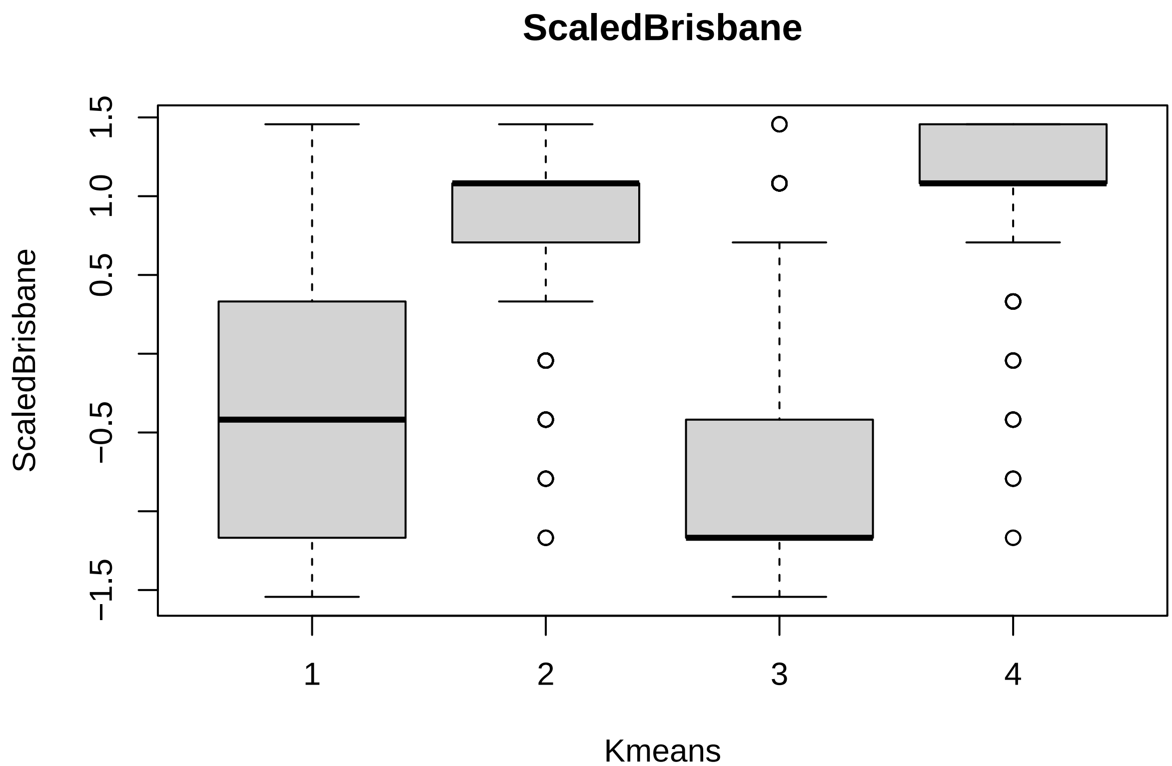
ScaledBrisbane
ScaledBrisbane
Kmeans
−2
−1
0
1
2
1
2
3
4

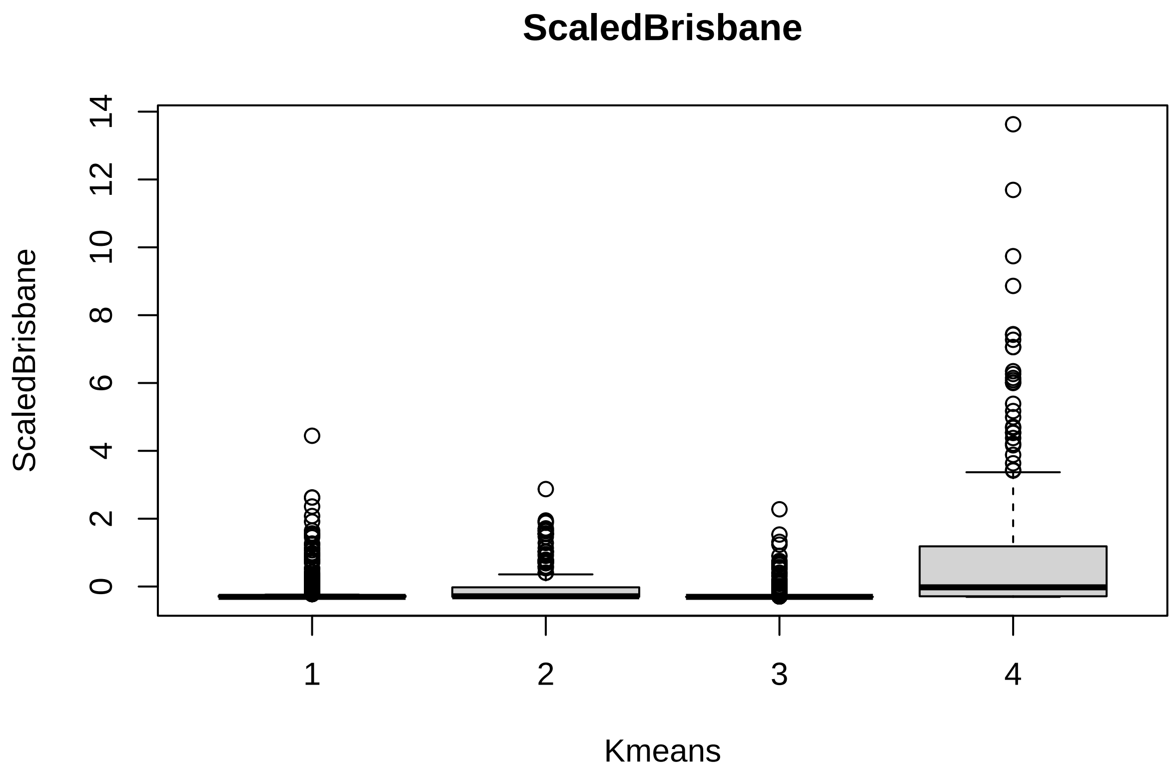

ScaledBrisbane
ScaledBrisbane
−3
−2
−1
0
1
2
3
1
2
3
4
Kmeans

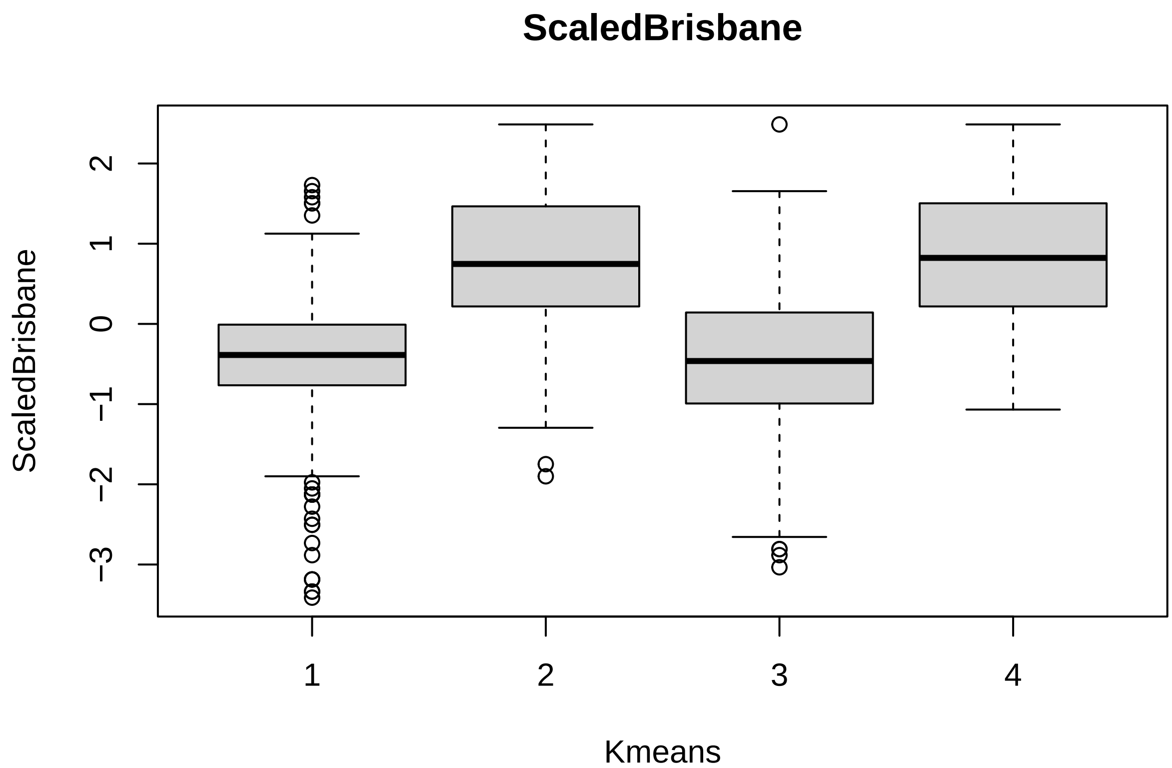

ScaledBrisbane
ScaledBrisbane
3
2
1
0
−1
−2
1
2
3
4
Kmeans

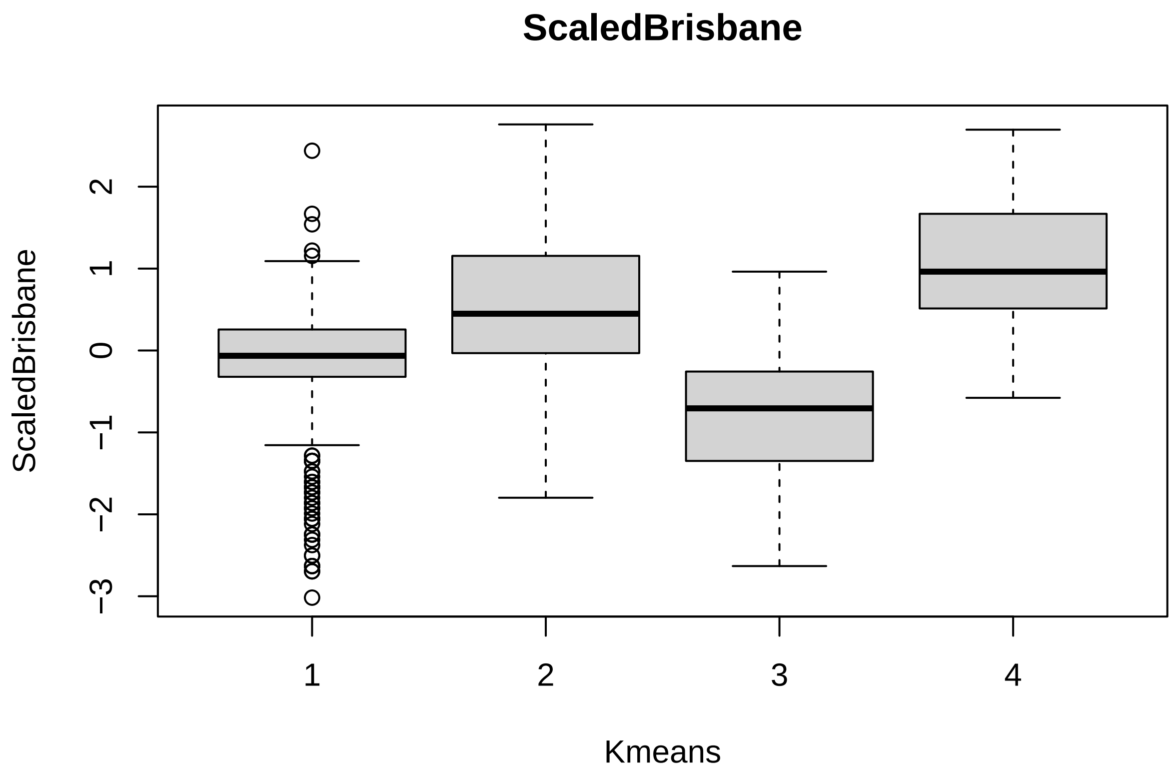

ScaledBrisbane
ScaledBrisbane
−2
−1
0
1
1
2
3
4
Kmeans

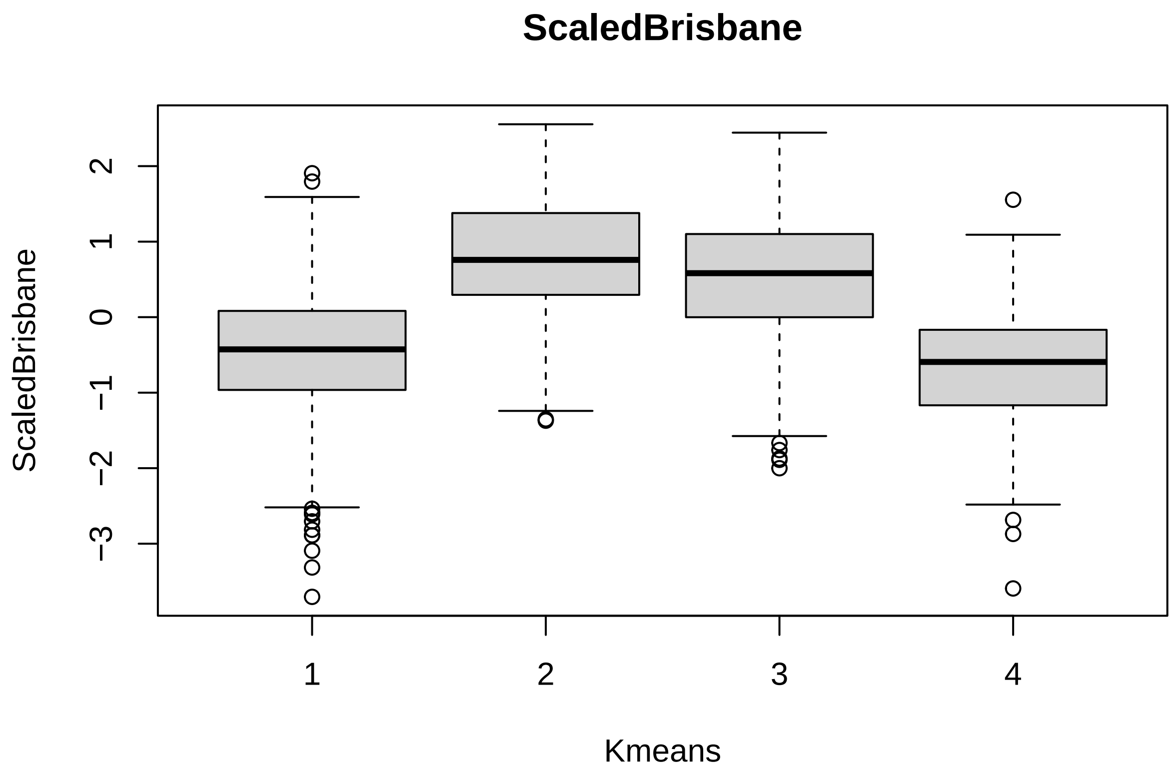

ScaledBrisbane
ScaledBrisbane
1.5
1.0
0.5
0.0
−0.5
−1.5
1
2
3
4
Kmeans

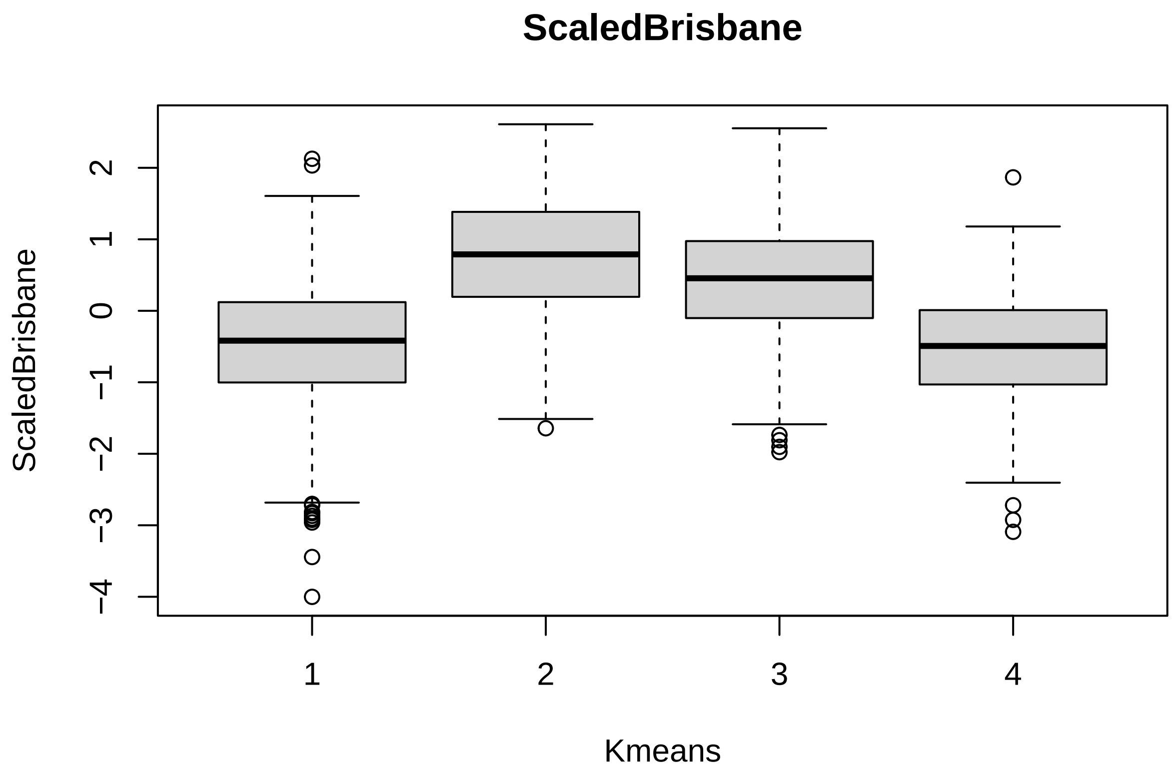

ScaledBrisbane
ScaledBrisbane
1.5
1.0
0.5
−0.5
−1.5
1
2
3
4
Kmeans

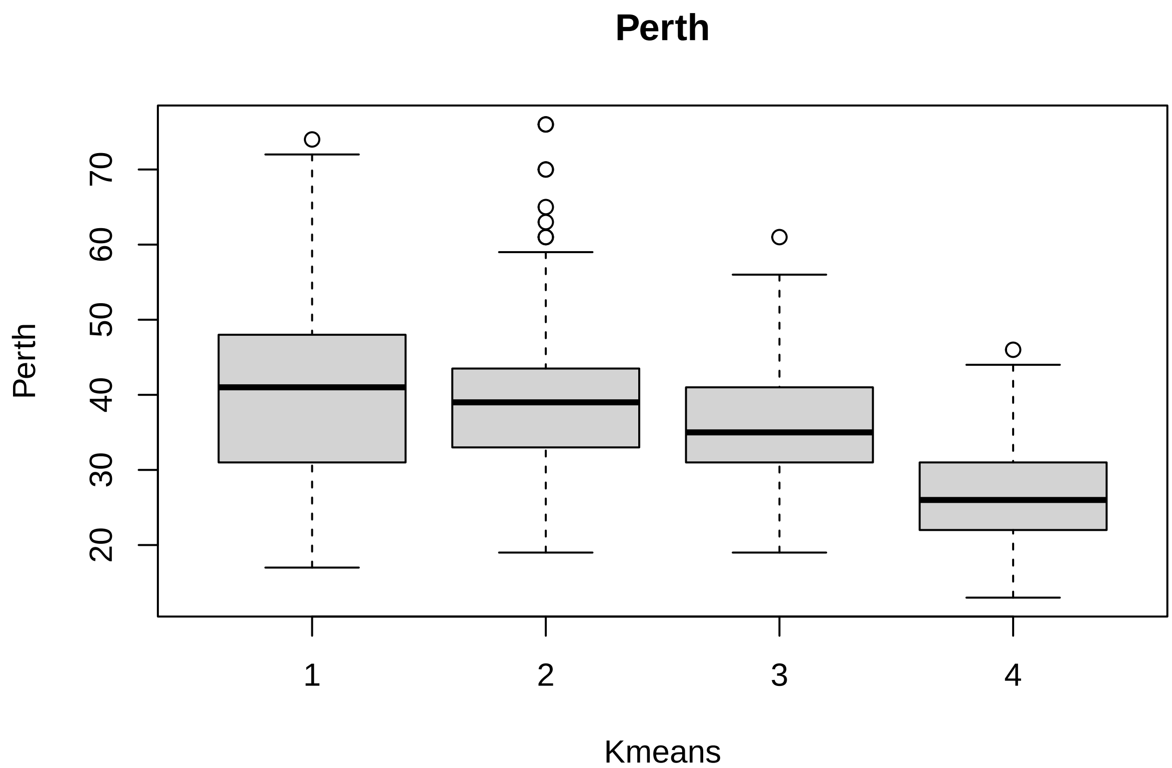

ScaledBrisbane
ScaledBrisbane
0
2
4
6
8
10
12
14
1
2
3
4
Kmeans

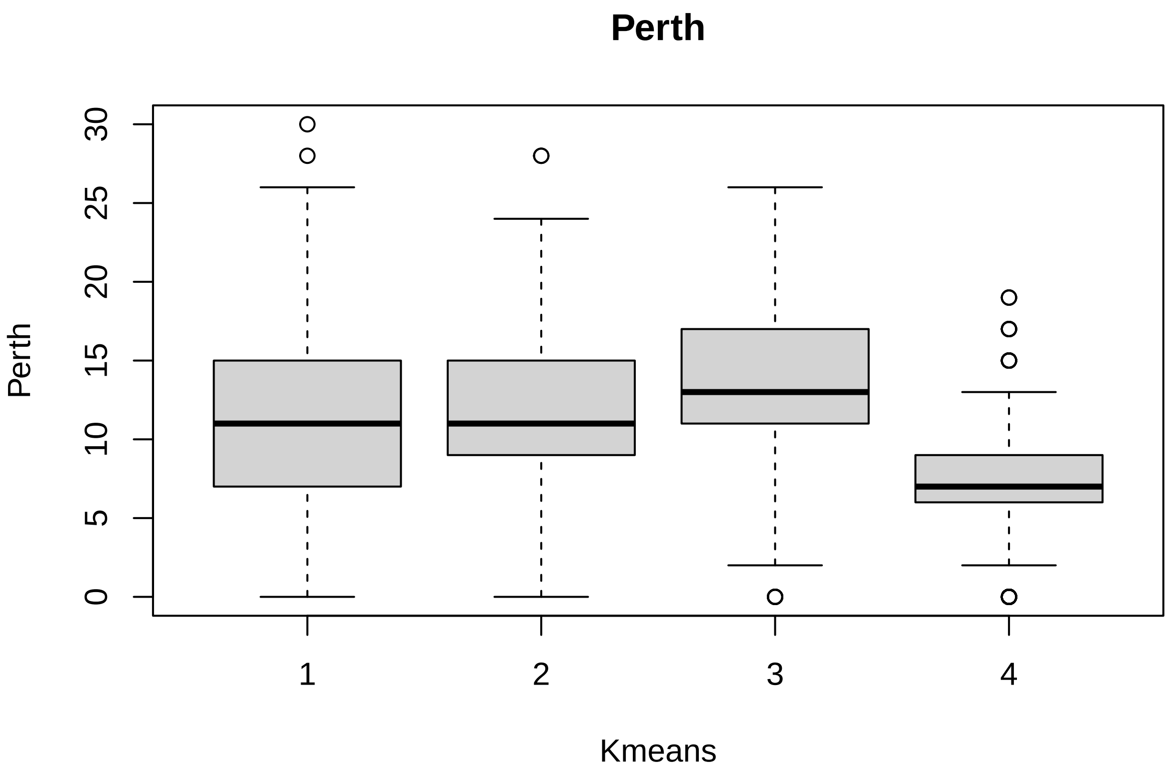

ScaledBrisbane
ScaledBrisbane
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

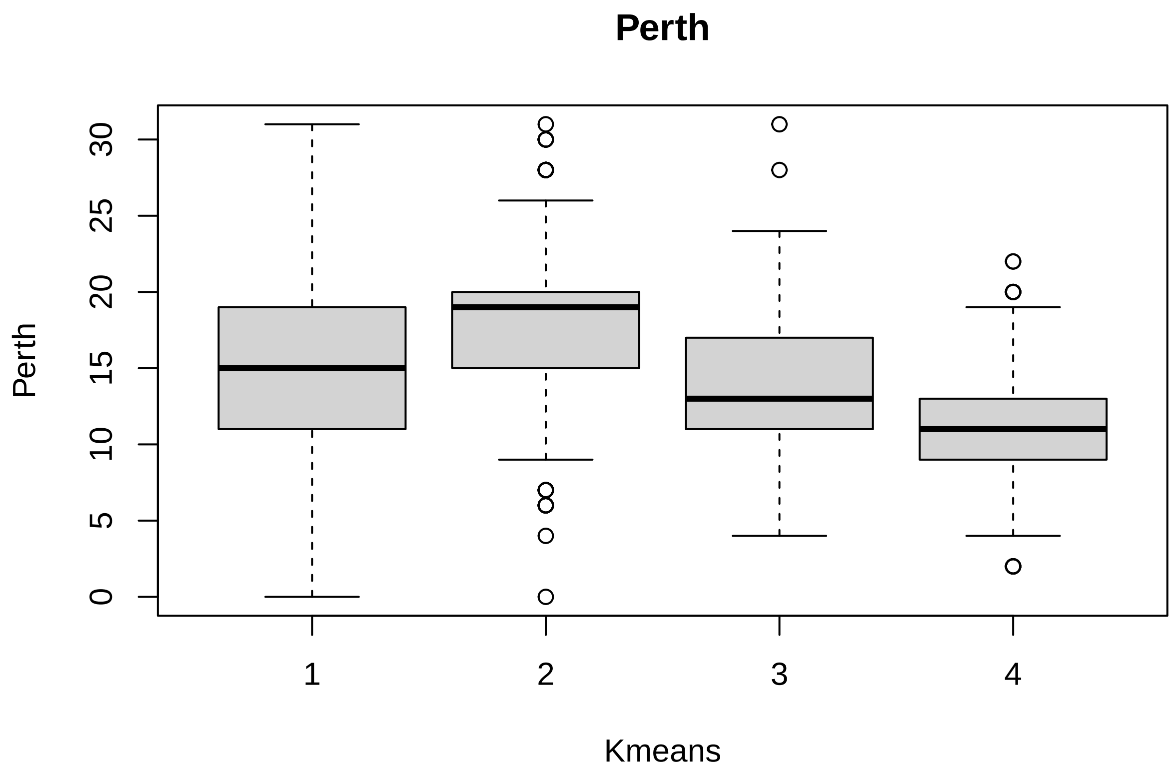

ScaledBrisbane
ScaledBrisbane
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

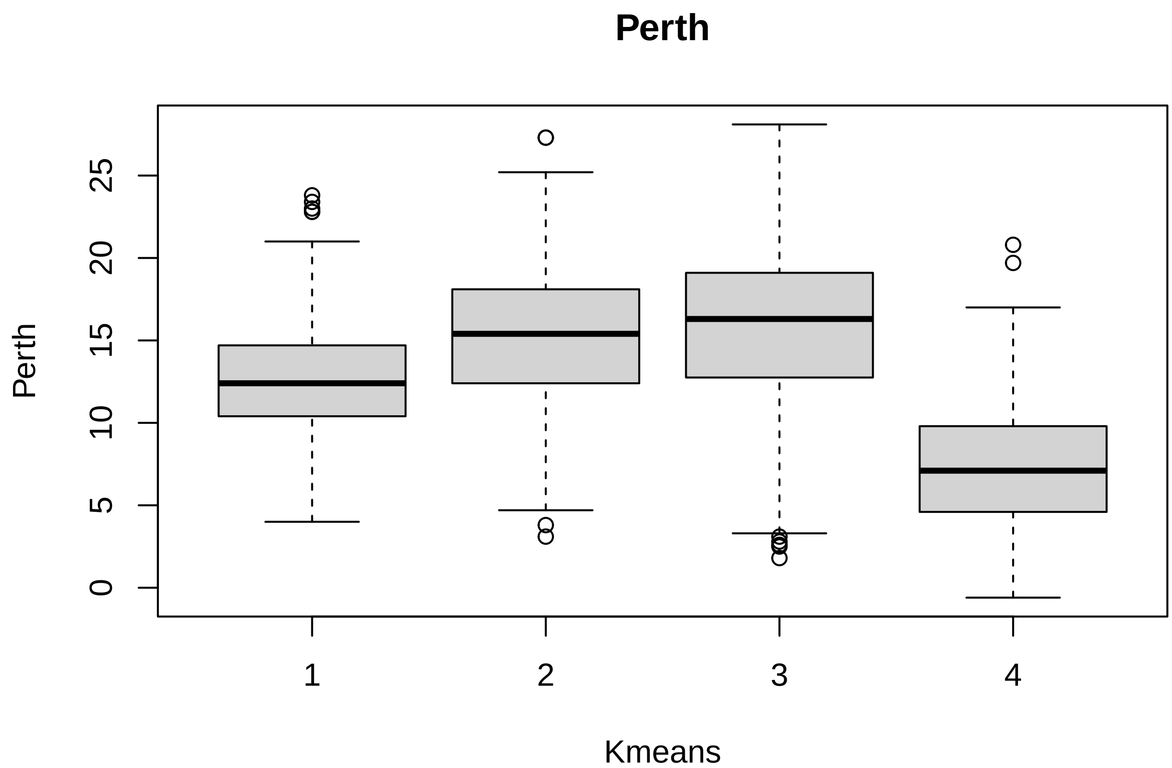

ScaledBrisbane
ScaledBrisbane
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

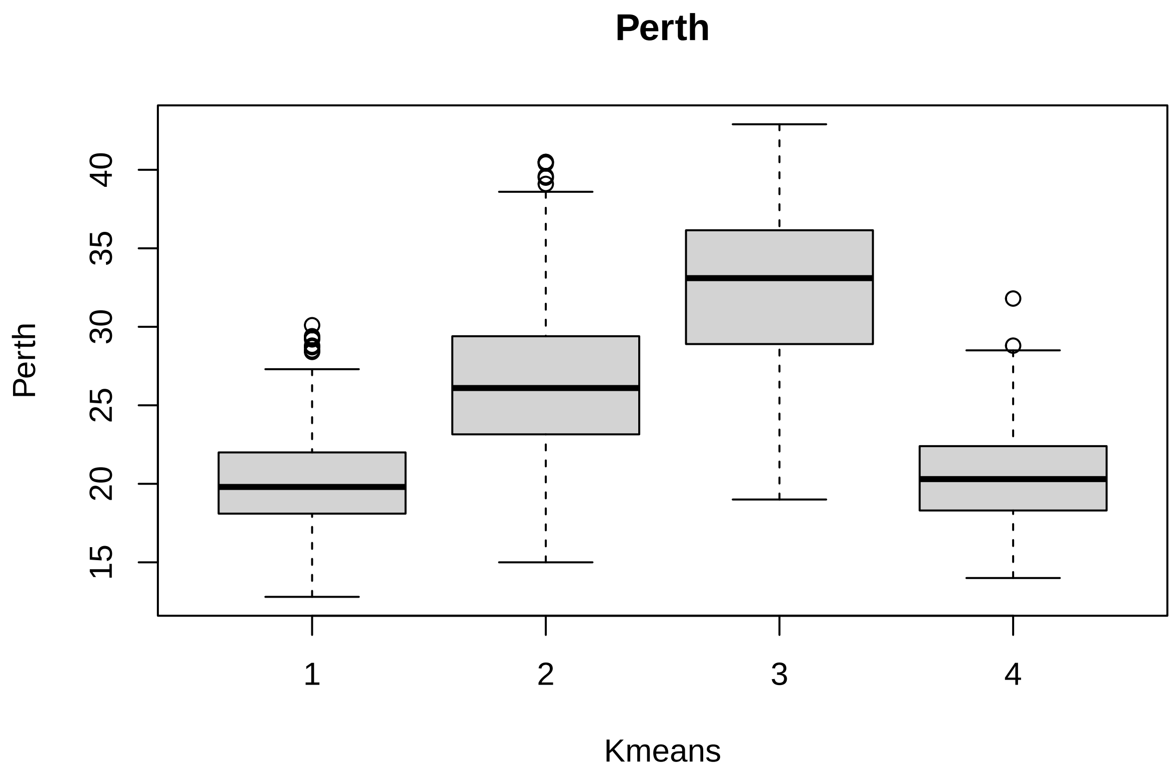

ScaledBrisbane
ScaledBrisbane
Kmeans
1
2
3
4
−4
−3
−2
−1
0
1
2

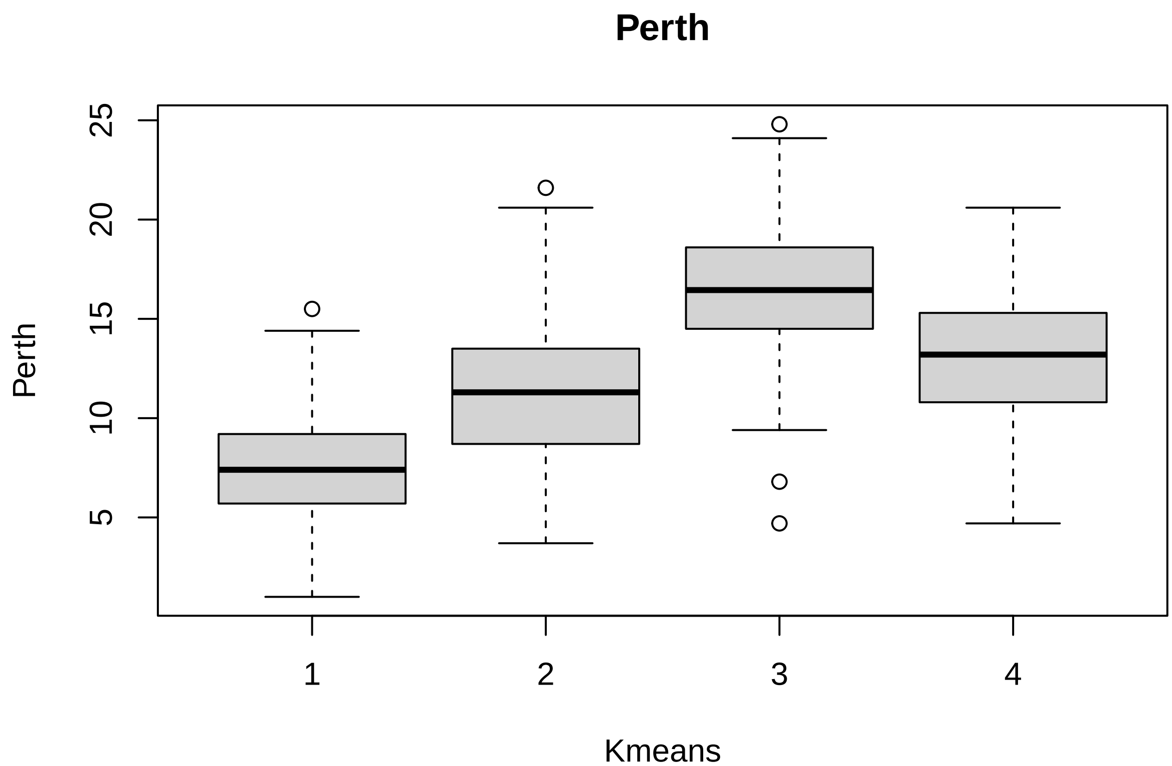

Perth
Perth
20
30
40
50
60
70
1
2
3
4
Kmeans

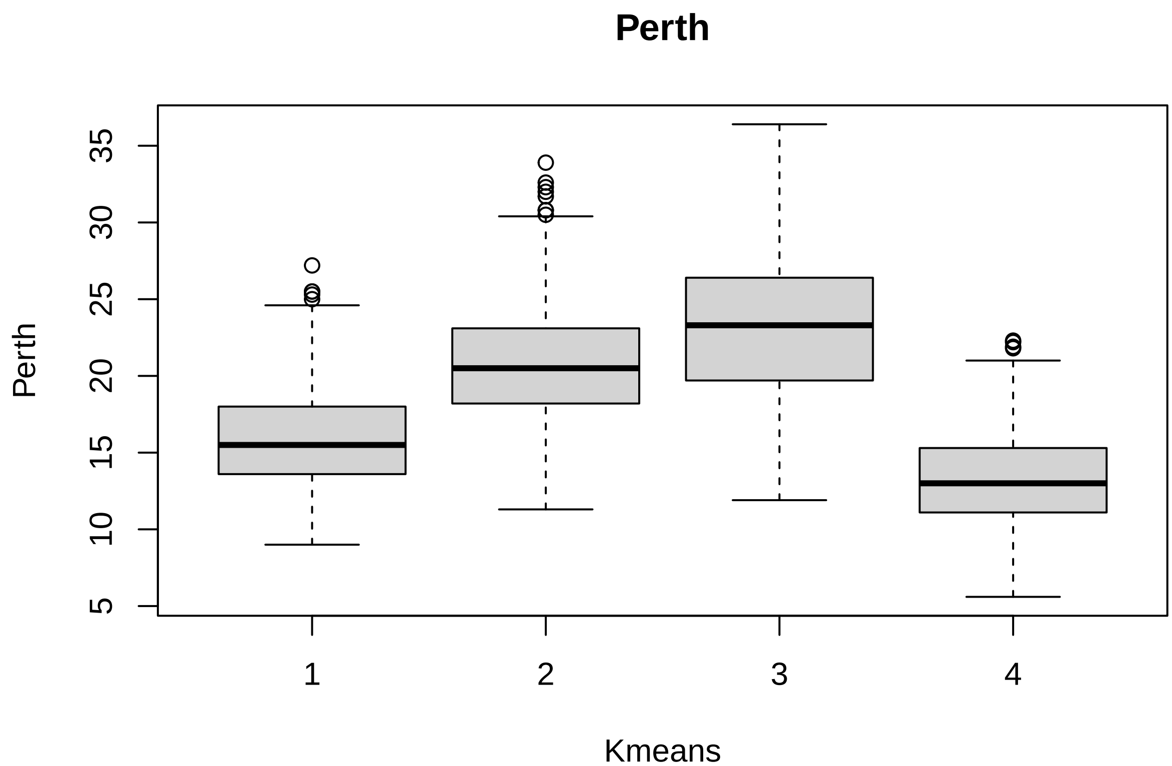

Perth
Perth
0
5
10
15
20
25
30
1
2
3
4
Kmeans

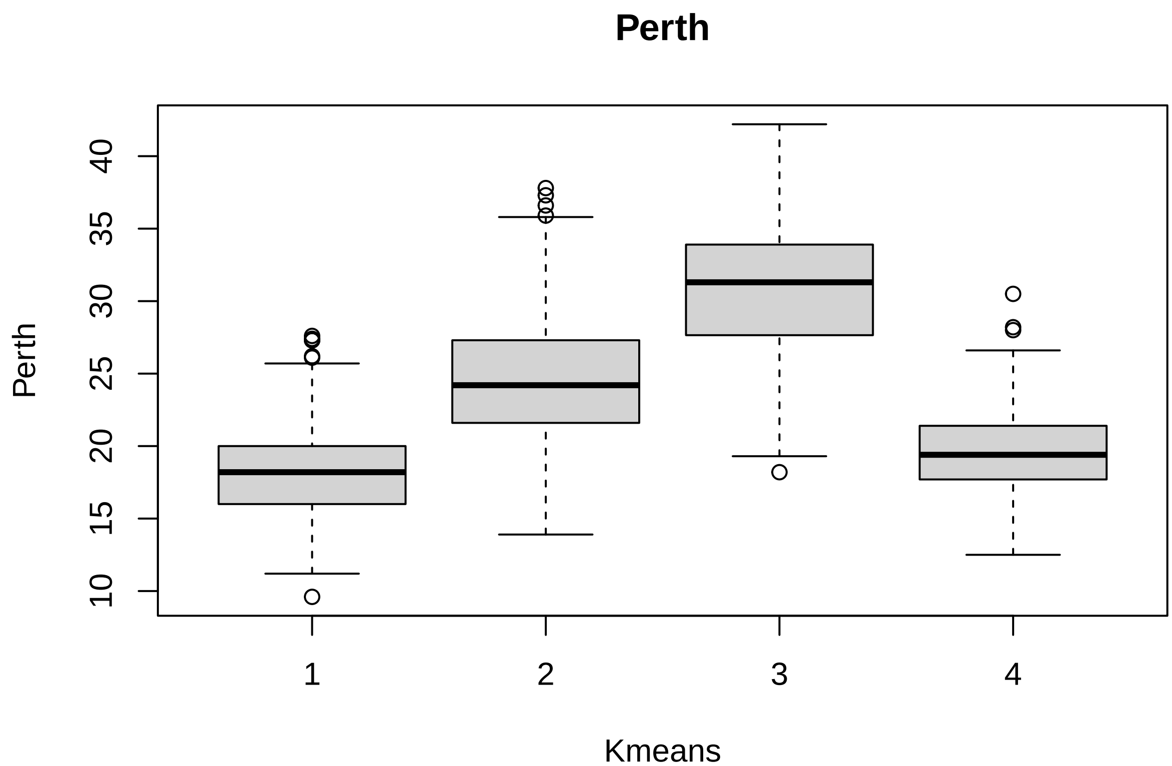

Perth
Perth
0
5
10
15
20
25
30
1
2
3
4
Kmeans

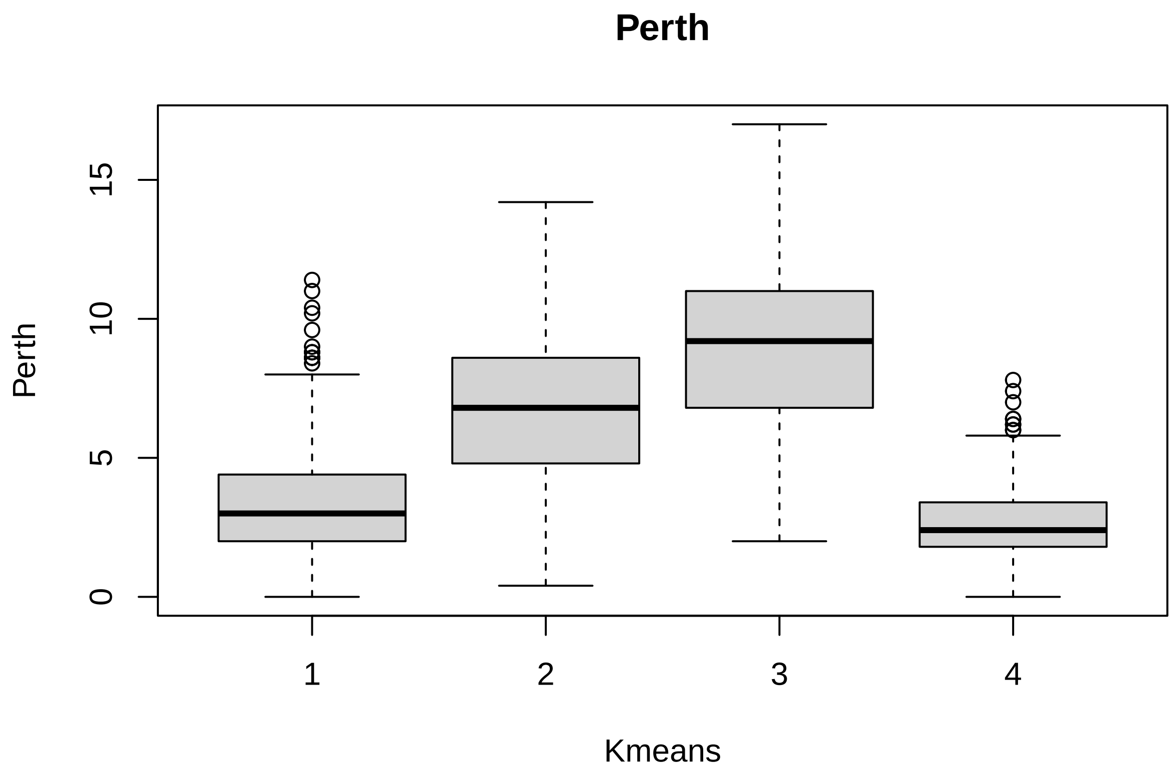

Perth
Perth
0
5
10
15
20
25
1
2
3
4
Kmeans

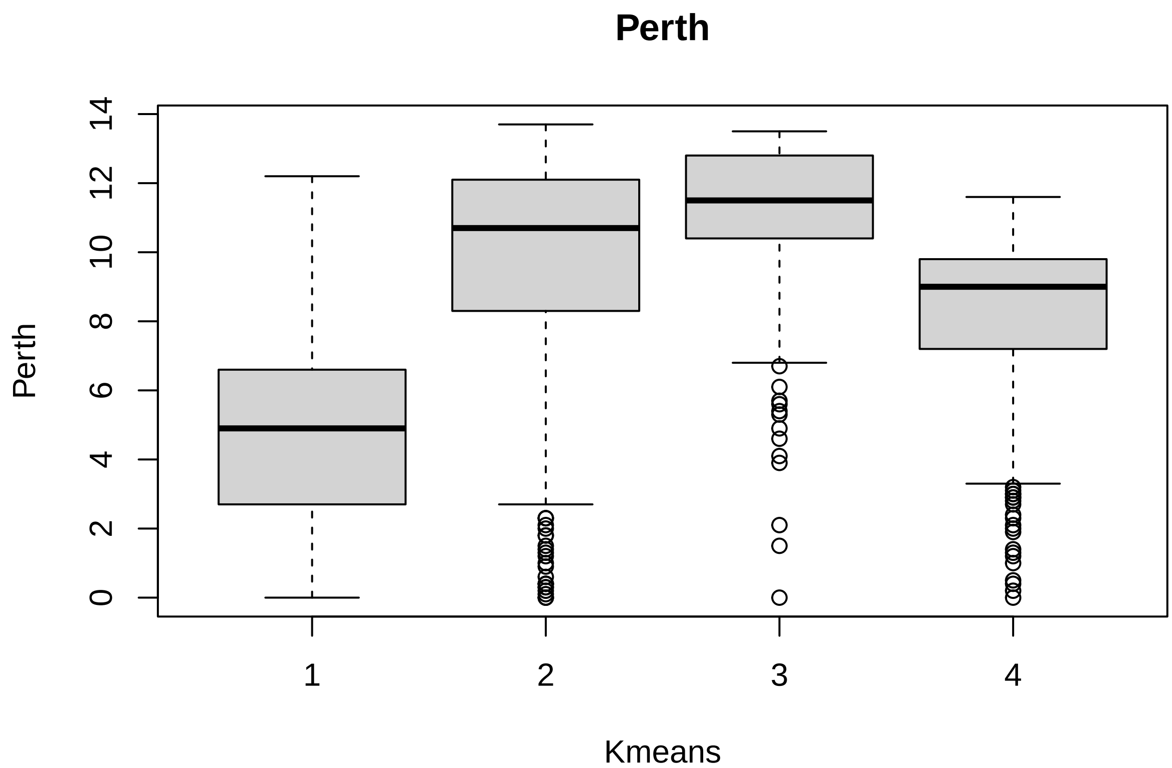

Perth
Perth
15
20
25
30
35
40
1
2
3
4
Kmeans

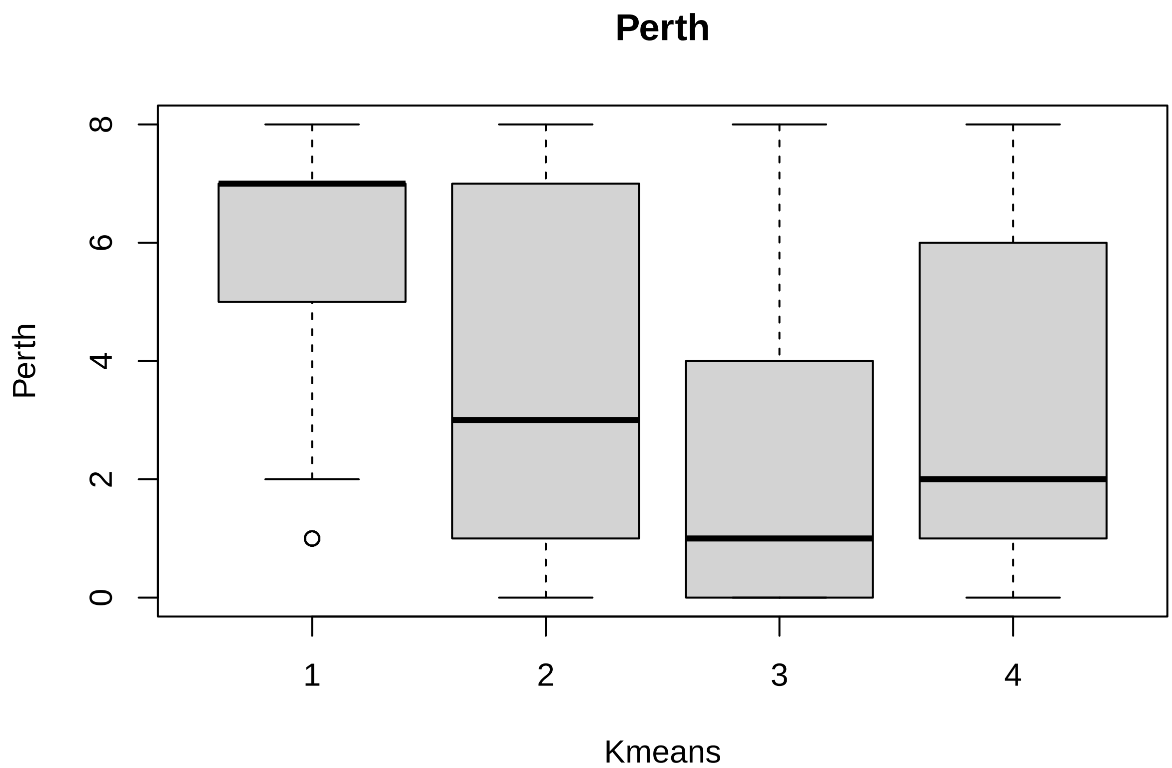

Perth
25
20
15
10
5
Perth
1
2
3
4
Kmeans

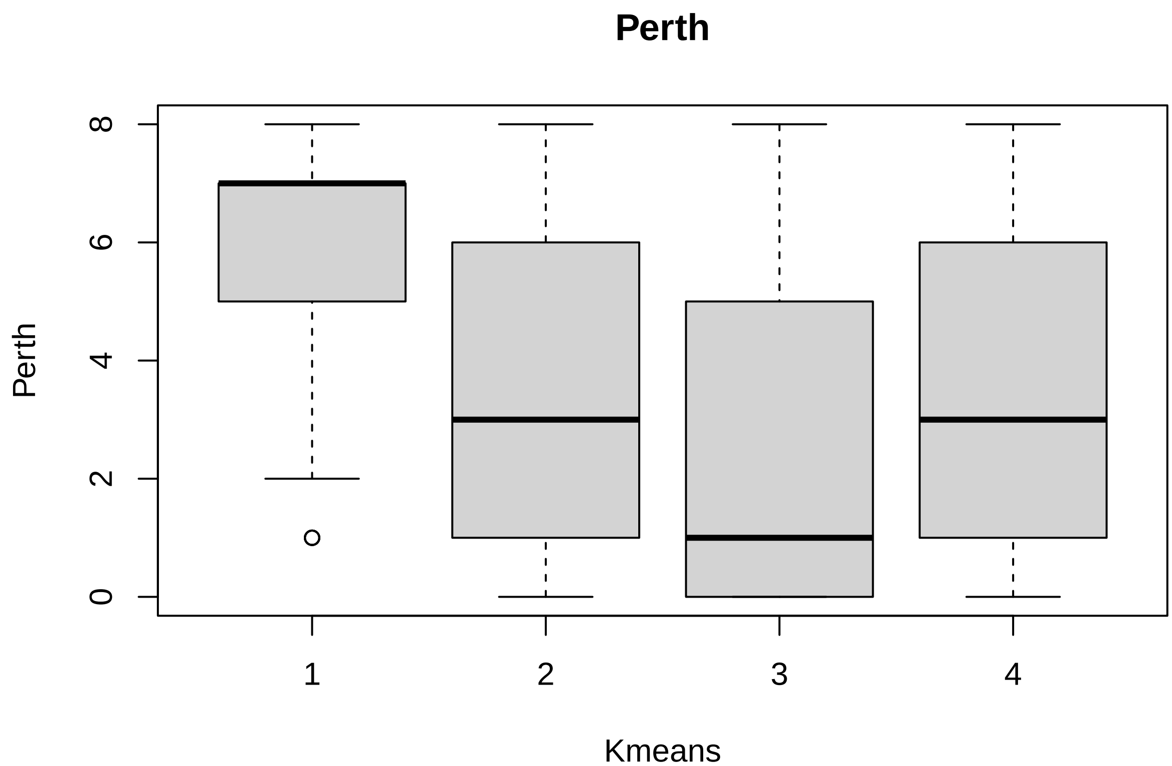

Perth
Perth
35
30
25
20
15
10
5
1
2
3
4
Kmeans

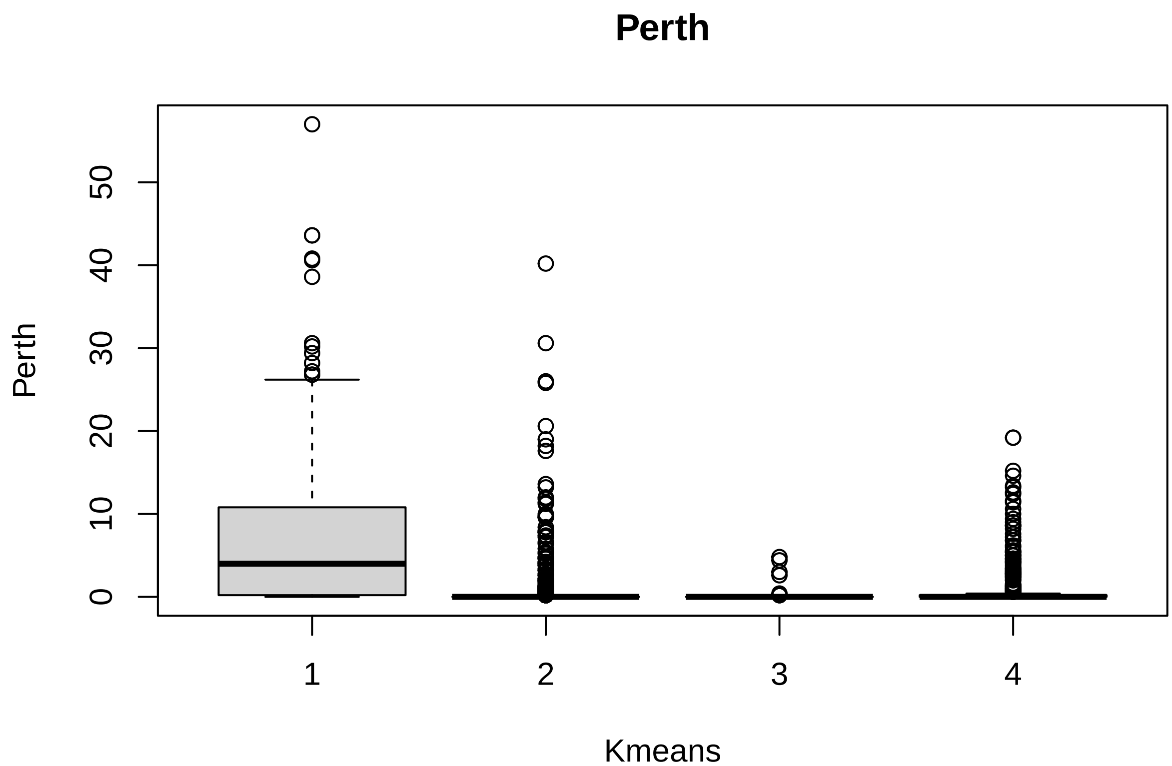

Perth
Perth
40
35
30
25
20
15
10
1
2
3
4
Kmeans


Perth
Perth
0
5
10
15
1
2
3
4
Kmeans

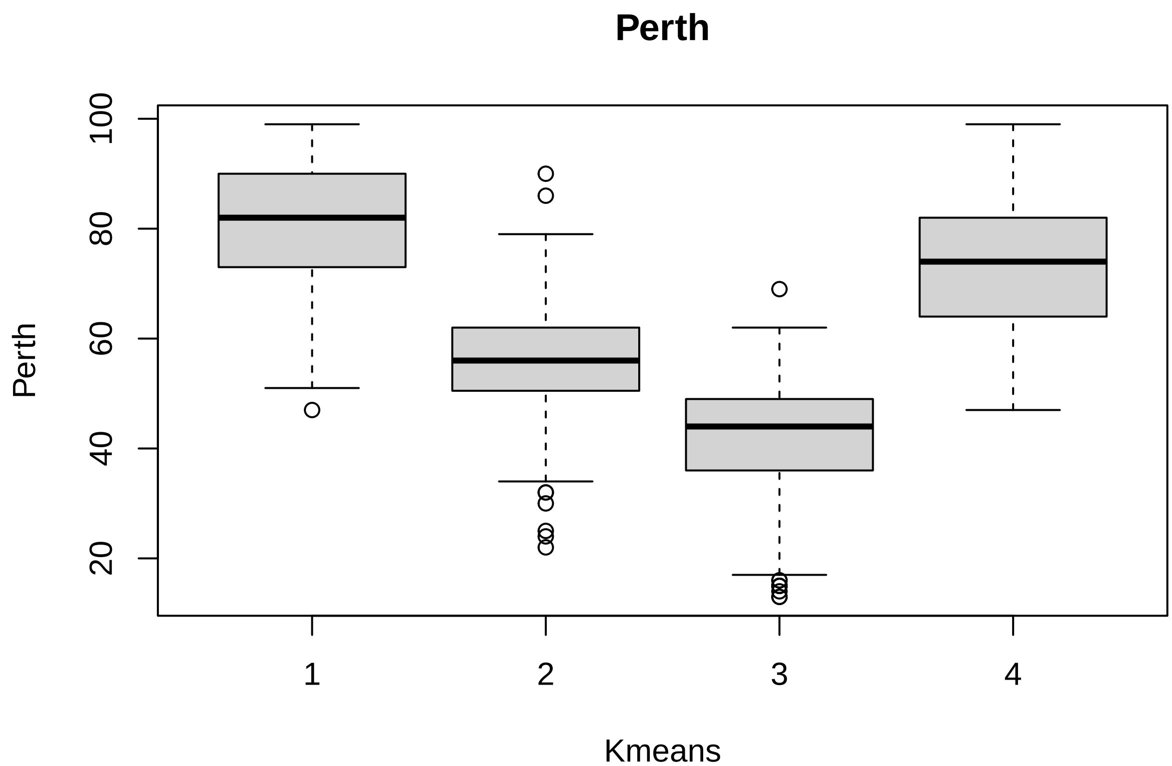

Perth
Perth
0
2
4
6
8
10
12
14
1
2
3
4
Kmeans

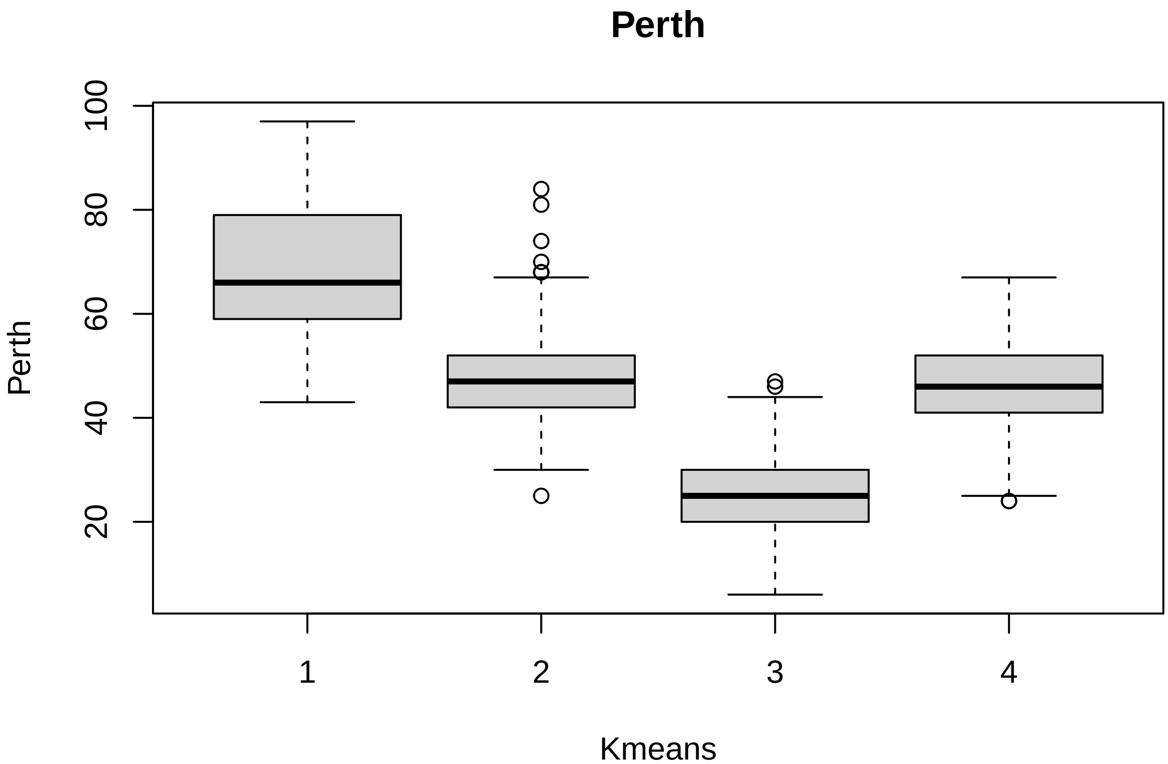

Perth
Perth
0
2
4
6
8
1
2
3
4
Kmeans

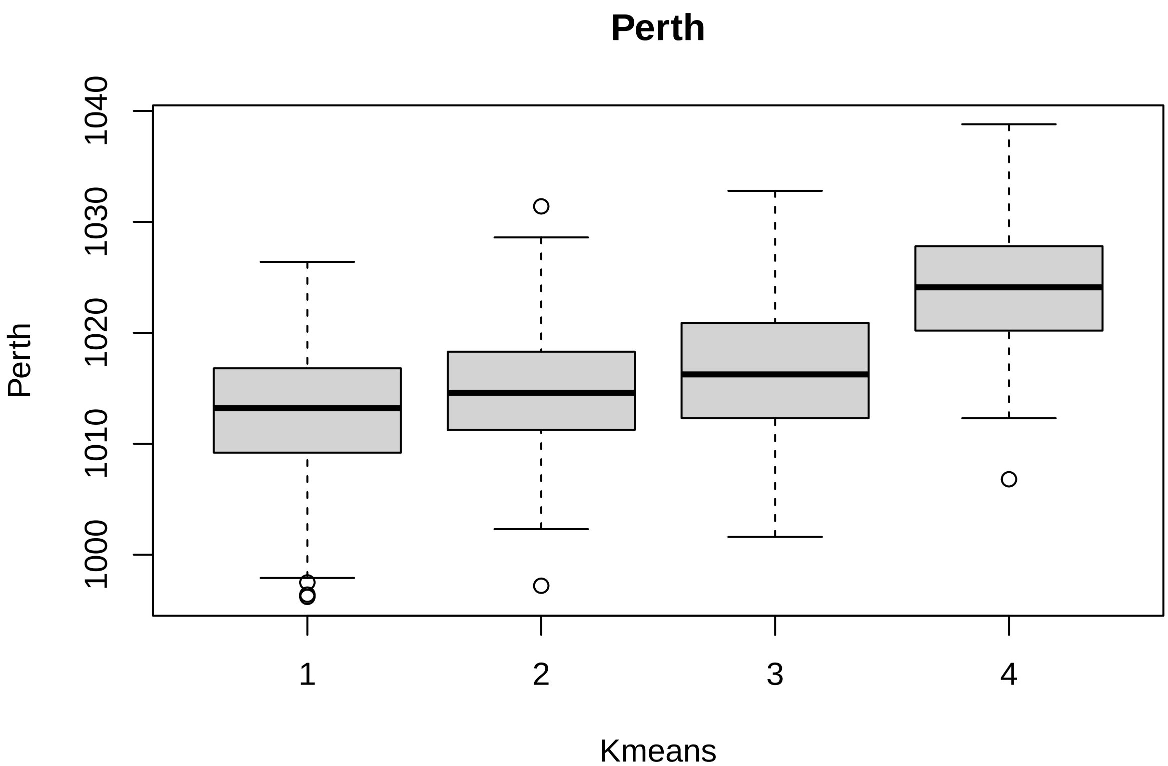

Perth
Perth
0
2
4
6
8
1
2
3
4
Kmeans



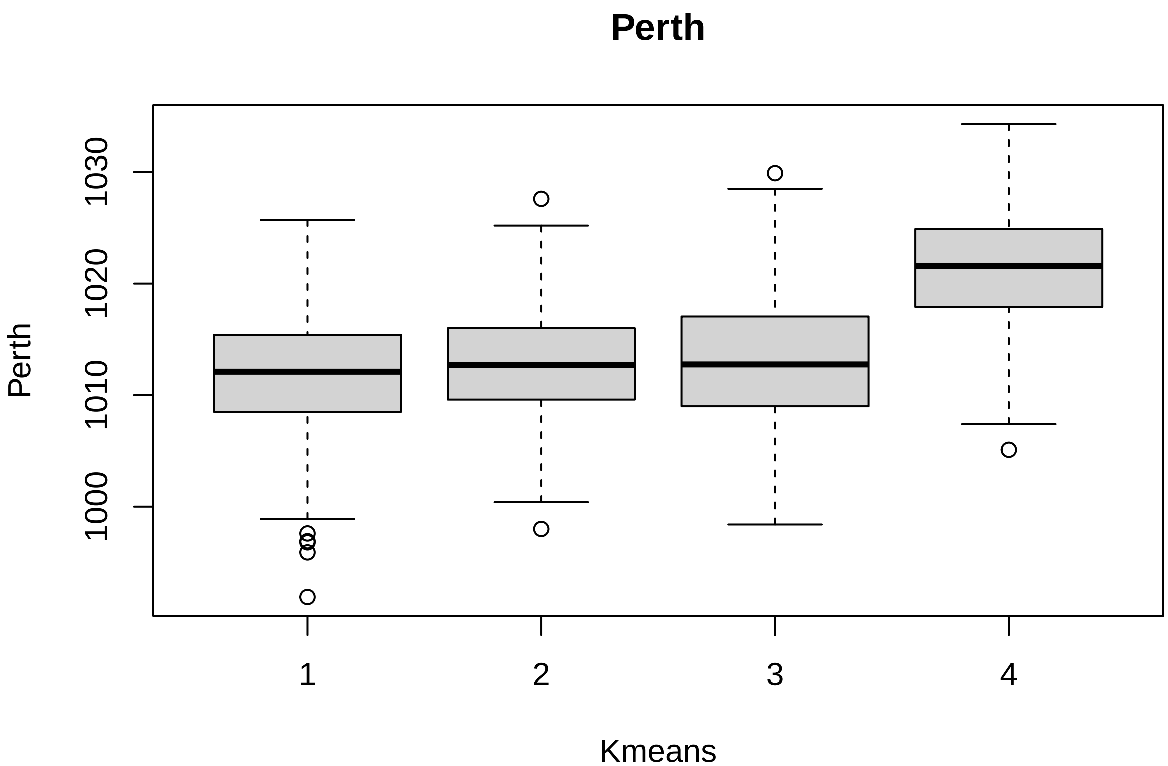
Perth
Perth
0
10
20
30
40
50
1
2
3
4
Kmeans

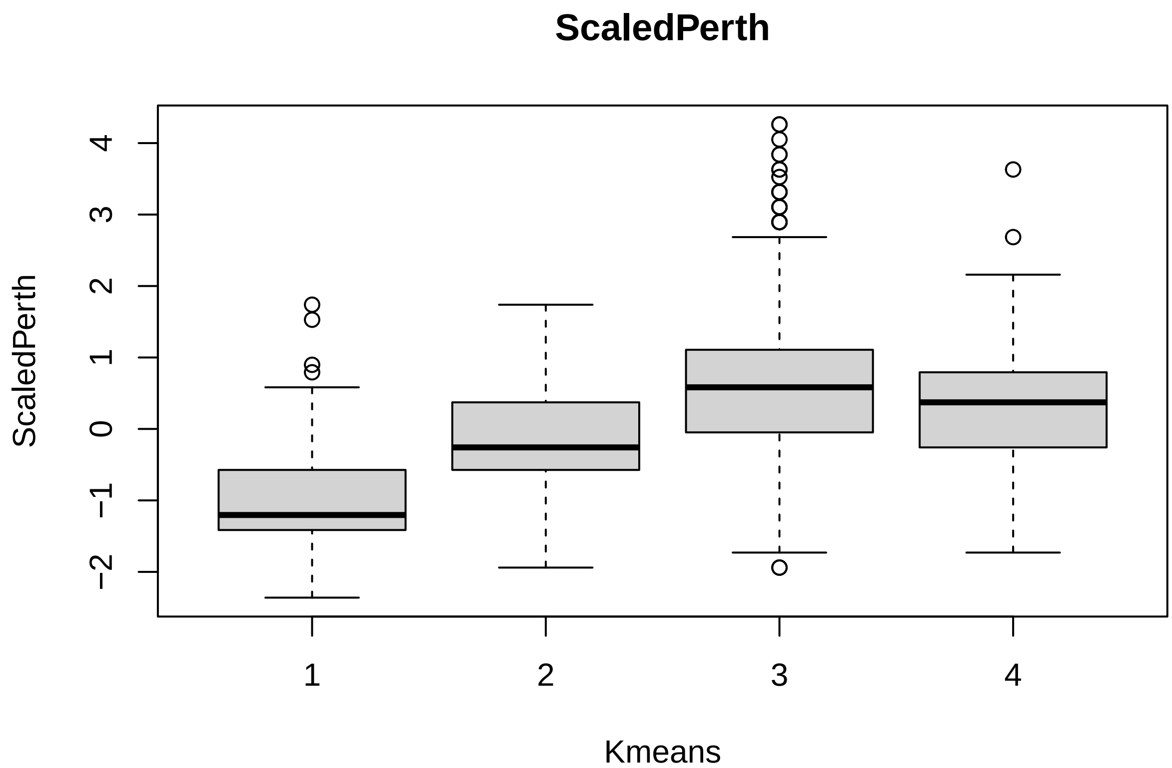

Perth
100
80
60
40
20
Perth
1
2
3
4
Kmeans


Perth
Perth
100
80
60
40
20
1
2
3
4
Kmeans

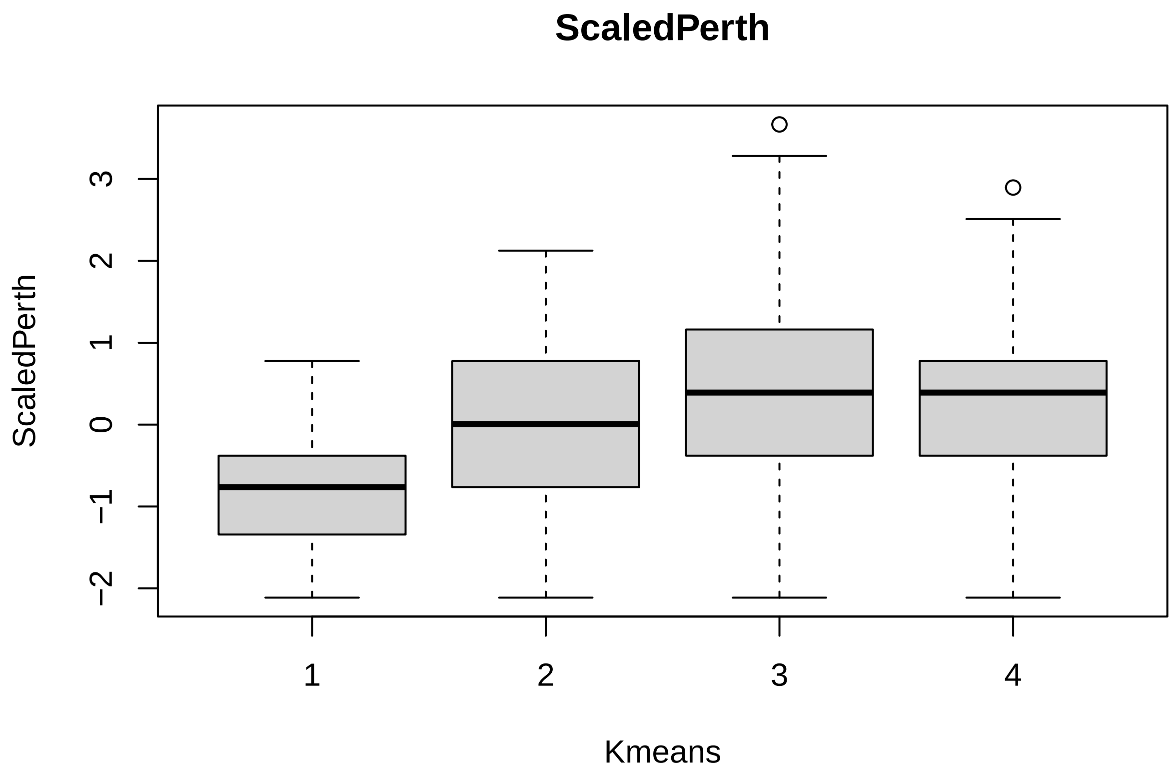


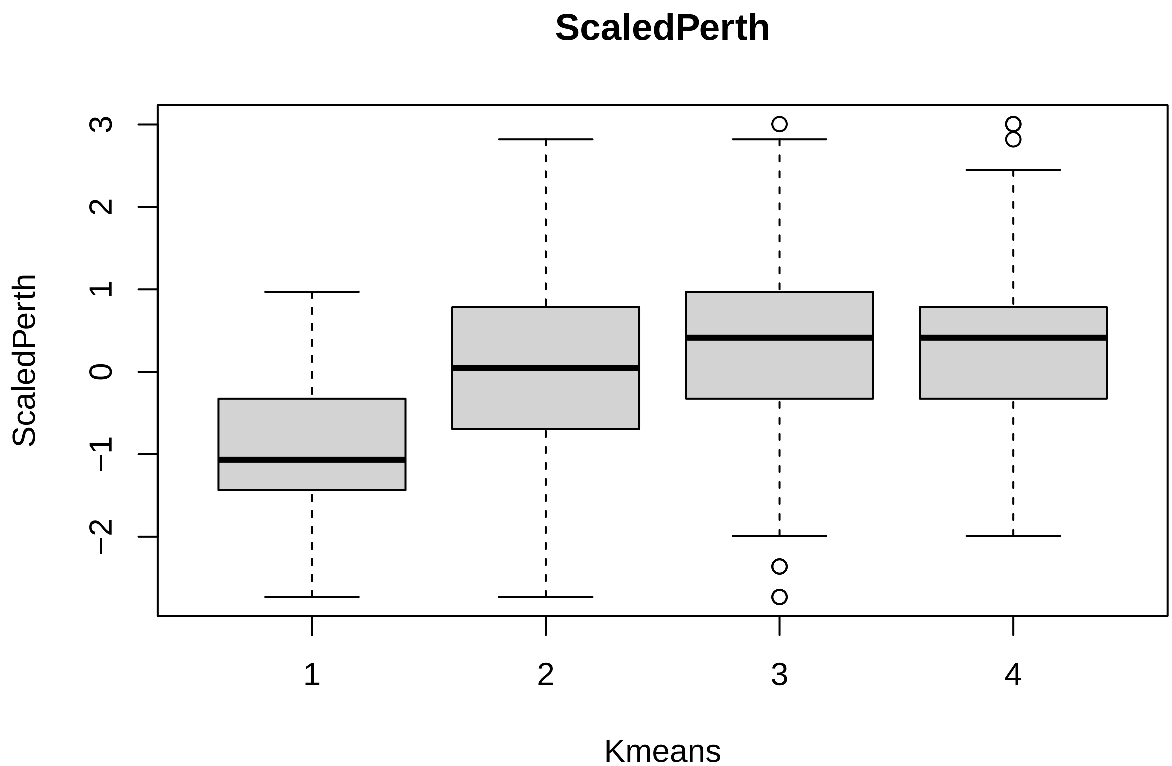
Perth
1040
1030
1020
1010
1000
Perth
1
2
3
4
Kmeans

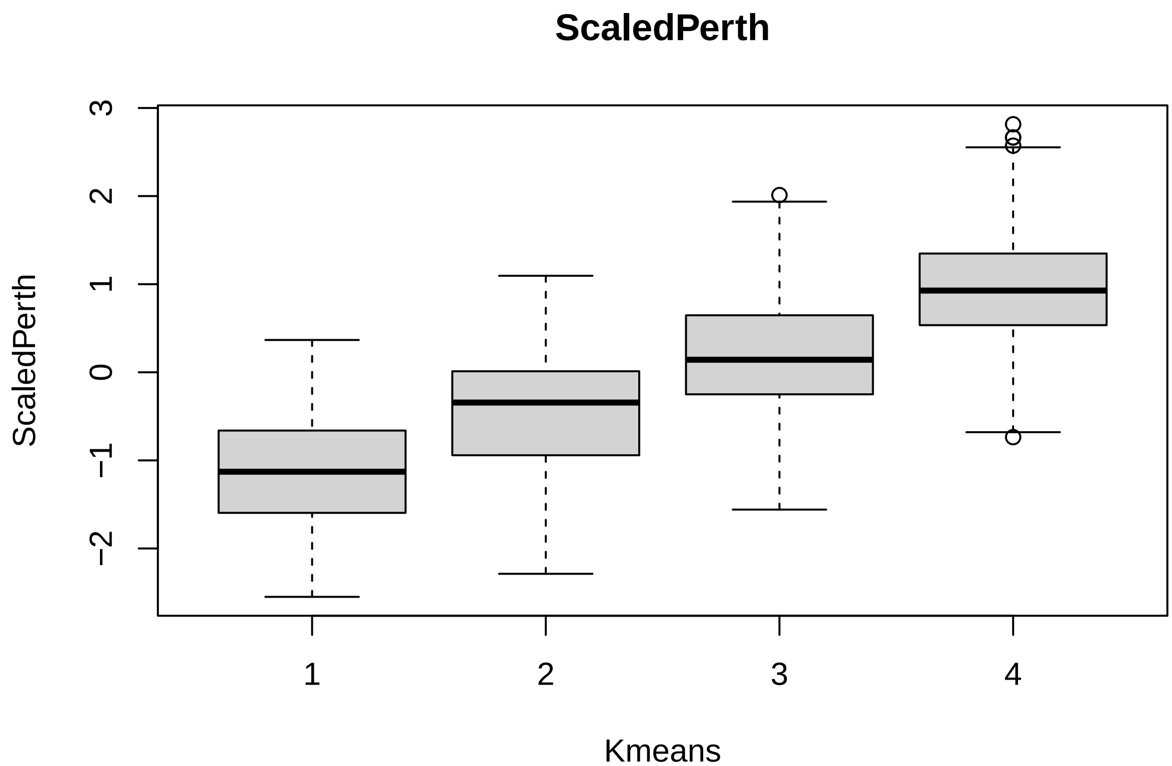

Perth
Perth
1000
1010
1020
1030
1
2
3
4
Kmeans

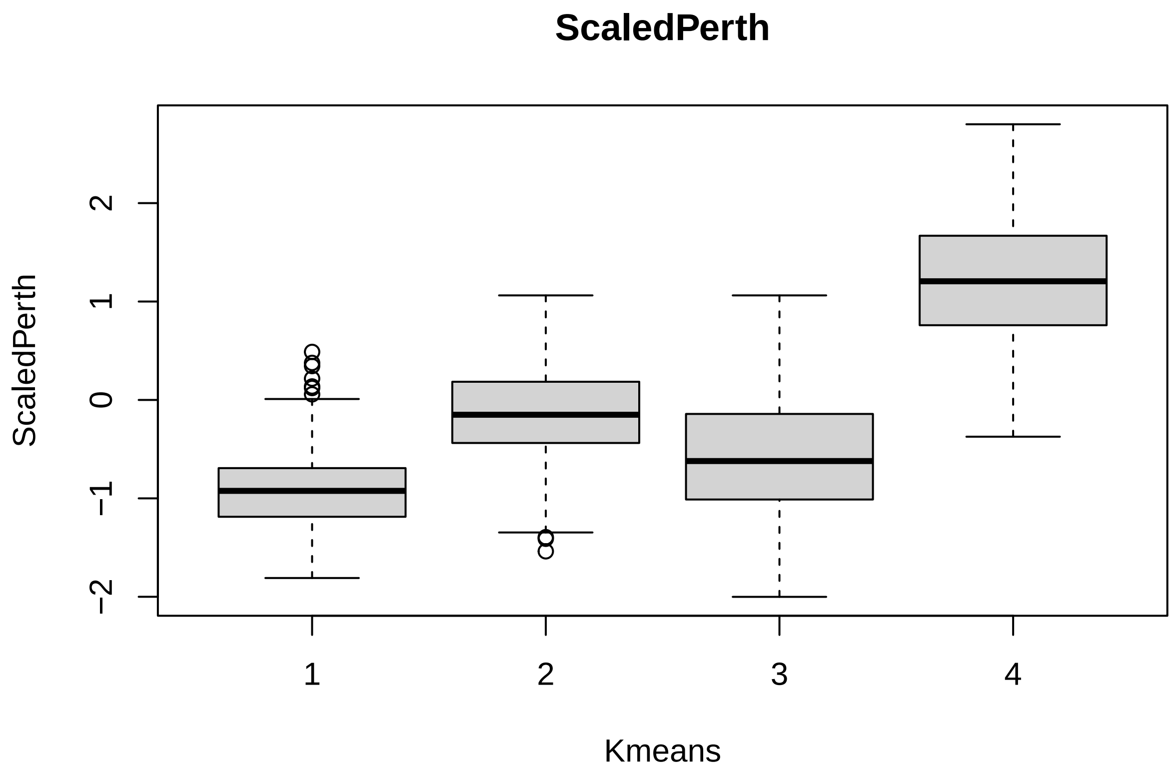


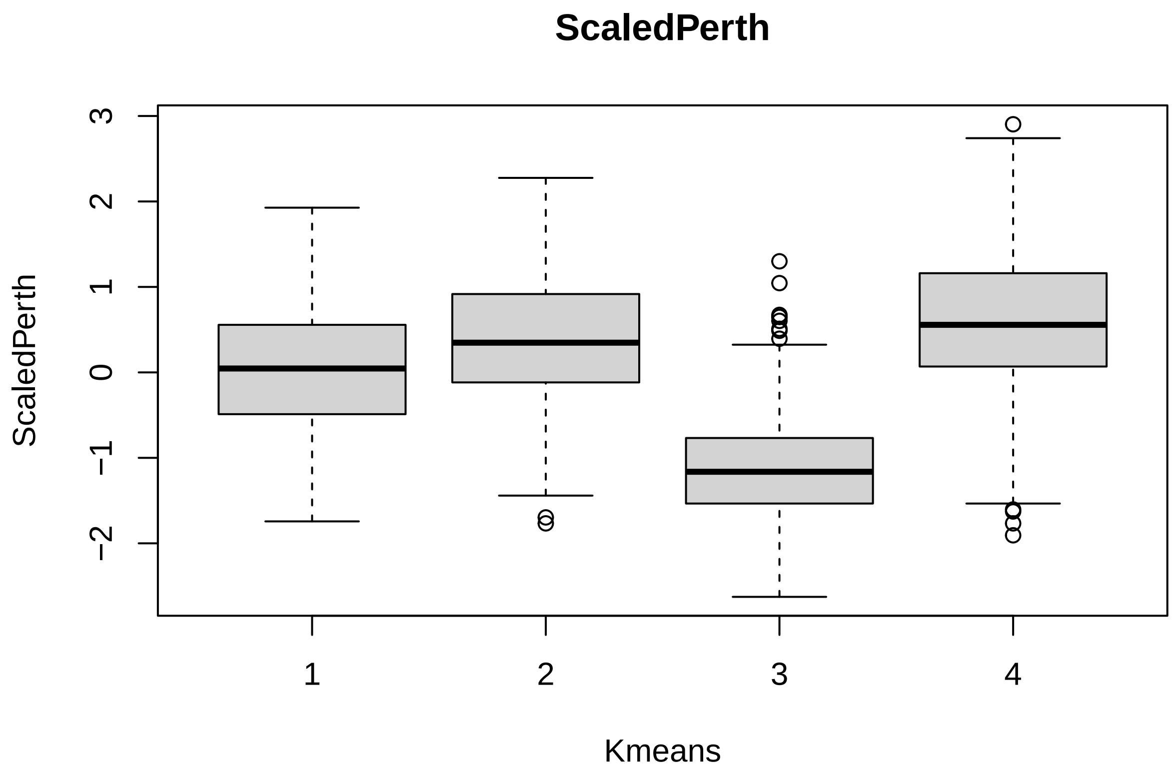
ScaledPerth
ScaledPerth
Kmeans
4
3
2
1
0
−1
−2
1
2
3
4

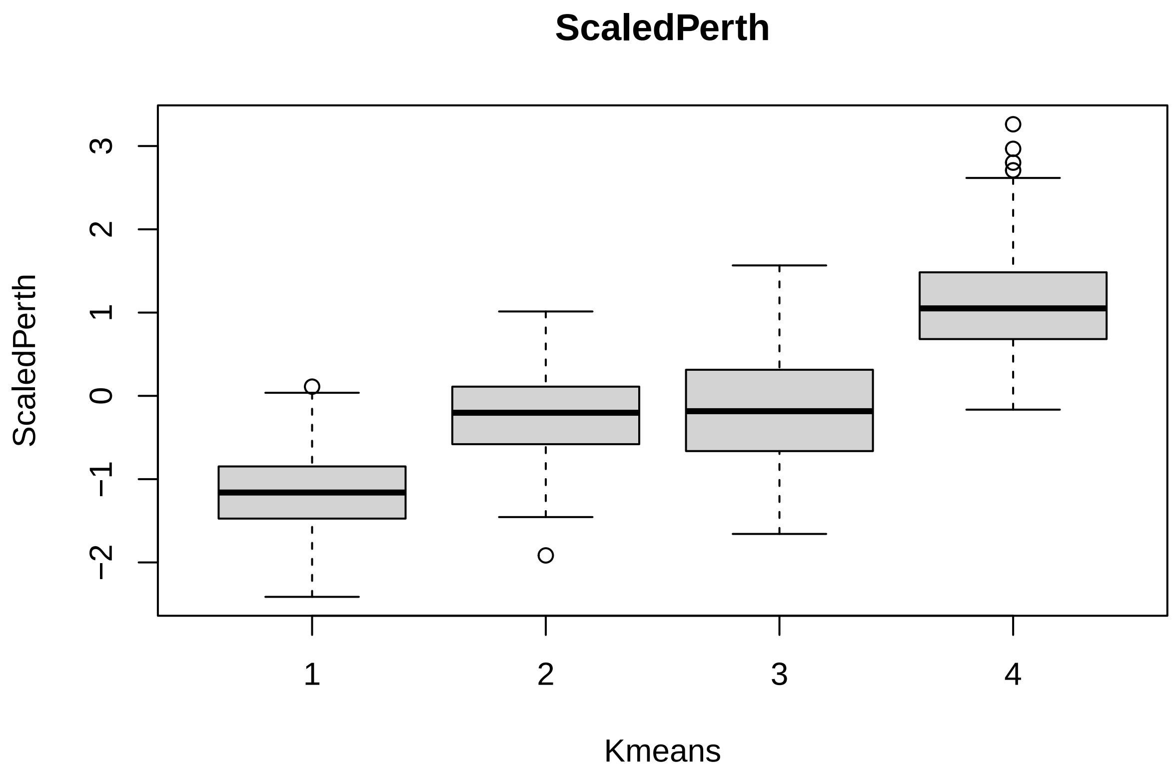

ScaledPerth
ScaledPerth
−2
−1
0
1
2
3
1
2
3
4
Kmeans

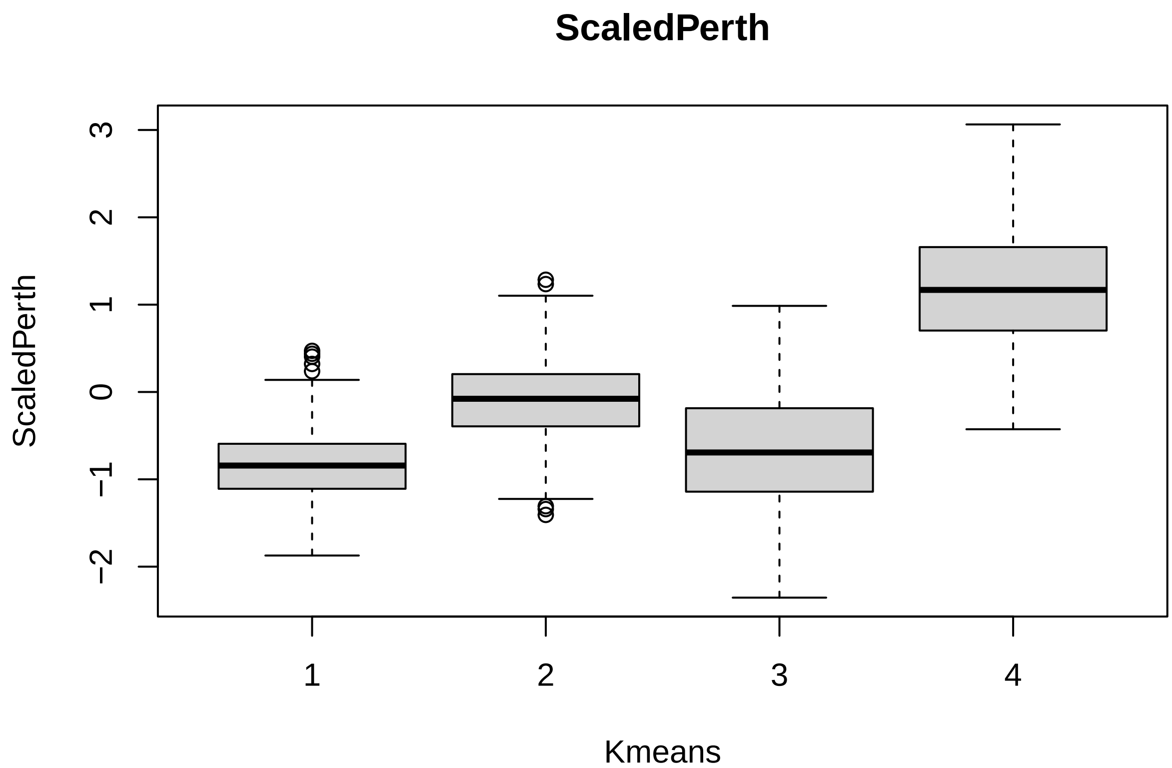

ScaledPerth
3
2
1
0
−1
−2
ScaledPerth
1
2
3
4
Kmeans

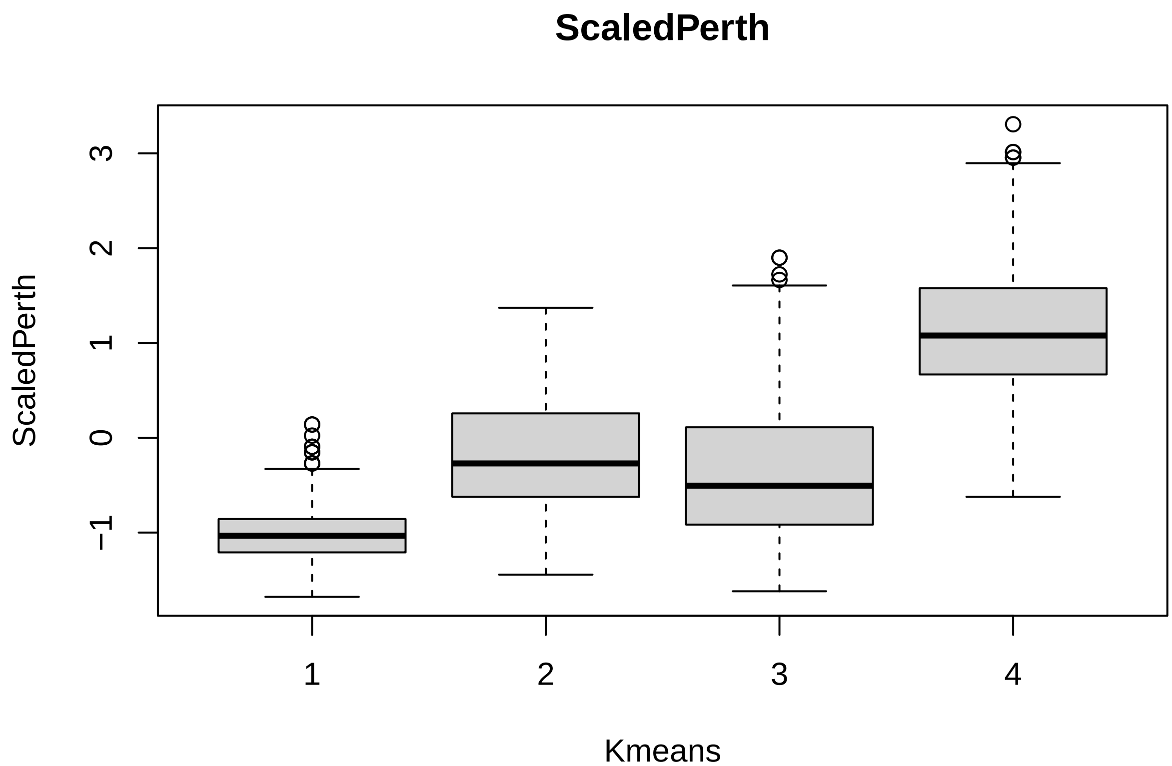

ScaledPerth
ScaledPerth
3
2
1
0
−1
−2
1
2
3
4
Kmeans

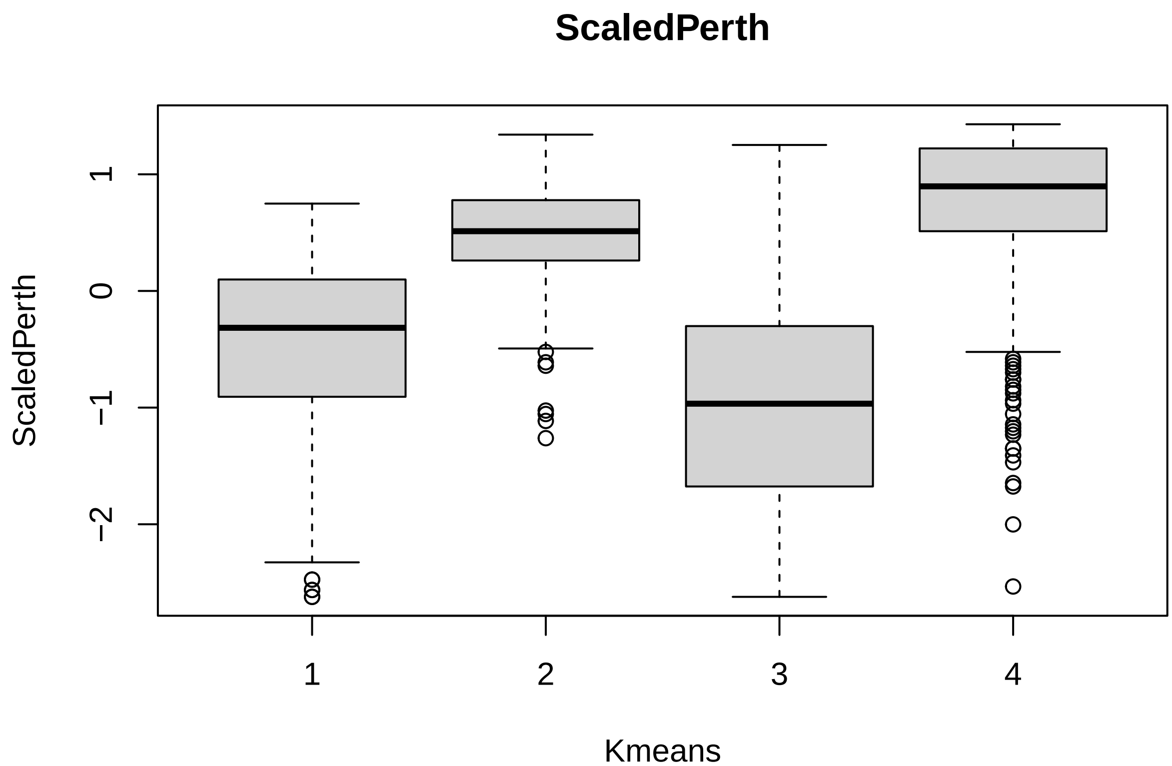

ScaledPerth
ScaledPerth
2
1
0
−1
−2
1
2
3
4
Kmeans

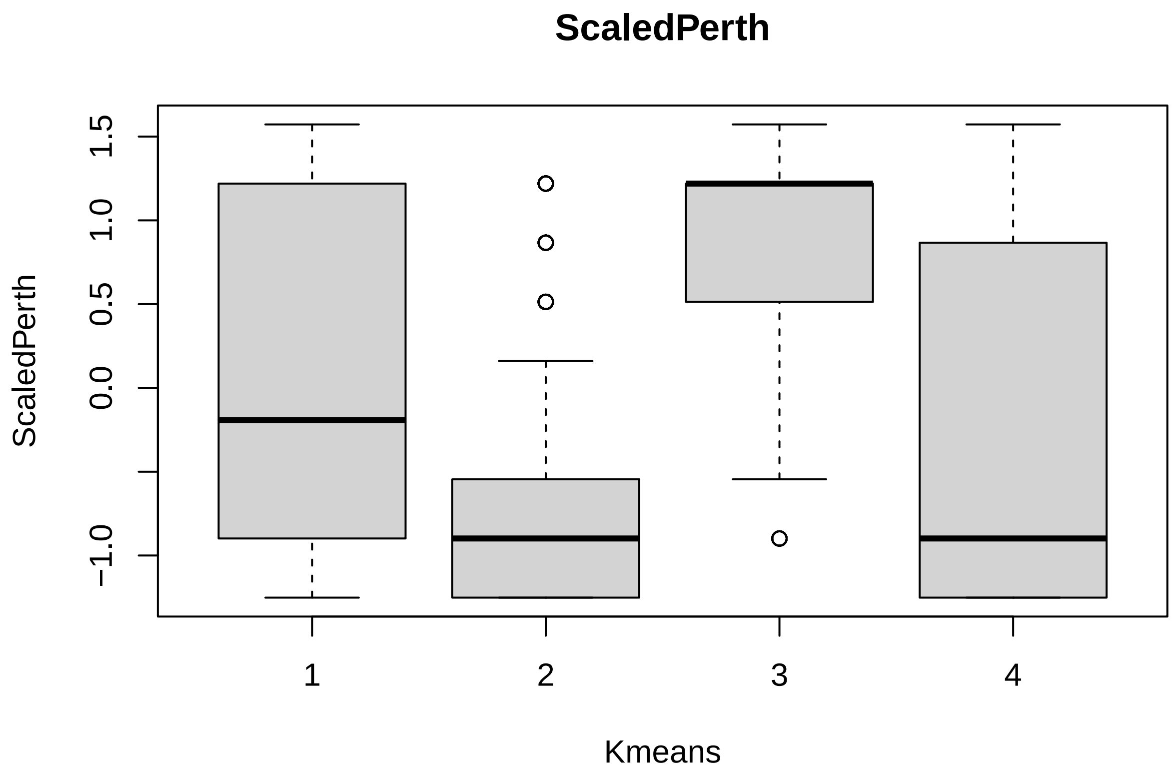

ScaledPerth
ScaledPerth
3
2
1
0
−1
−2
1
2
3
4
Kmeans

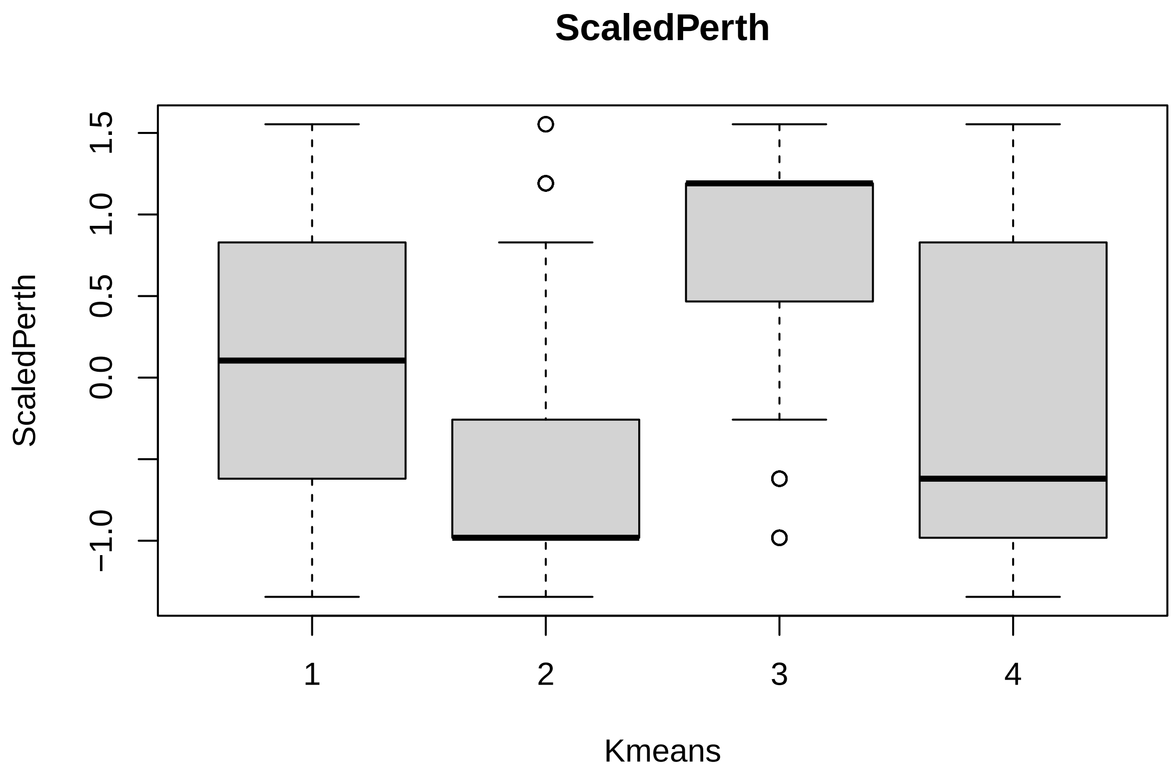

ScaledPerth
ScaledPerth
3
2
1
0
−1
−2
1
2
3
4
Kmeans

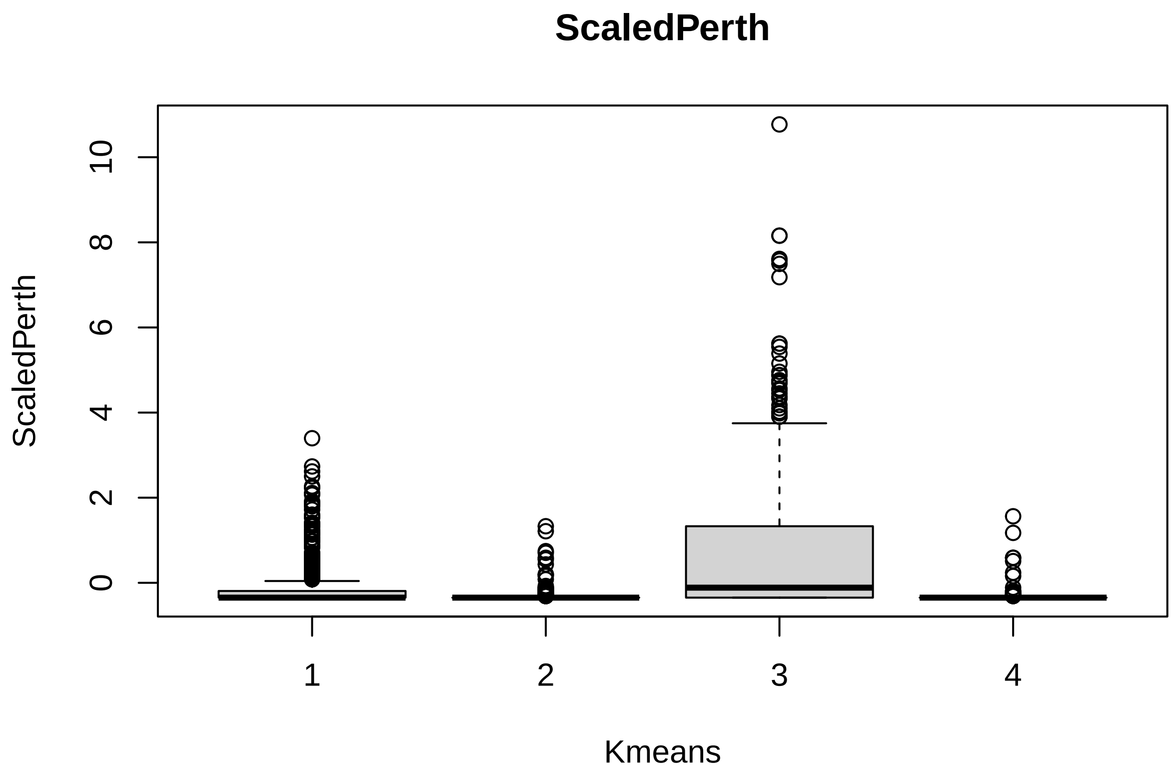

ScaledPerth
ScaledPerth
3
2
1
0
−1
−2
1
2
3
4
Kmeans

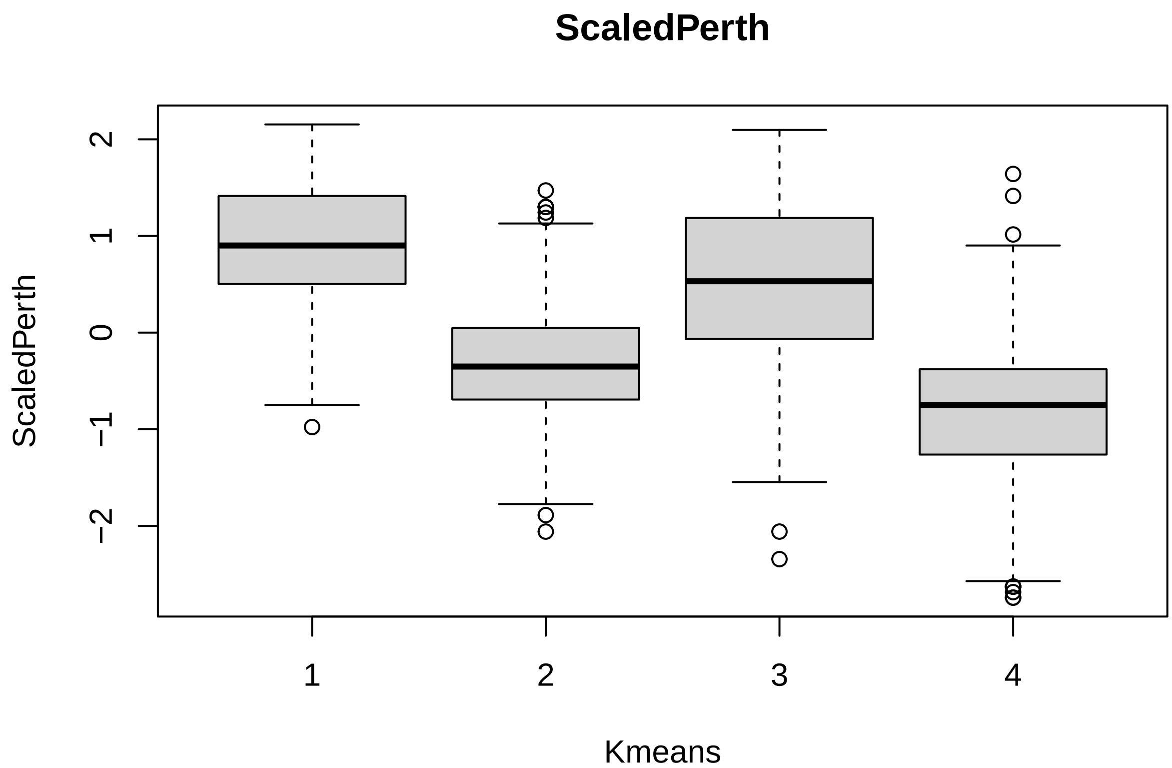

ScaledPerth
ScaledPerth
3
2
1
0
−1
1
2
3
4
Kmeans

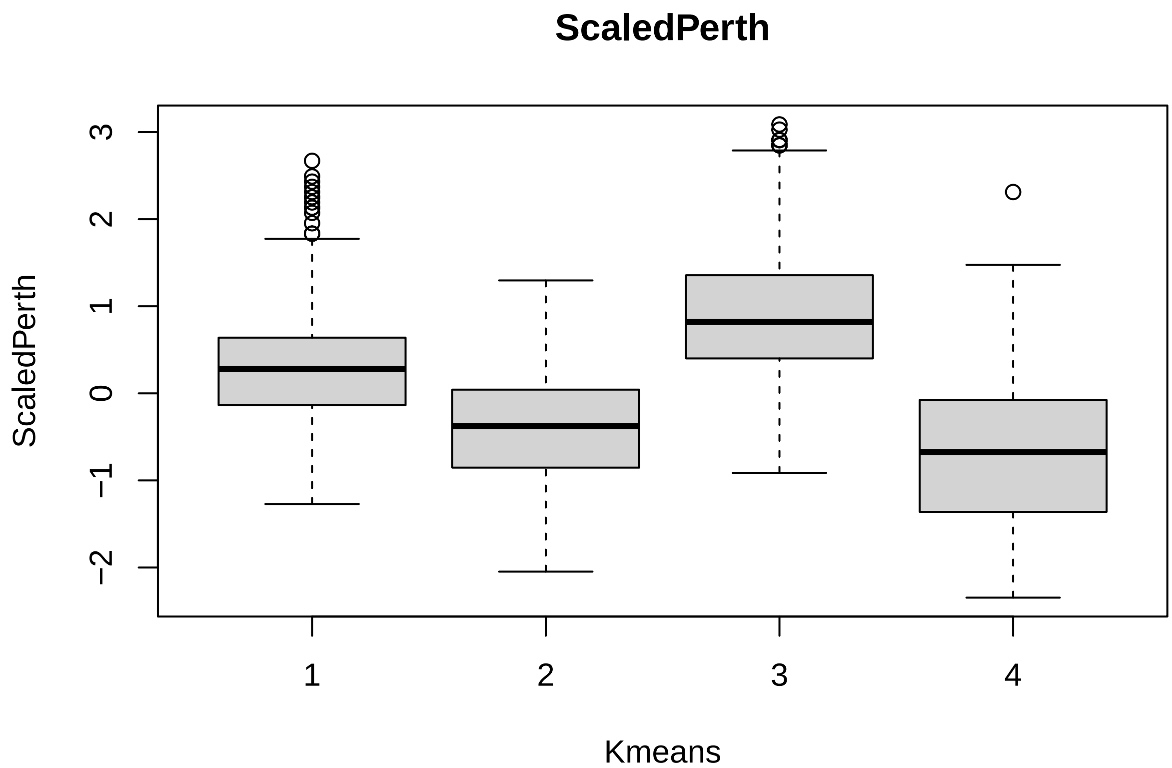

ScaledPerth
ScaledPerth
1
0
−1
−2
1
2
3
4
Kmeans

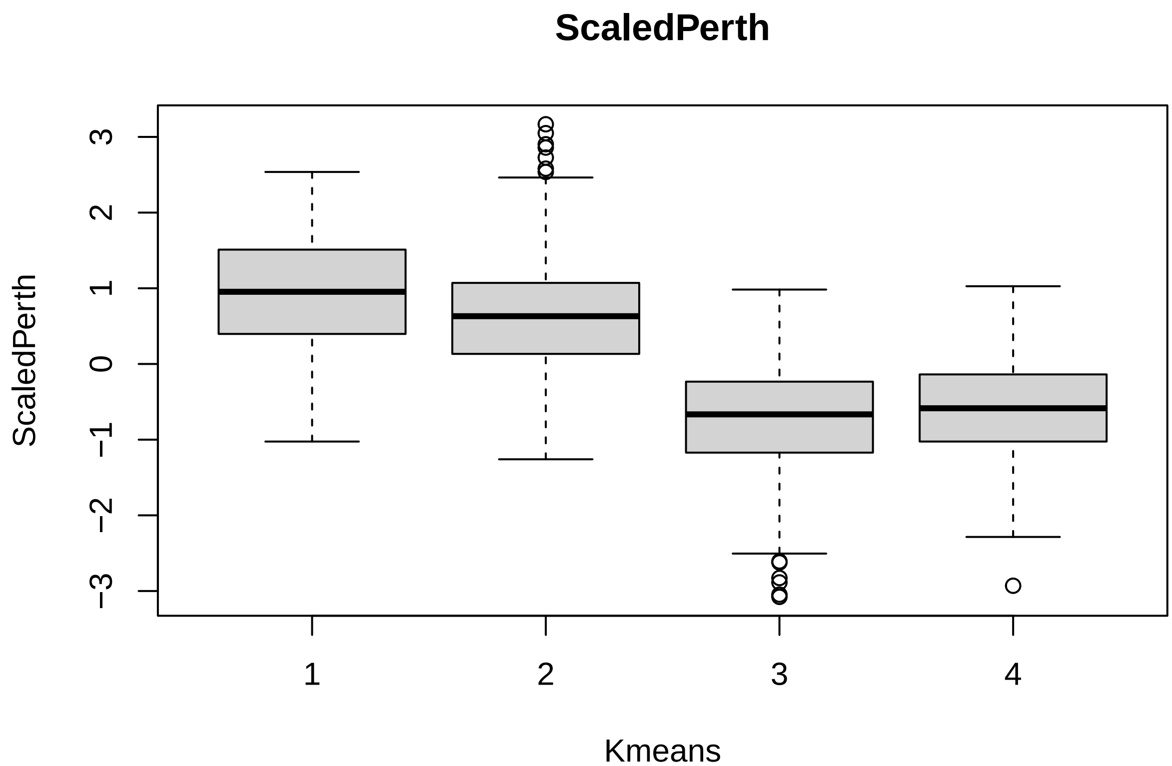

ScaledPerth
ScaledPerth
1.5
1.0
0.5
0.0
−1.0
1
2
3
4
Kmeans

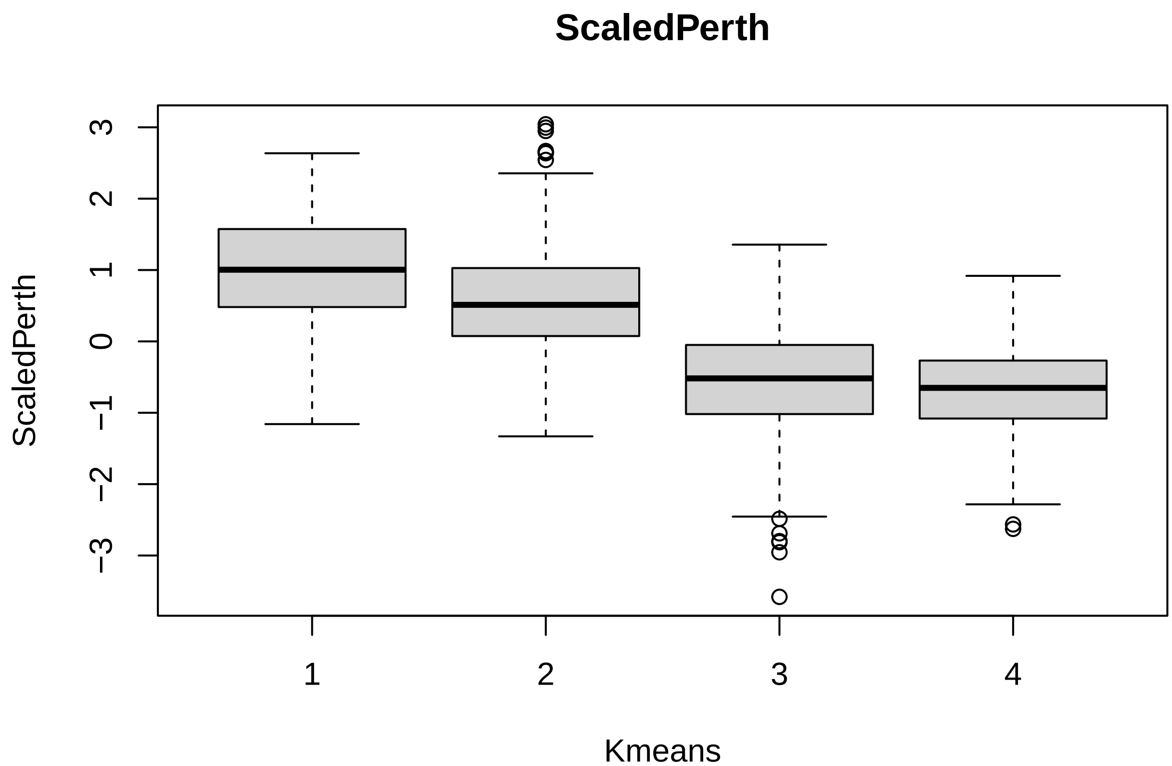

ScaledPerth
ScaledPerth
1.5
1.0
0.5
0.0
−1.0
1
2
3
4
Kmeans

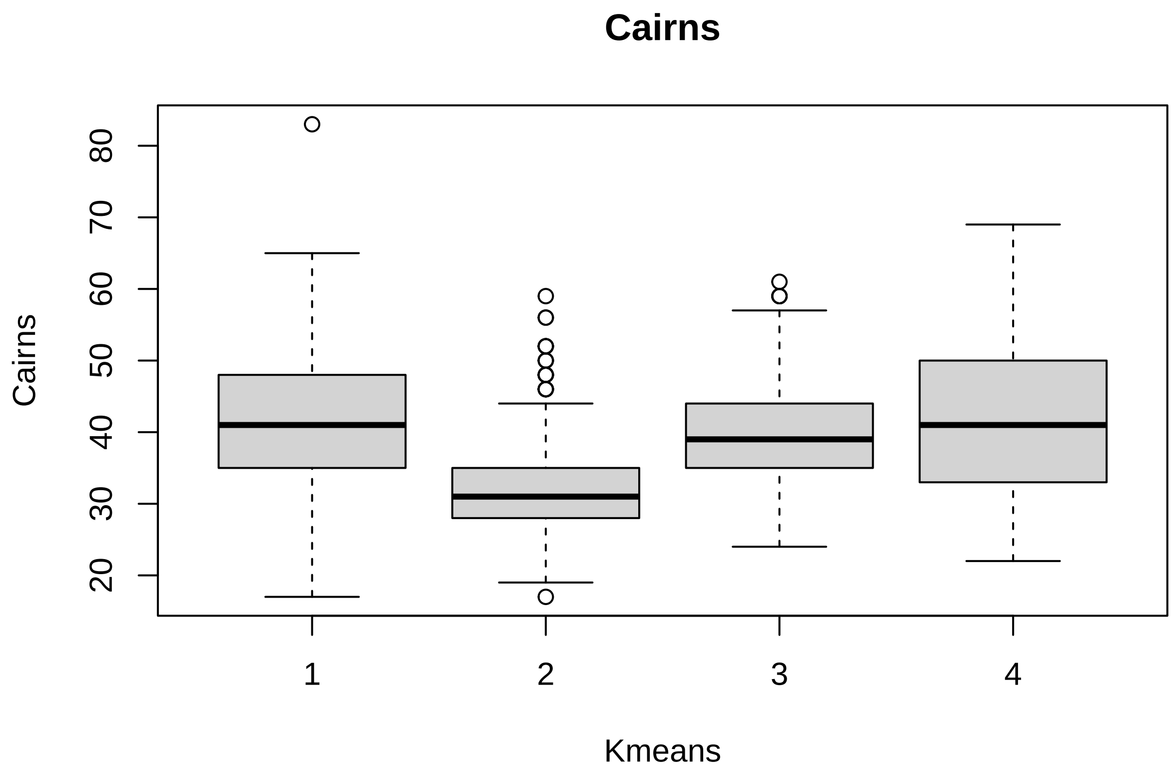

ScaledPerth
ScaledPerth
0
2
4
6
8
10
1
2
3
4
Kmeans

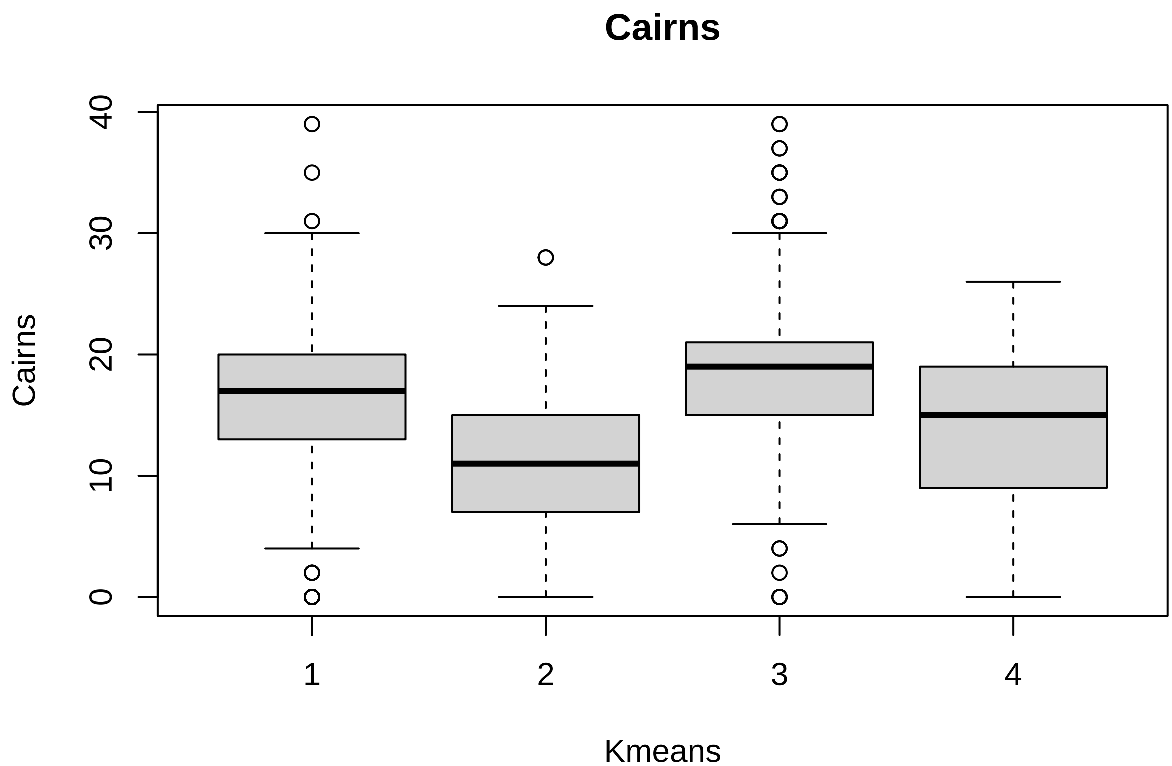

ScaledPerth
ScaledPerth
2
1
0
−1
−2
1
2
3
4
Kmeans

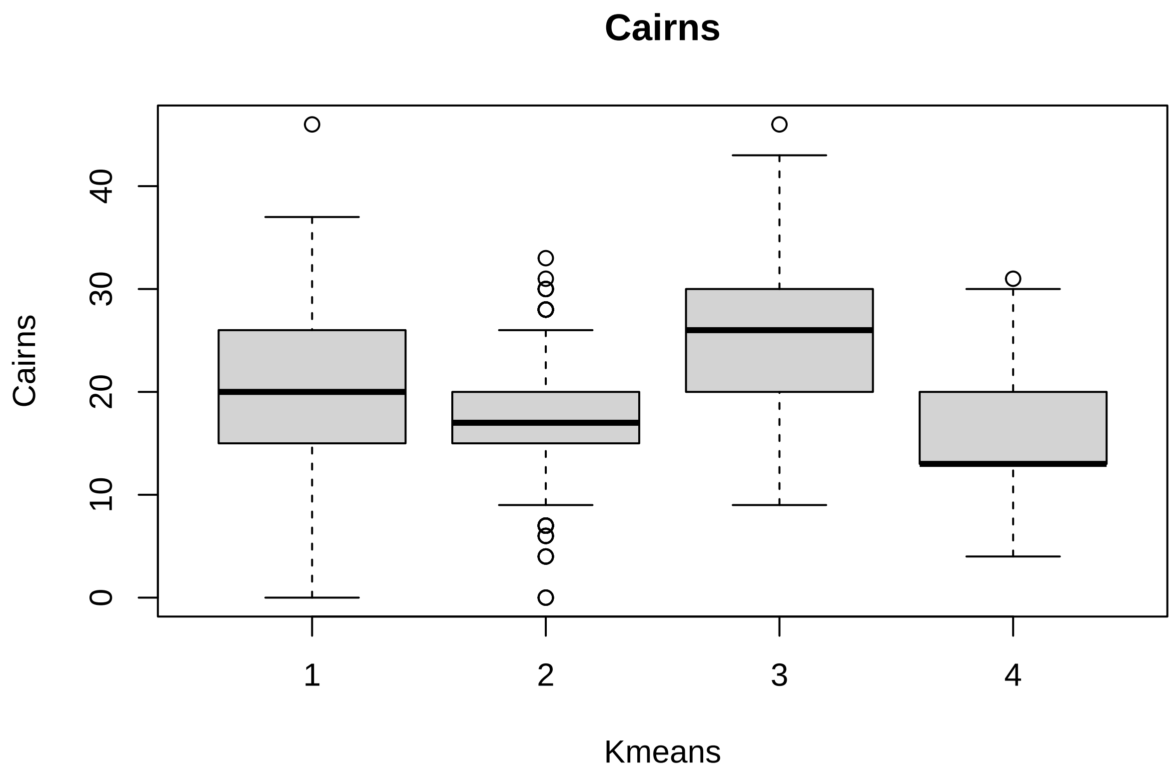

ScaledPerth
ScaledPerth
3
2
1
0
−1
−2
1
2
3
4
Kmeans

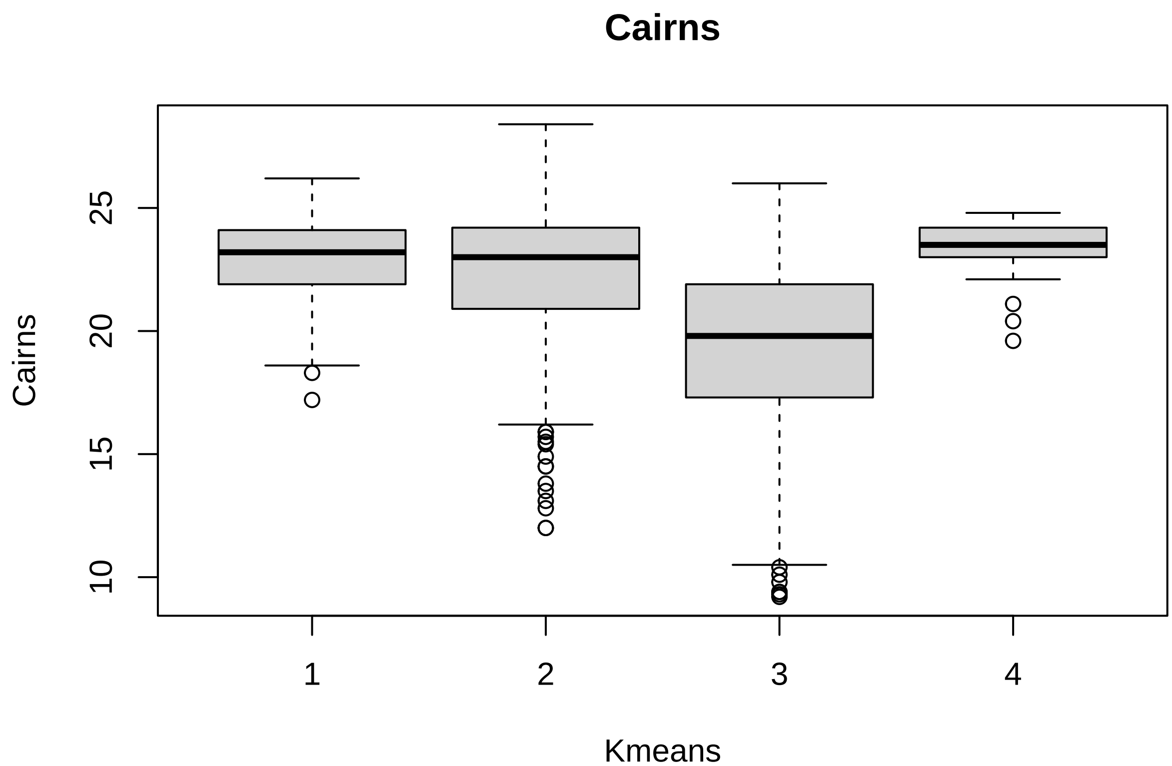

ScaledPerth
ScaledPerth
3
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

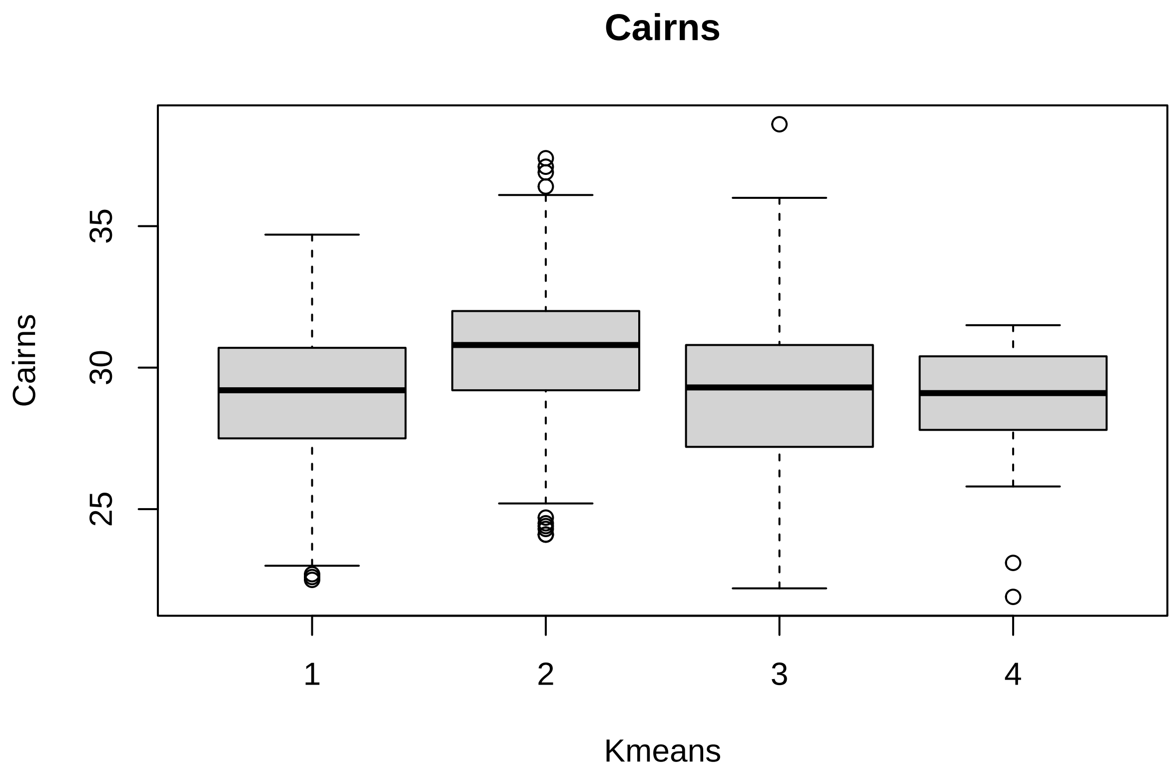

ScaledPerth
ScaledPerth
3
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

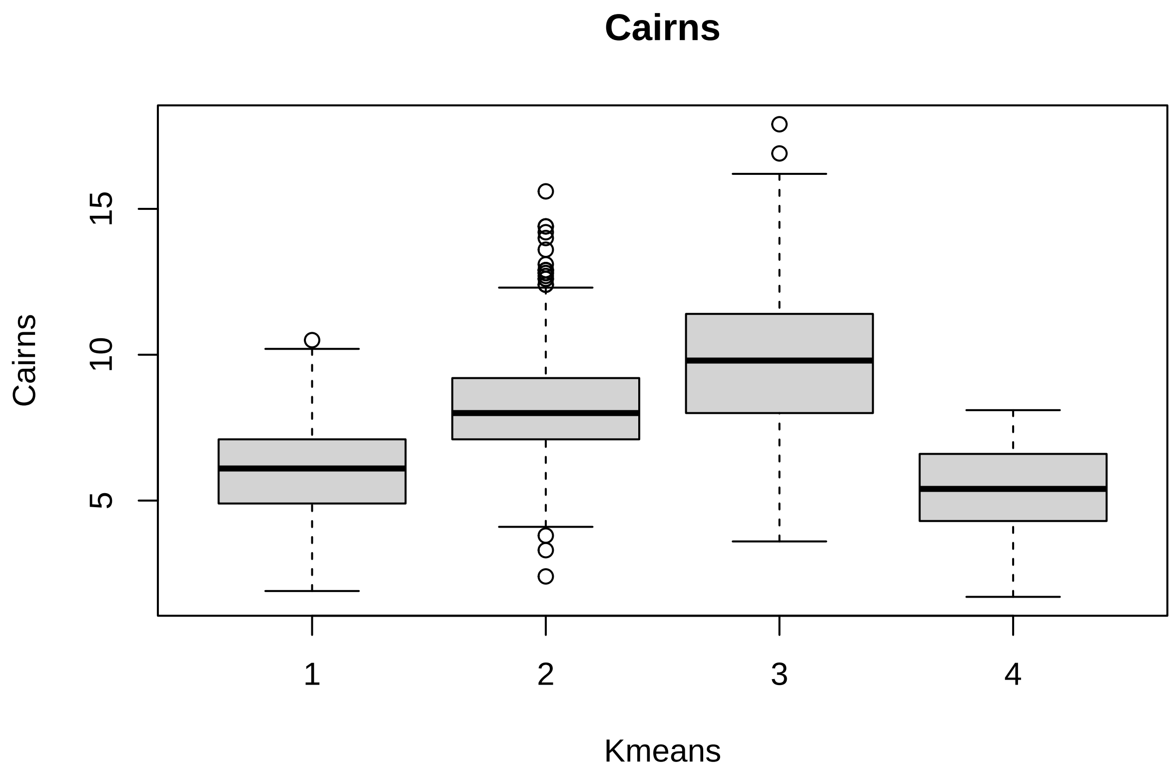

Cairns
Cairns
20
30
40
50
60
70
80
1
2
3
4
Kmeans

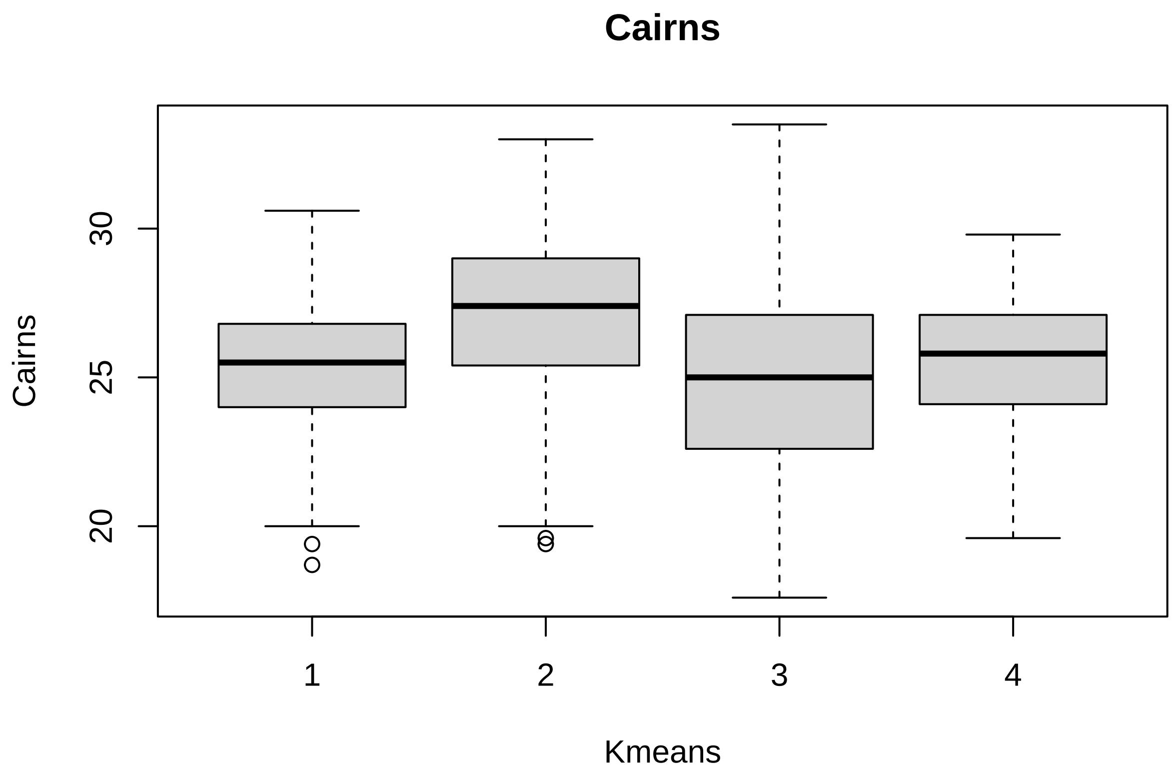

Cairns
Cairns
0
10
20
30
40
1
2
3
4
Kmeans

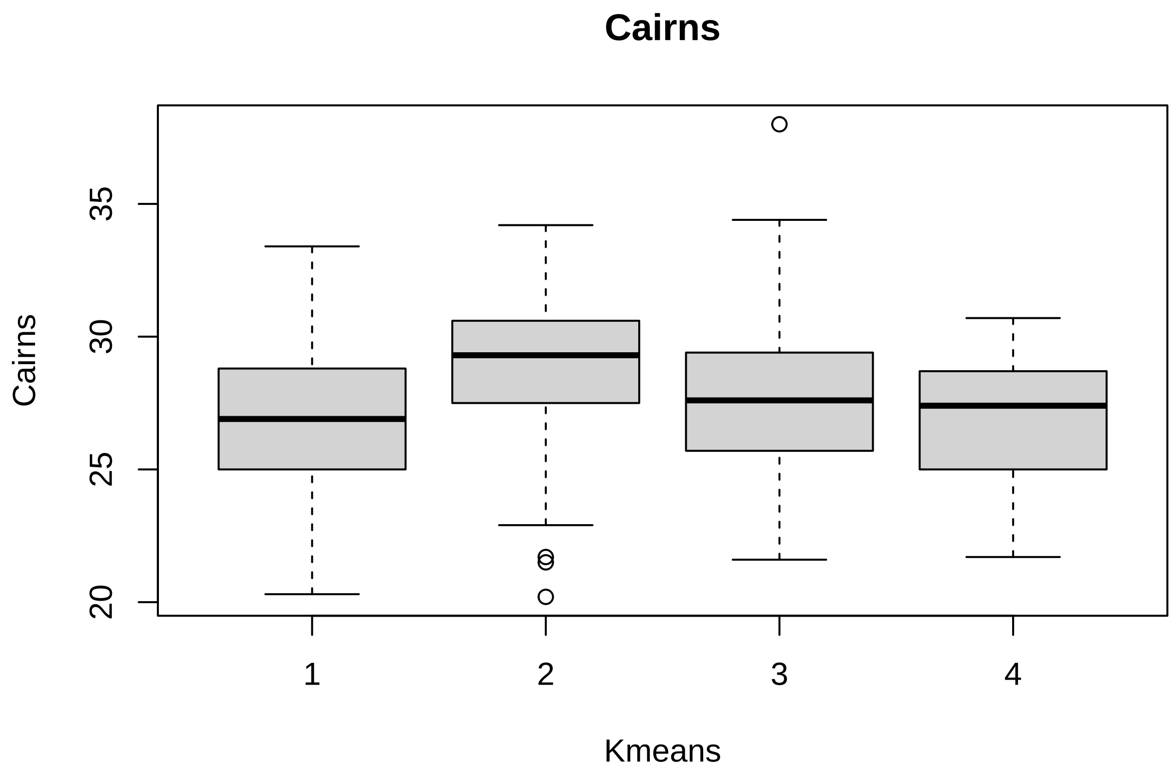

Cairns
Cairns
0
10
20
30
40
1
2
3
4
Kmeans

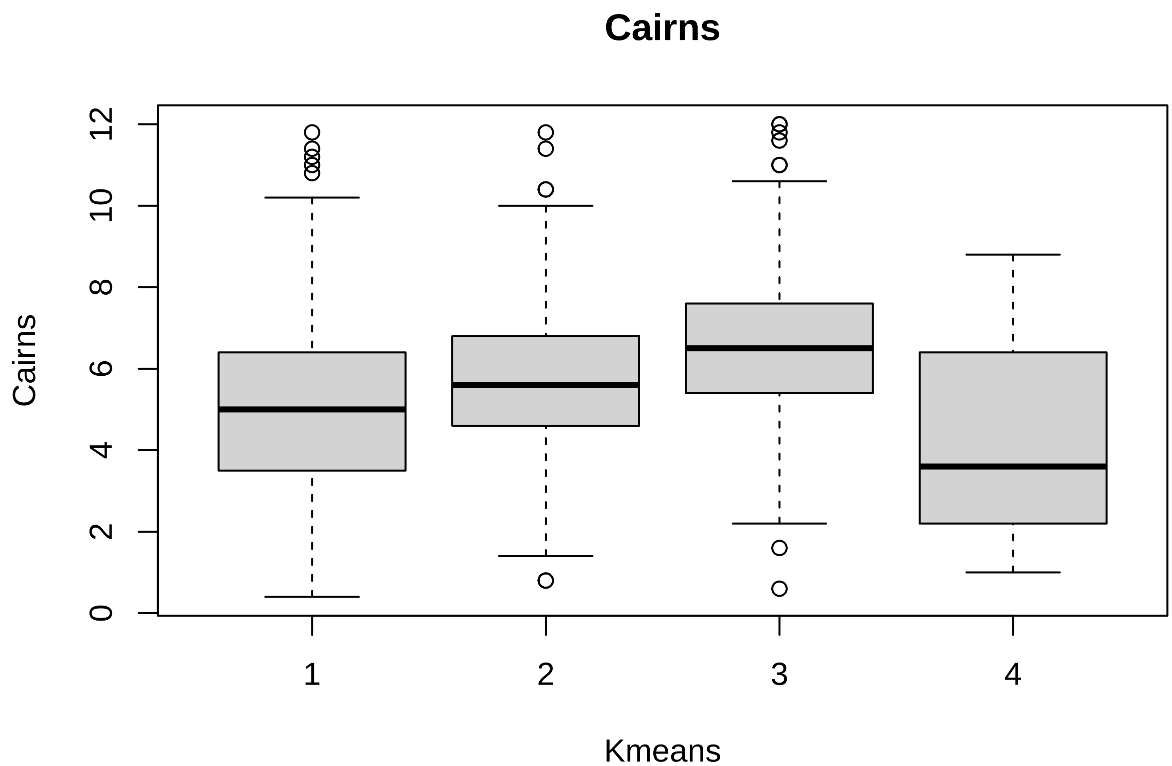

Cairns
Cairns
25
20
15
10
1
2
3
4
Kmeans

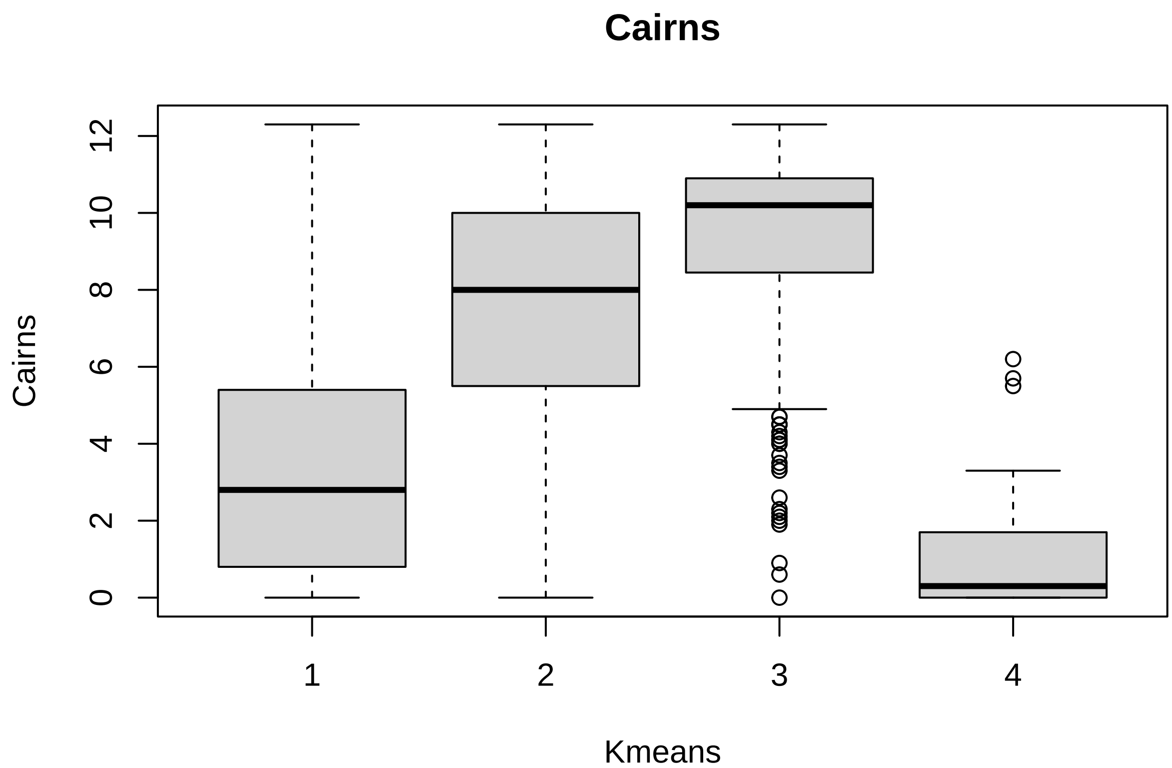

Cairns
Cairns
35
30
25
1
2
3
4
Kmeans

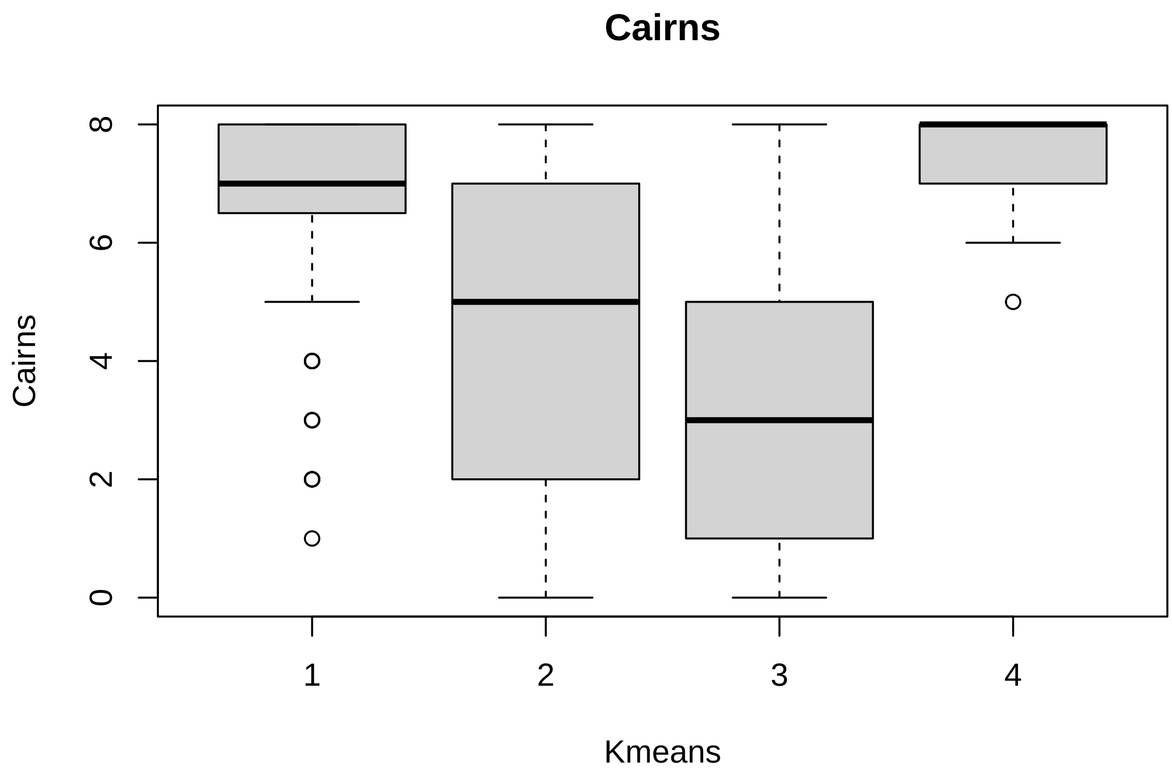

Cairns
Cairns
15
10
5
1
2
3
4
Kmeans

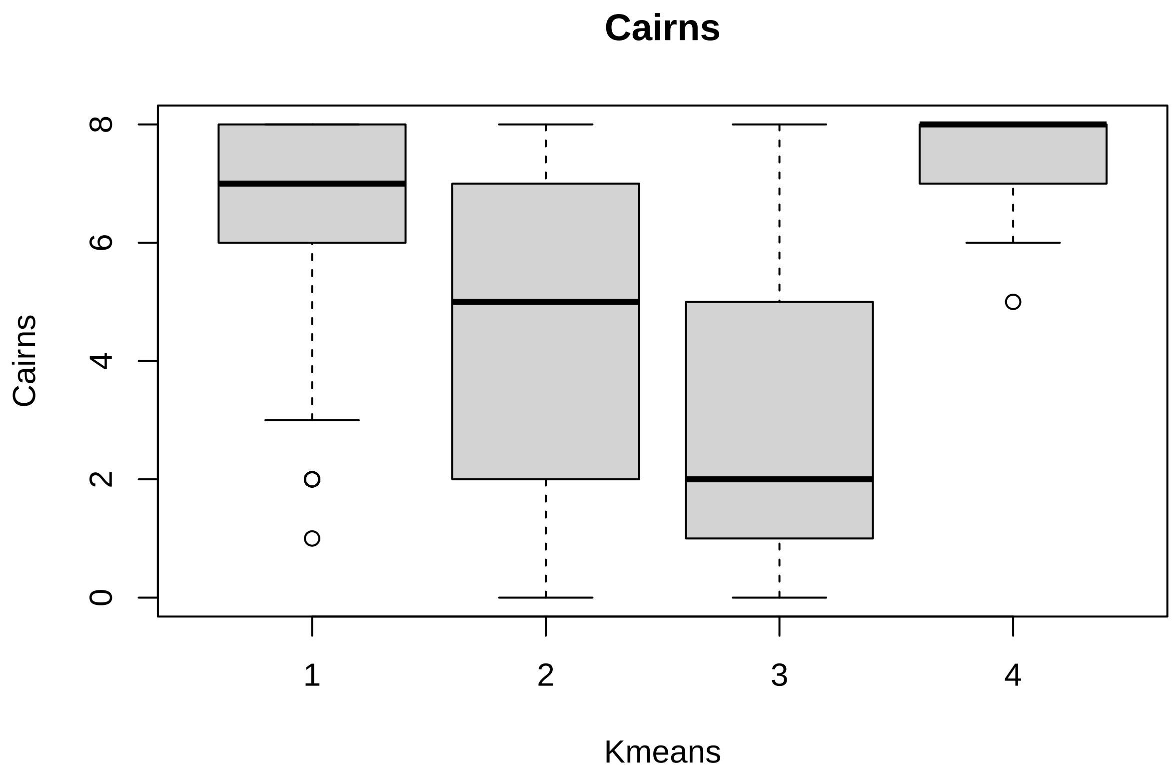

Cairns
Cairns
30
25
20
1
2
3
4
Kmeans

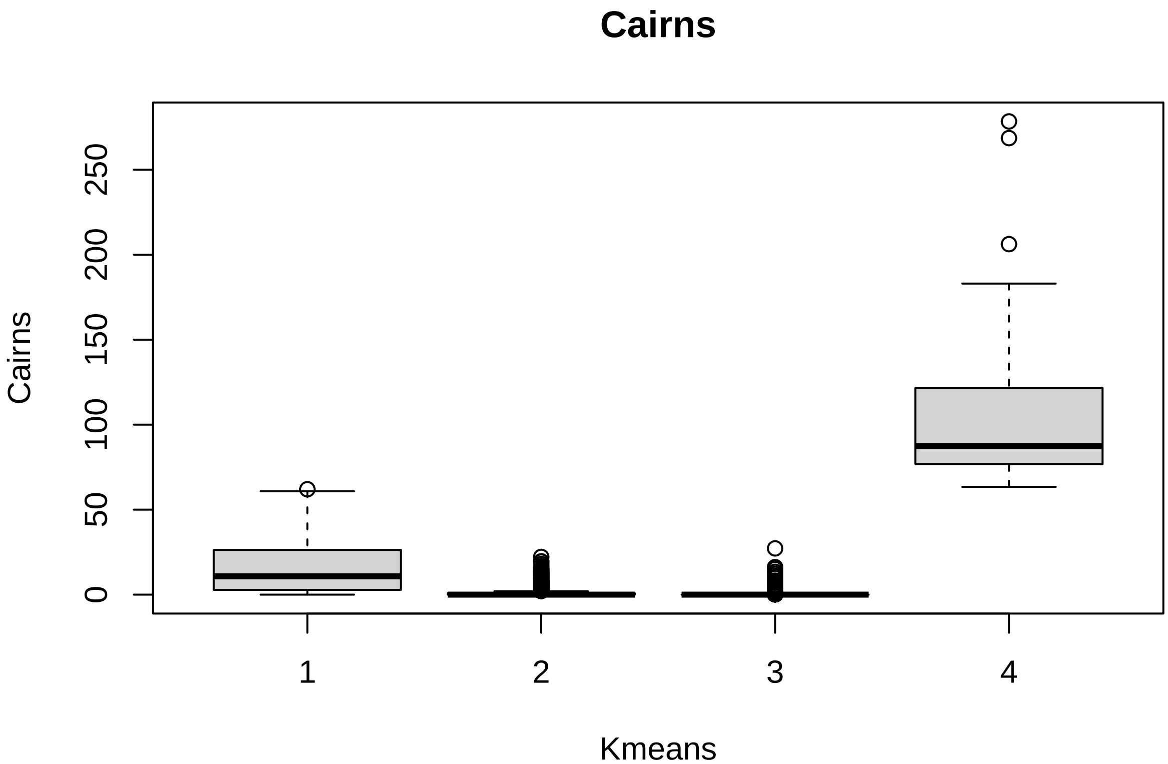

Cairns
Cairns
35
30
25
20
1
2
3
4
Kmeans


Cairns
Cairns
0
2
4
6
8
10
12
1
2
3
4
Kmeans

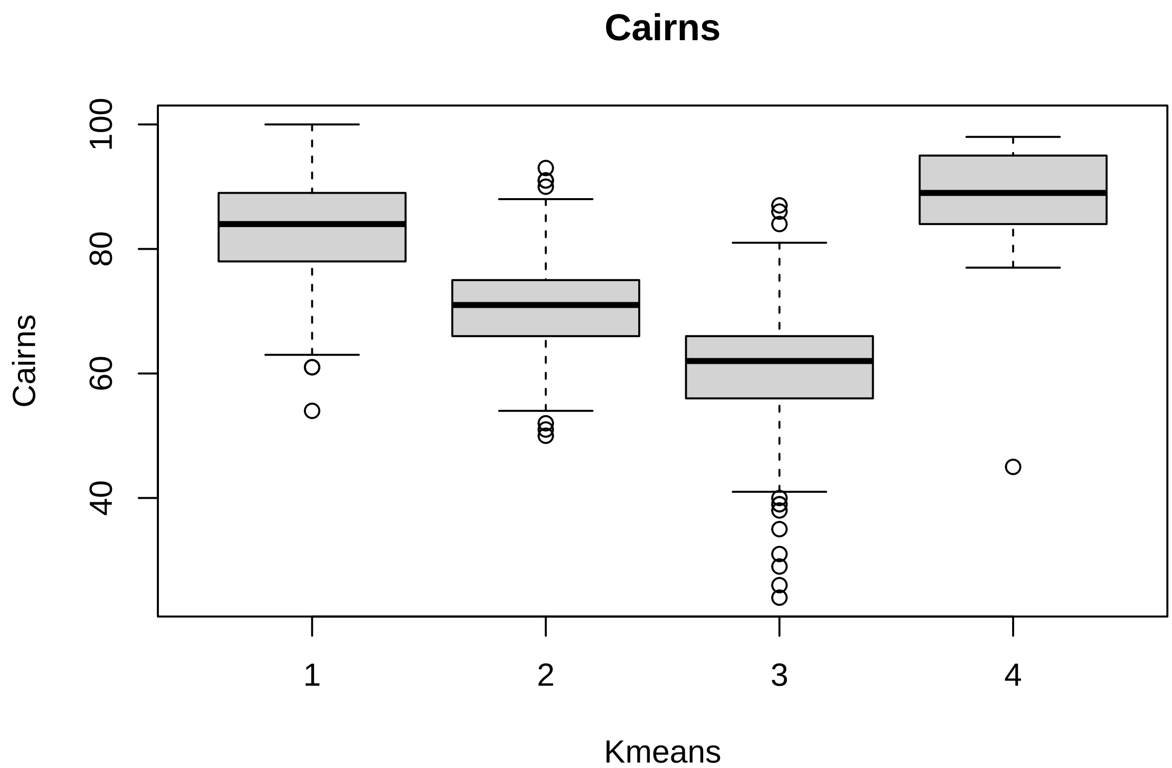

Cairns
Cairns
0
2
4
6
8
10
12
1
2
3
4
Kmeans

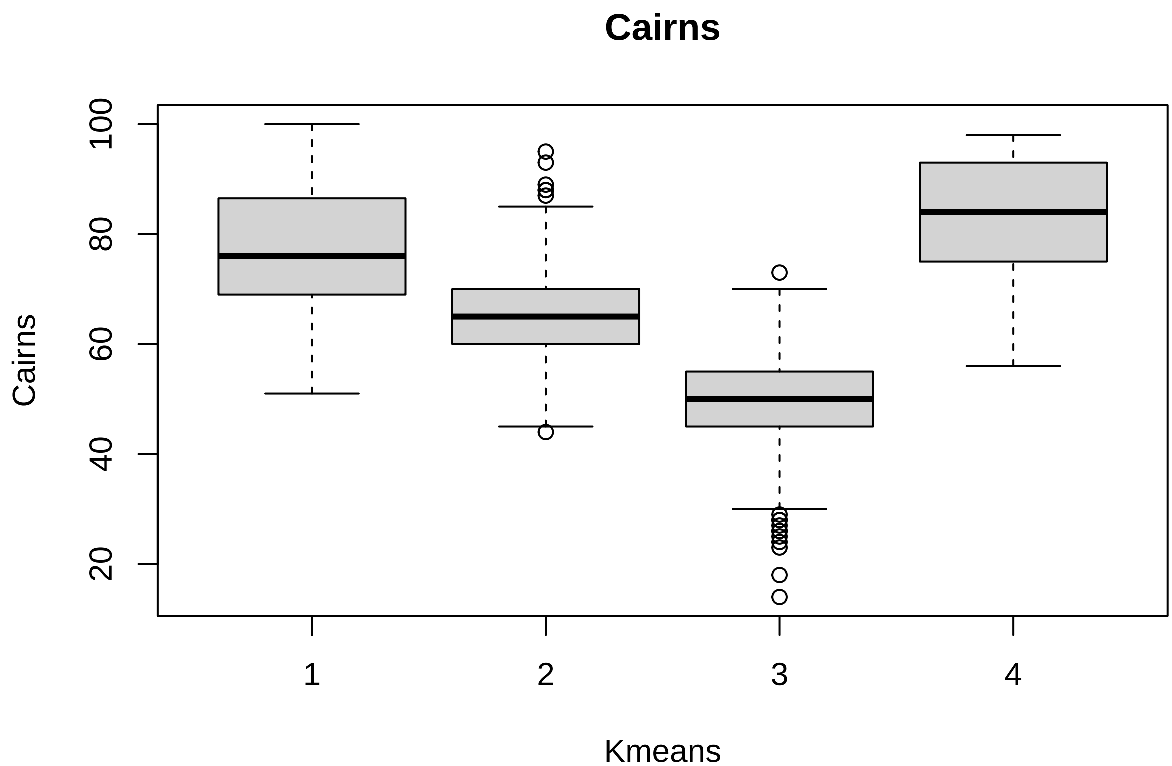

Cairns
Cairns
0
2
4
6
8
1
2
3
4
Kmeans

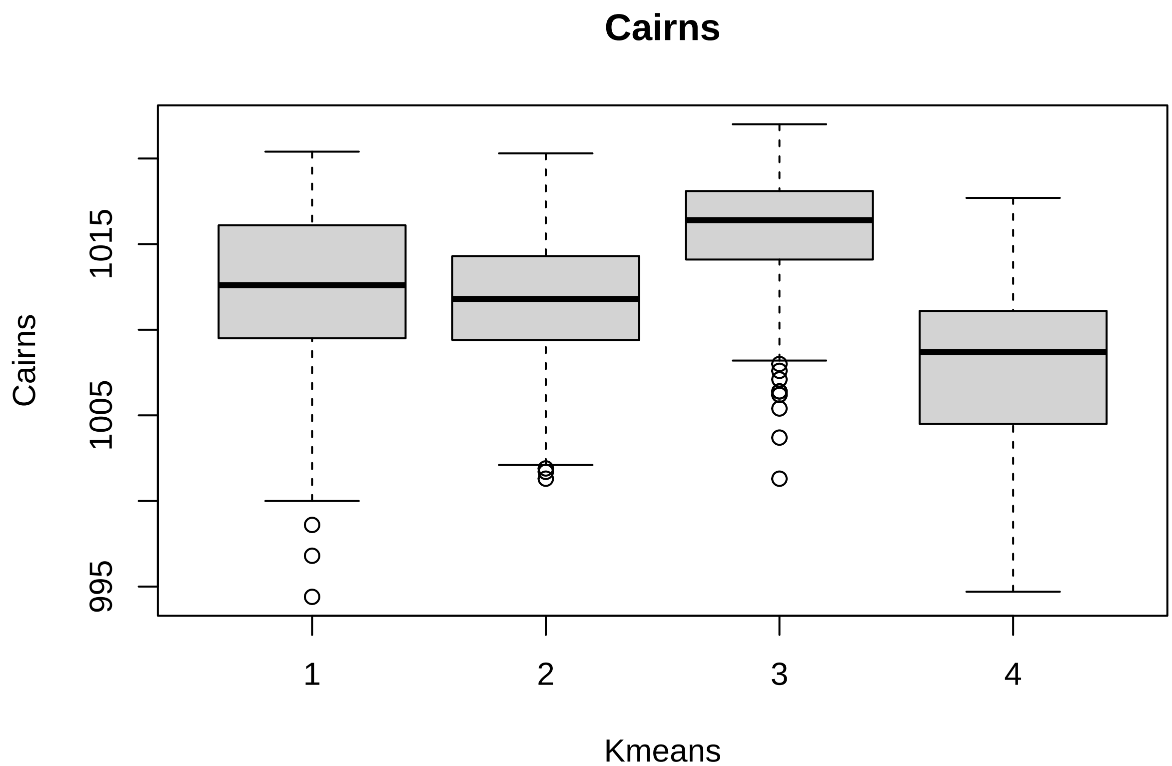

Cairns
Cairns
0
2
4
6
8
1
2
3
4
Kmeans

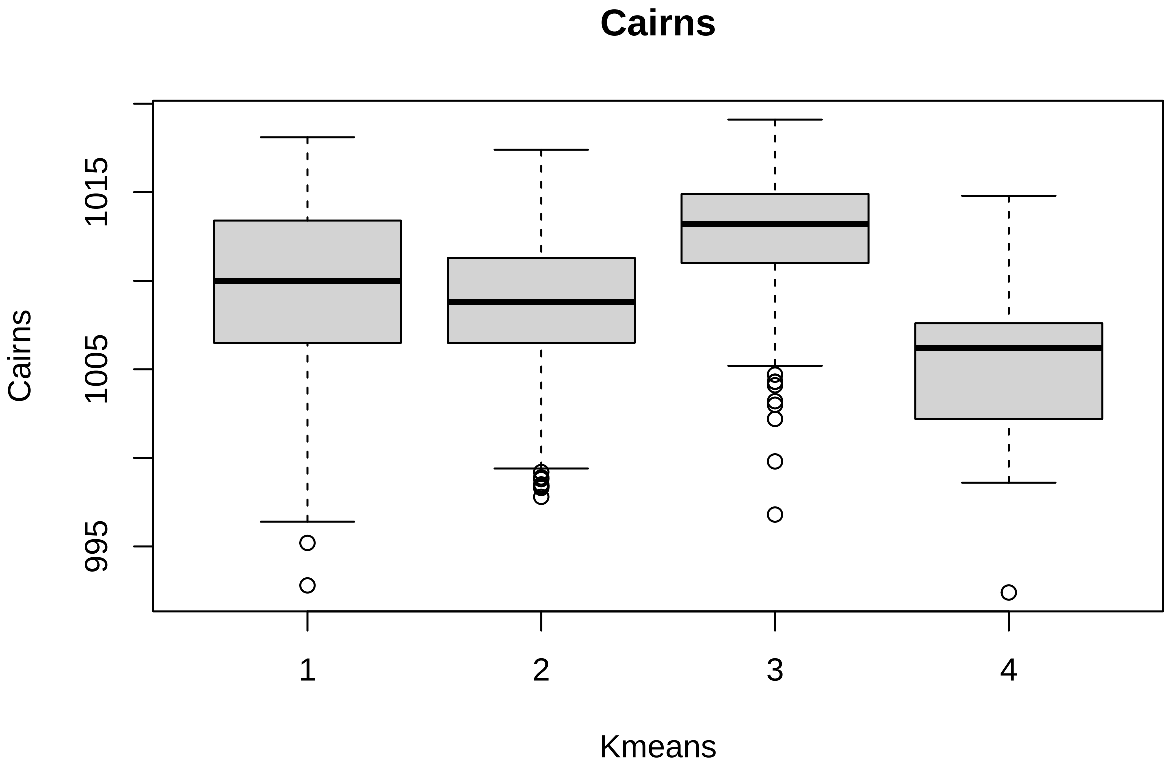


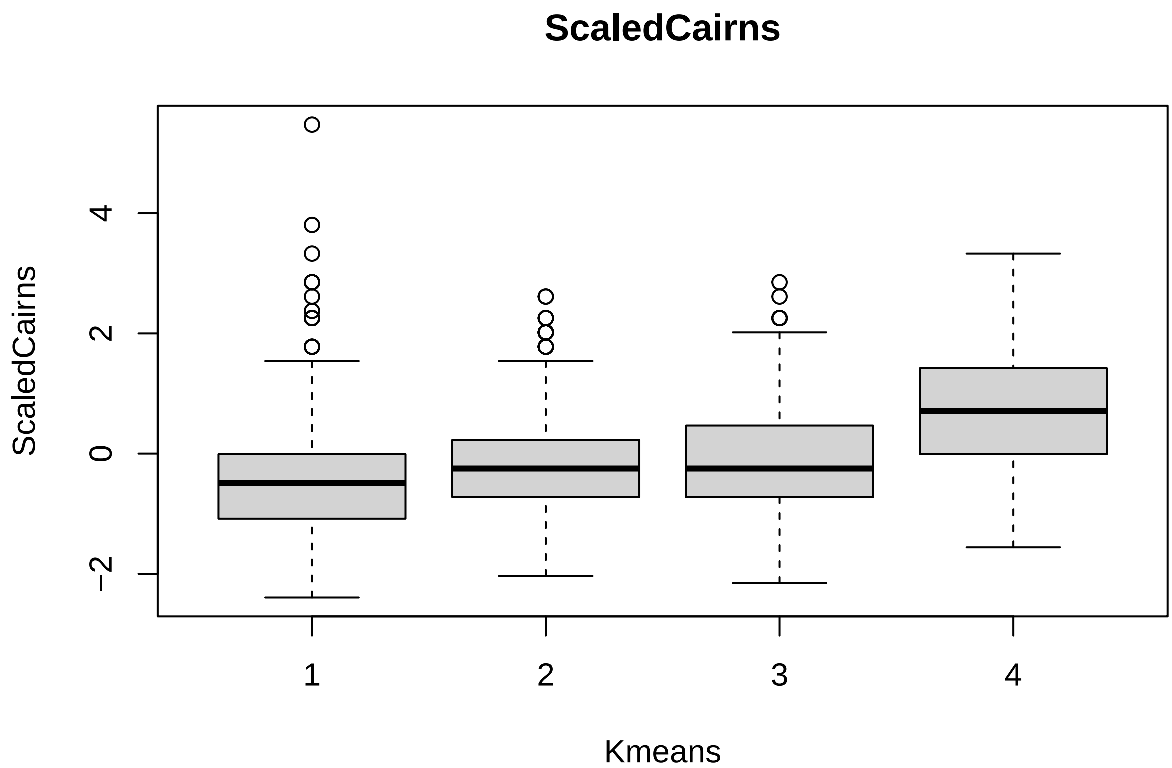
Cairns
0
50
100
150
200
250
Cairns
1
2
3
4
Kmeans

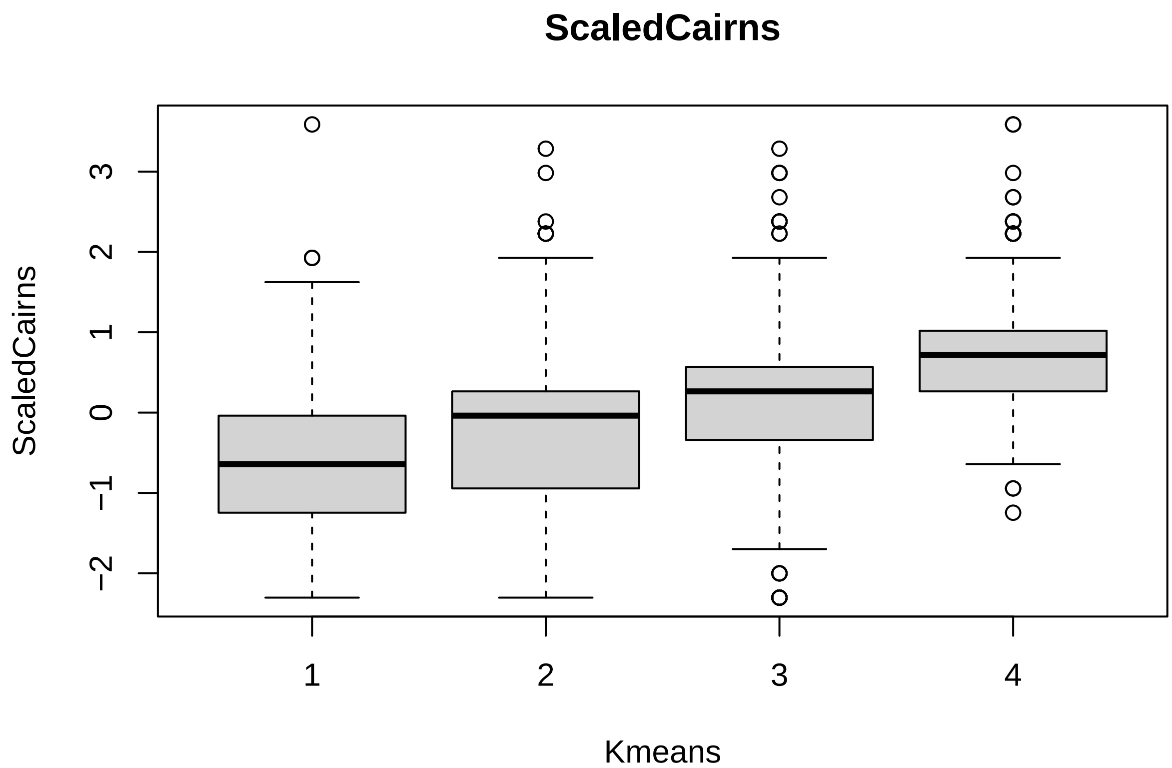

Cairns
Cairns
100
80
60
40
1
2
3
4
Kmeans

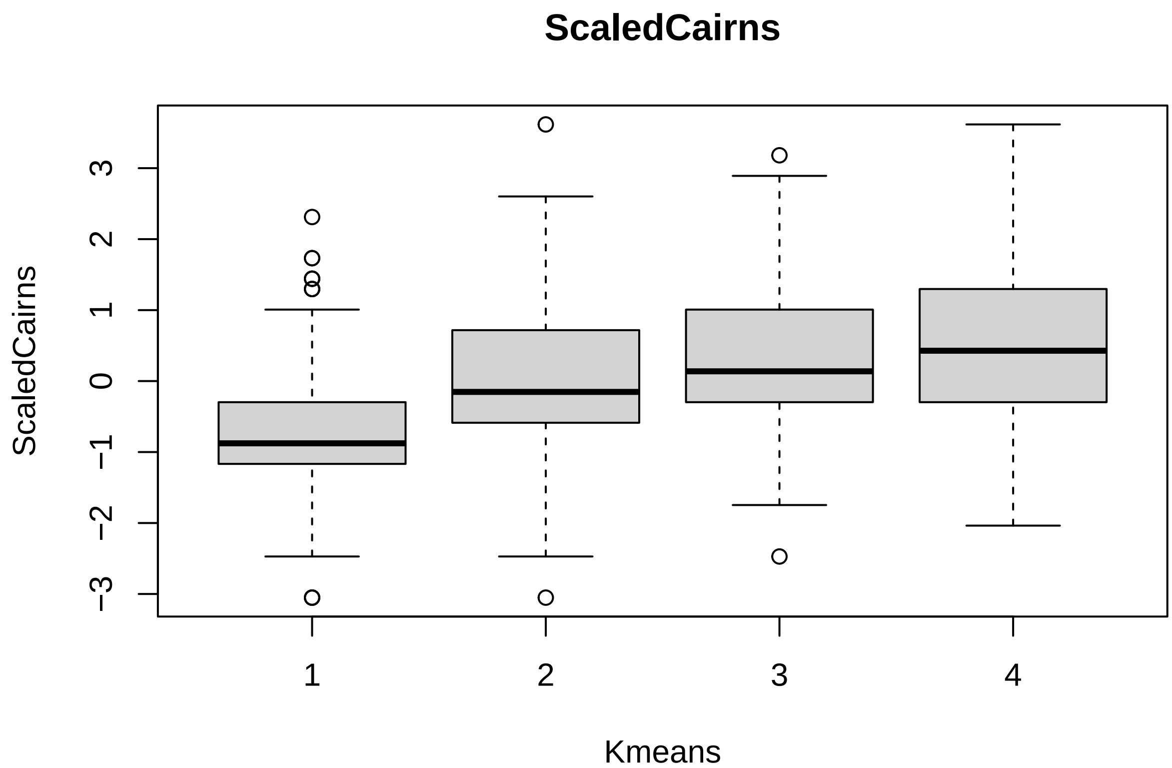

Cairns
100
80
60
40
20
Cairns
1
2
3
4
Kmeans

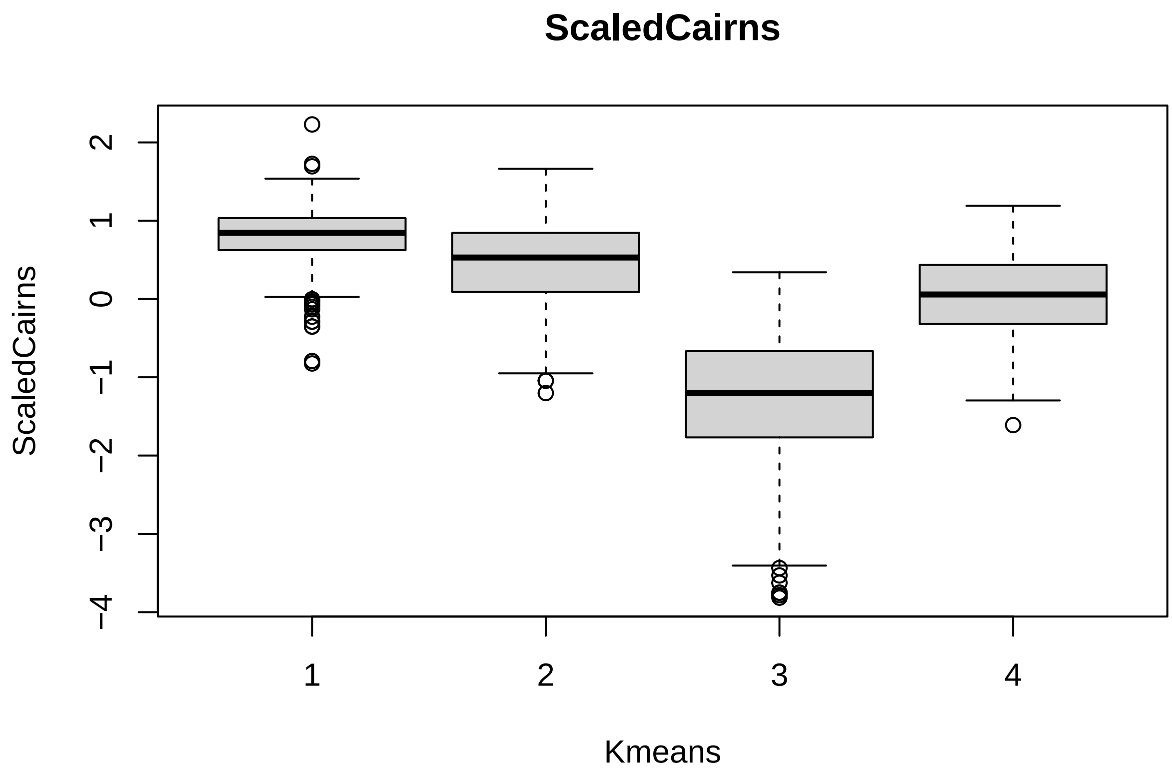

Cairns
Cairns
1015
1005
995
1
2
3
4
Kmeans

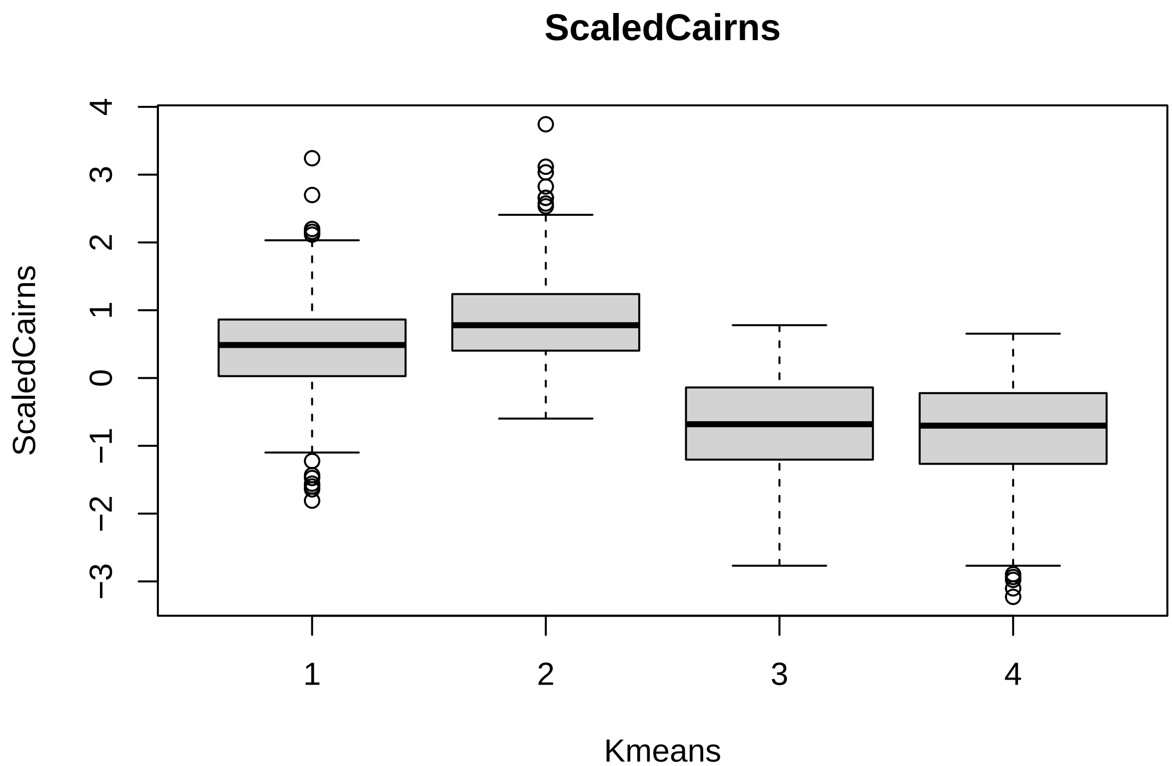

Cairns
Cairns
1015
1005
995
1
2
3
4
Kmeans

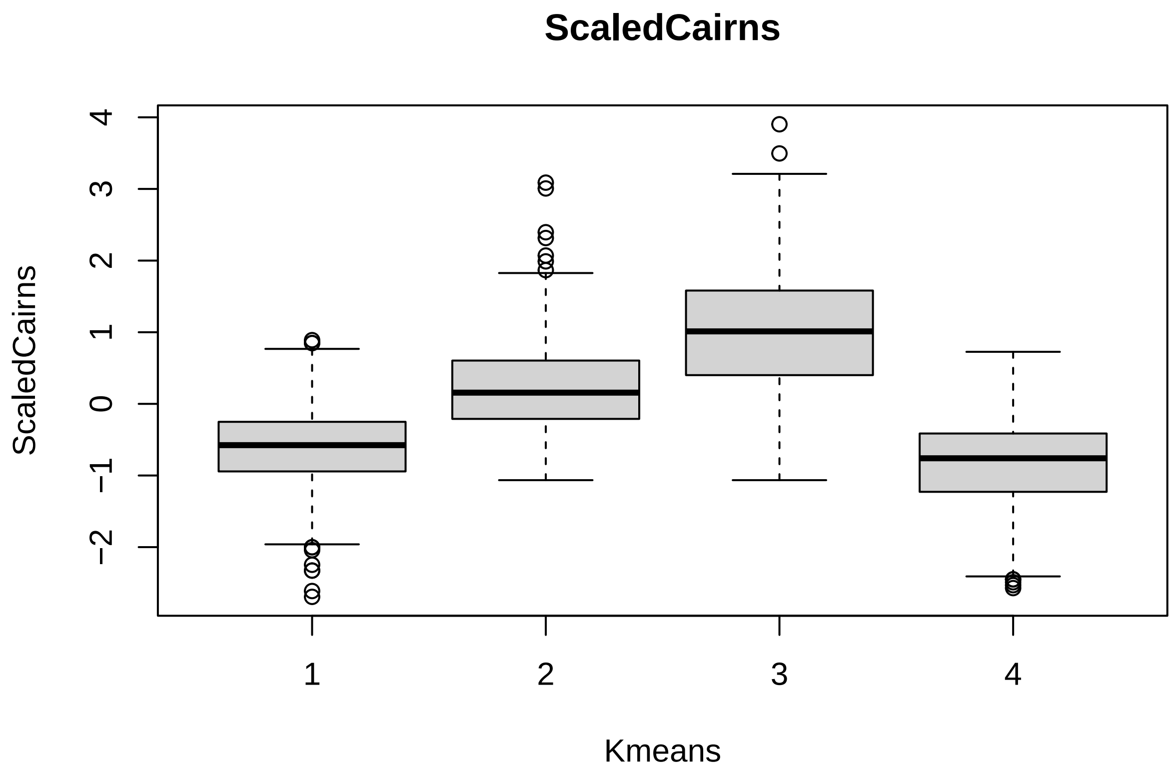

ScaledCairns
ScaledCairns
−2
0
2
4
1
2
3
4
Kmeans


ScaledCairns
ScaledCairns
3
2
1
0
−1
−2
1
2
3
4
Kmeans

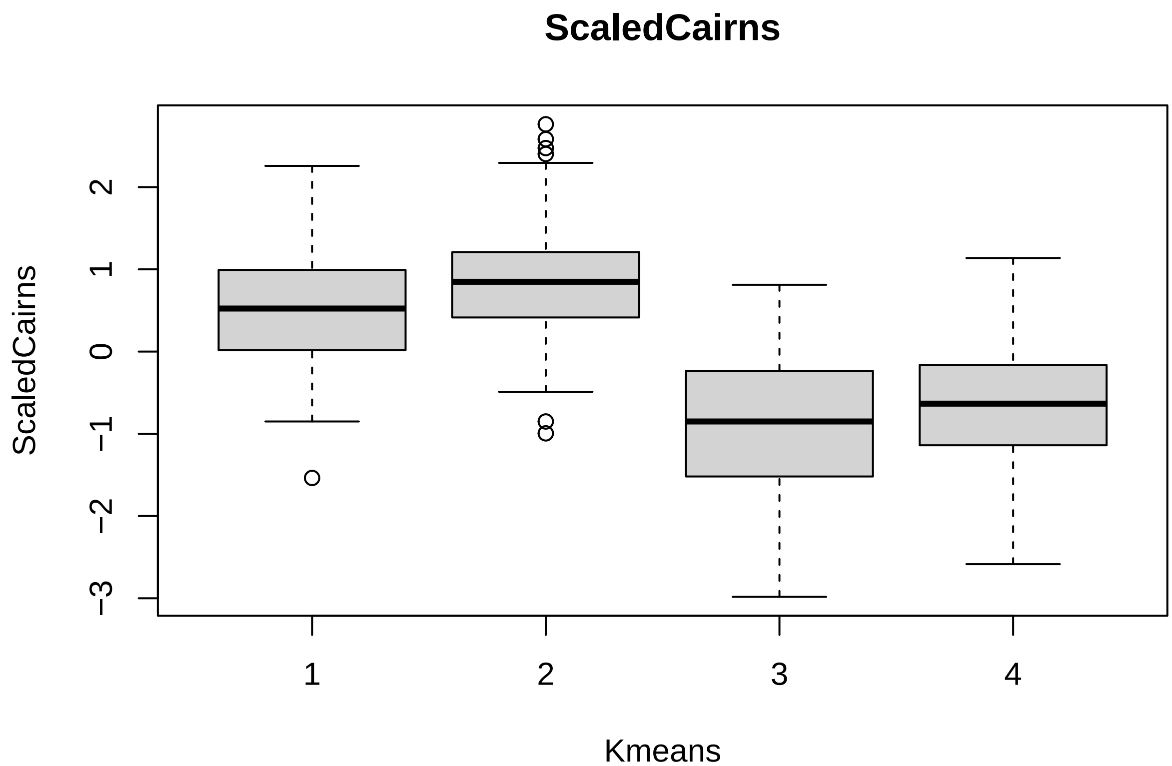

ScaledCairns
ScaledCairns
3
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

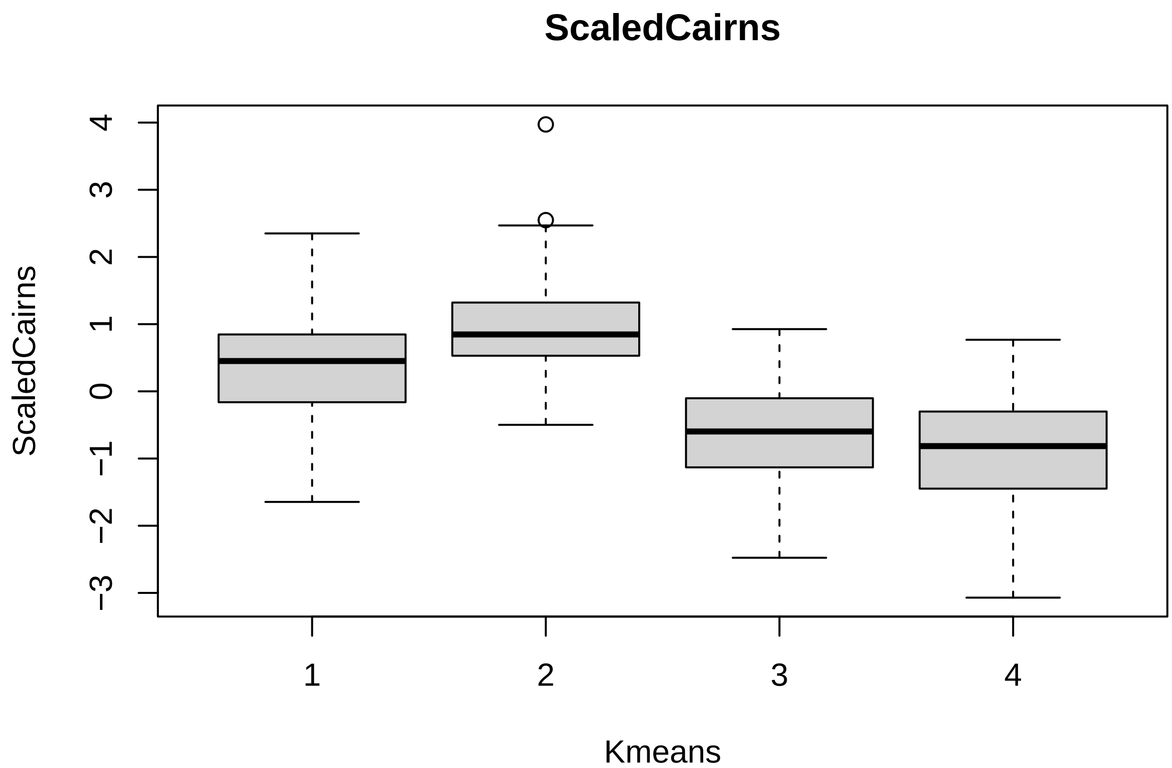

ScaledCairns
2
1
0
−1
−2
−3
−4
ScaledCairns
1
2
3
4
Kmeans

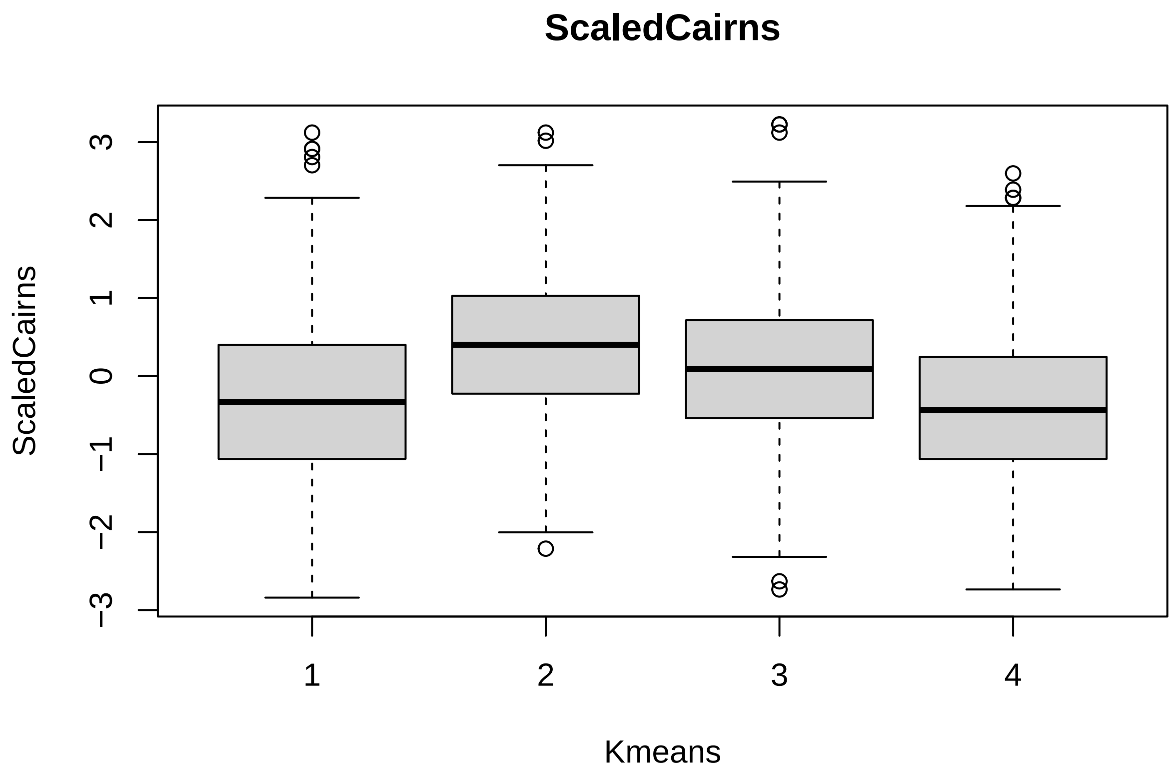

ScaledCairns
ScaledCairns
4
3
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

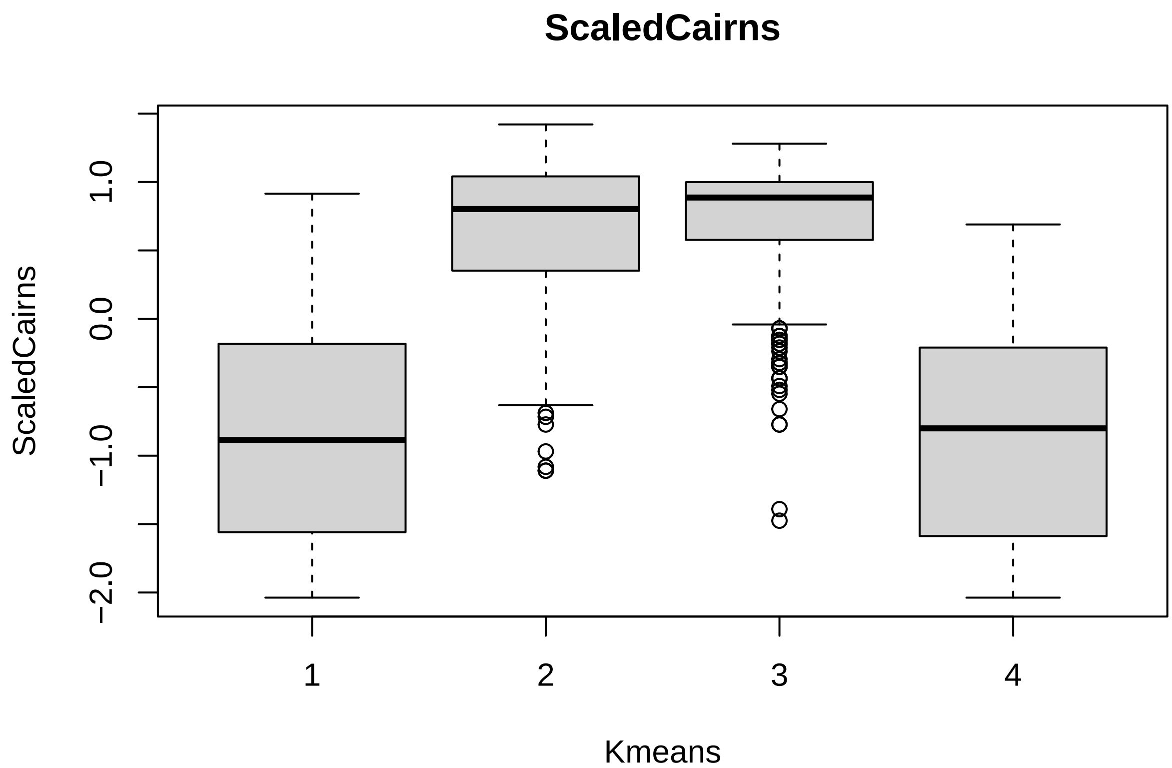


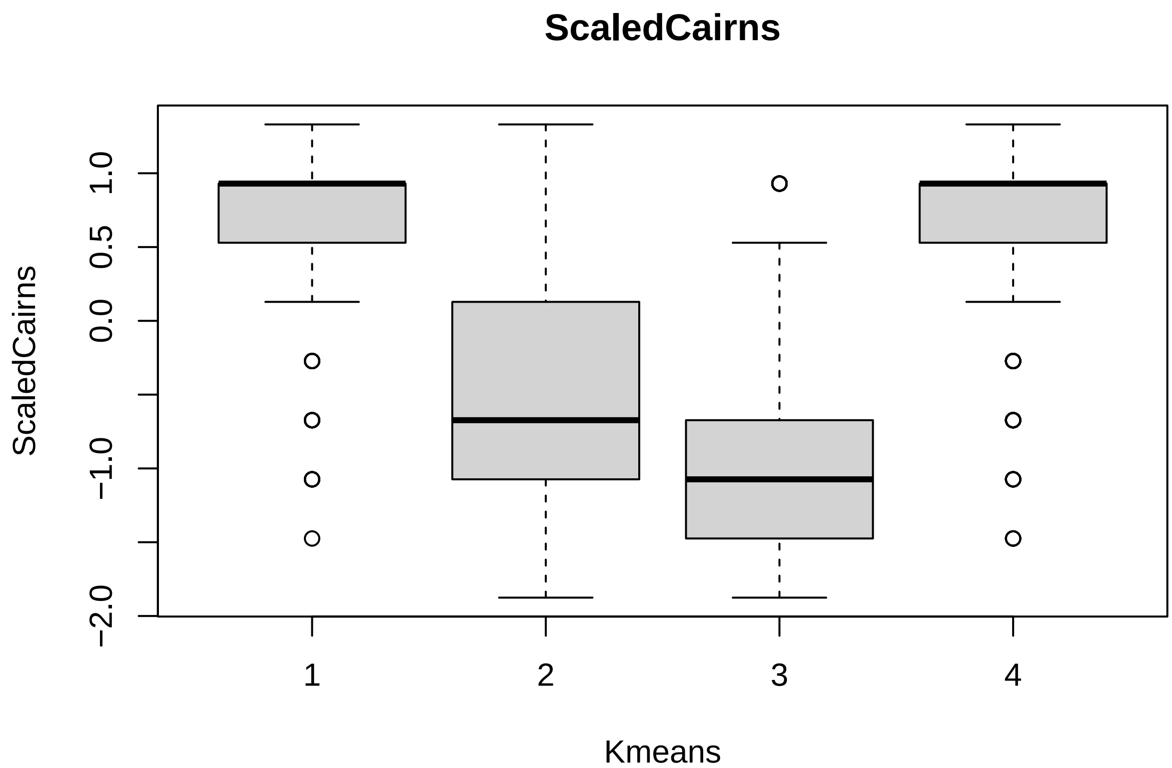
ScaledCairns
ScaledCairns
Kmeans
1
2
3
4
−2
−1
0
1
2
3
4

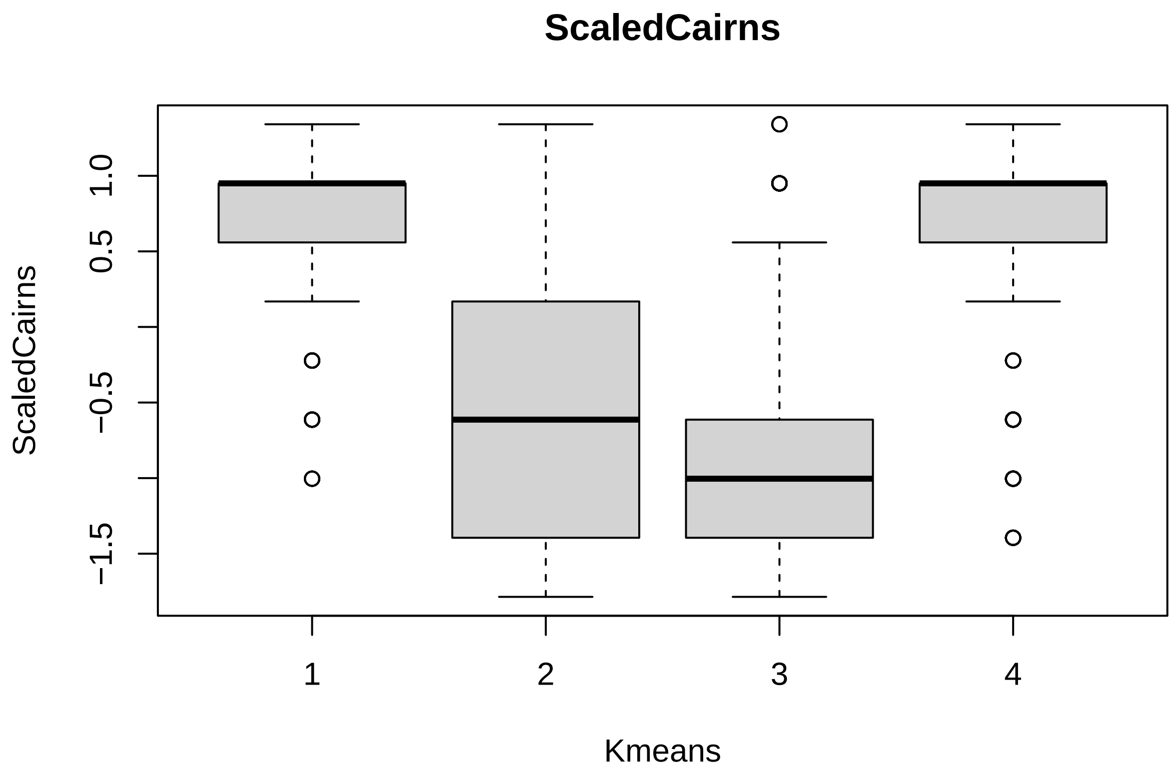

ScaledCairns
ScaledCairns
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

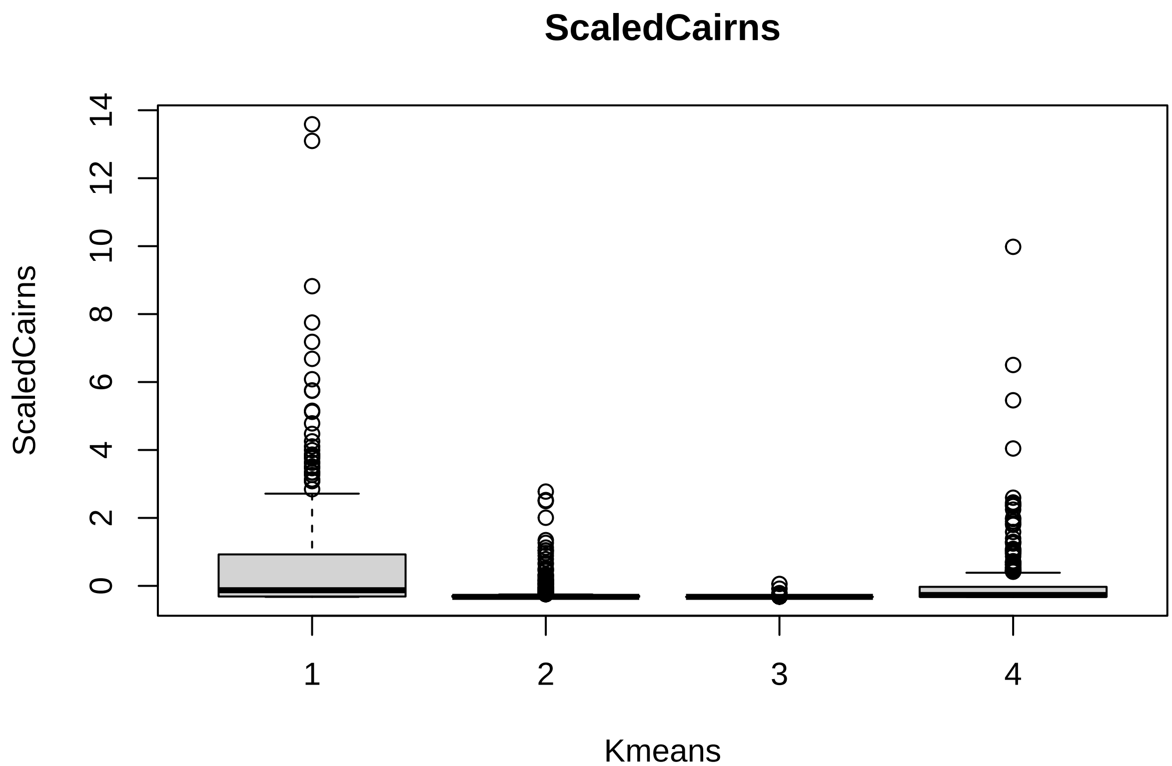

ScaledCairns
ScaledCairns
4
3
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

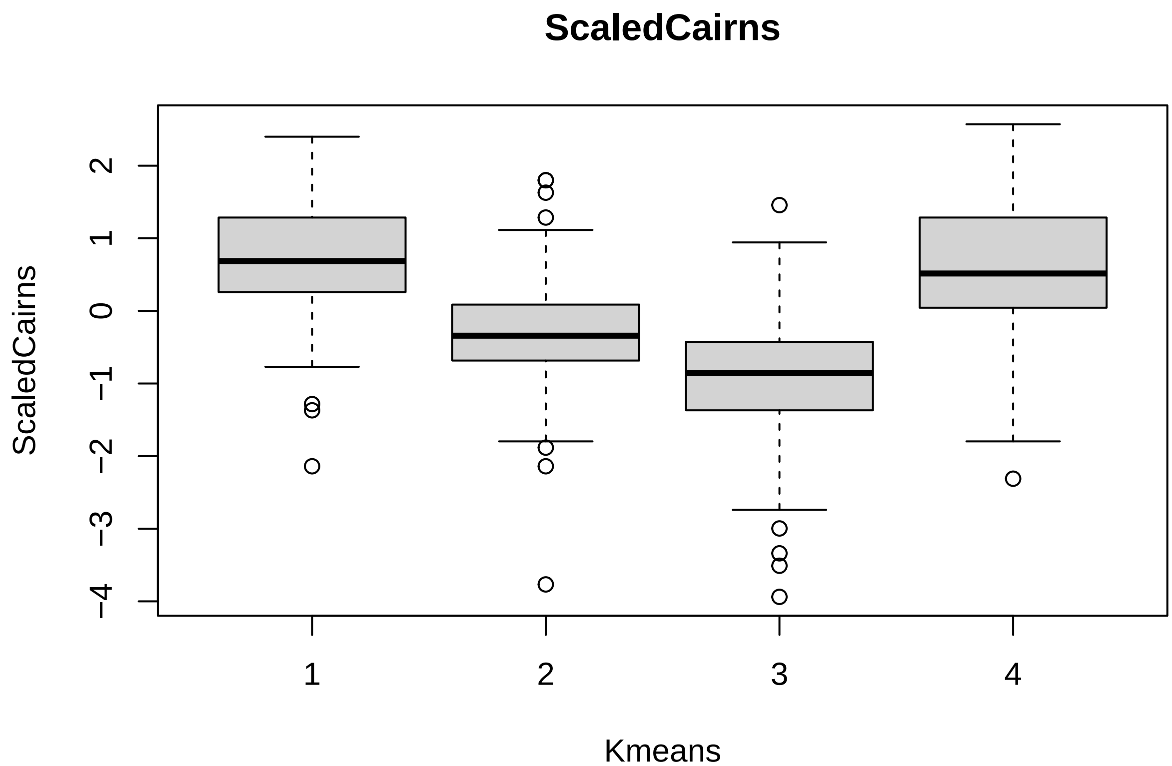

ScaledCairns
ScaledCairns
3
2
1
0
−1
−2
−3
1
2
3
4
Kmeans

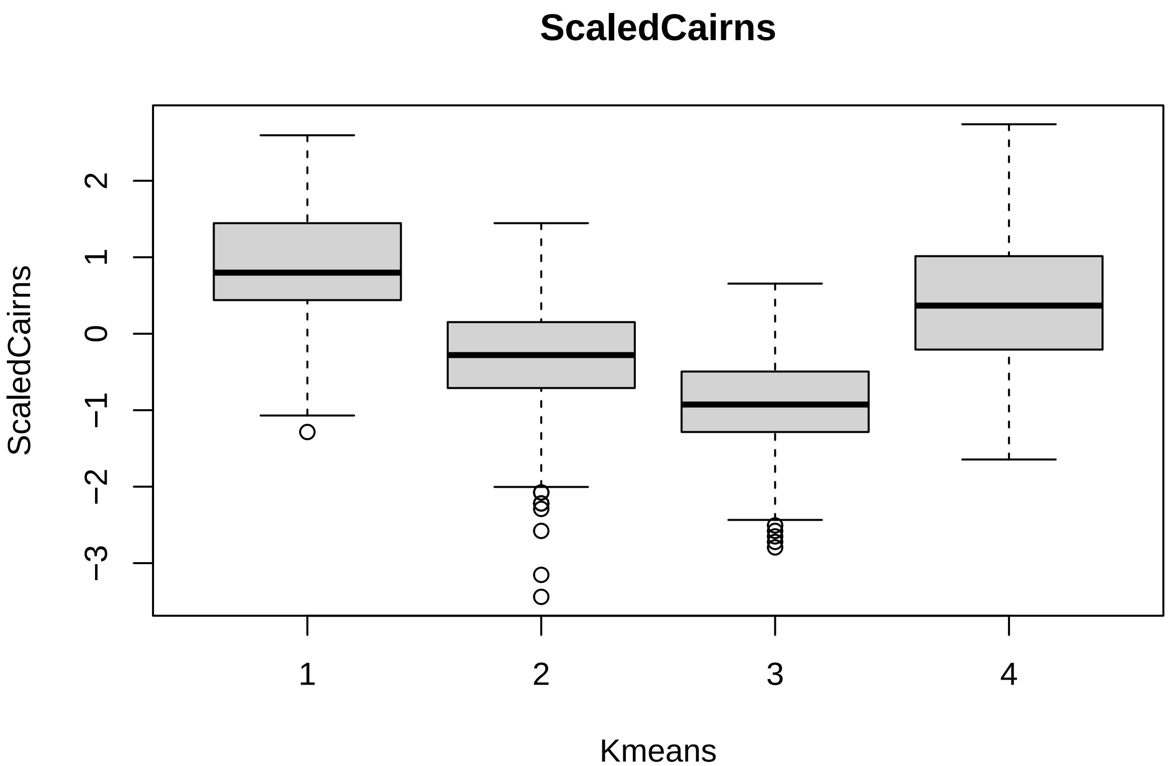

ScaledCairns
ScaledCairns
−2.0
−1.0
0.0
1.0
1
2
3
4
Kmeans

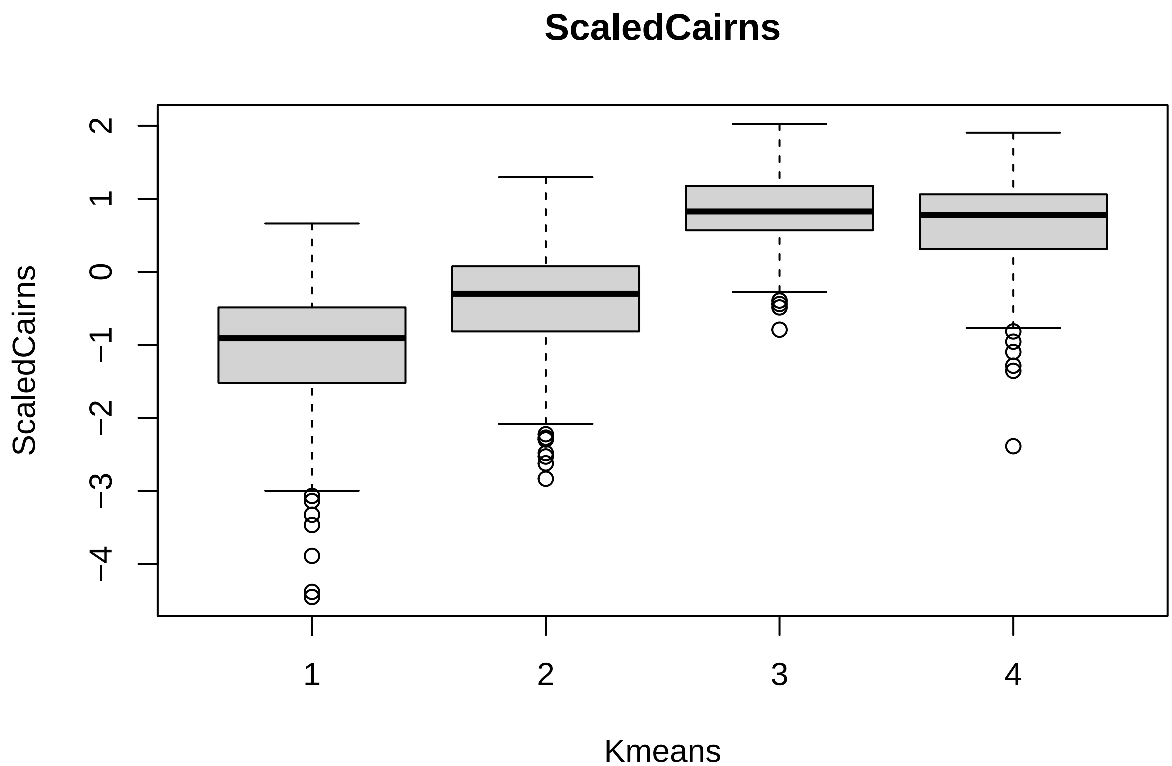

ScaledCairns
ScaledCairns
−2.0
−1.0
0.0
0.5
1.0
1
2
3
4
Kmeans

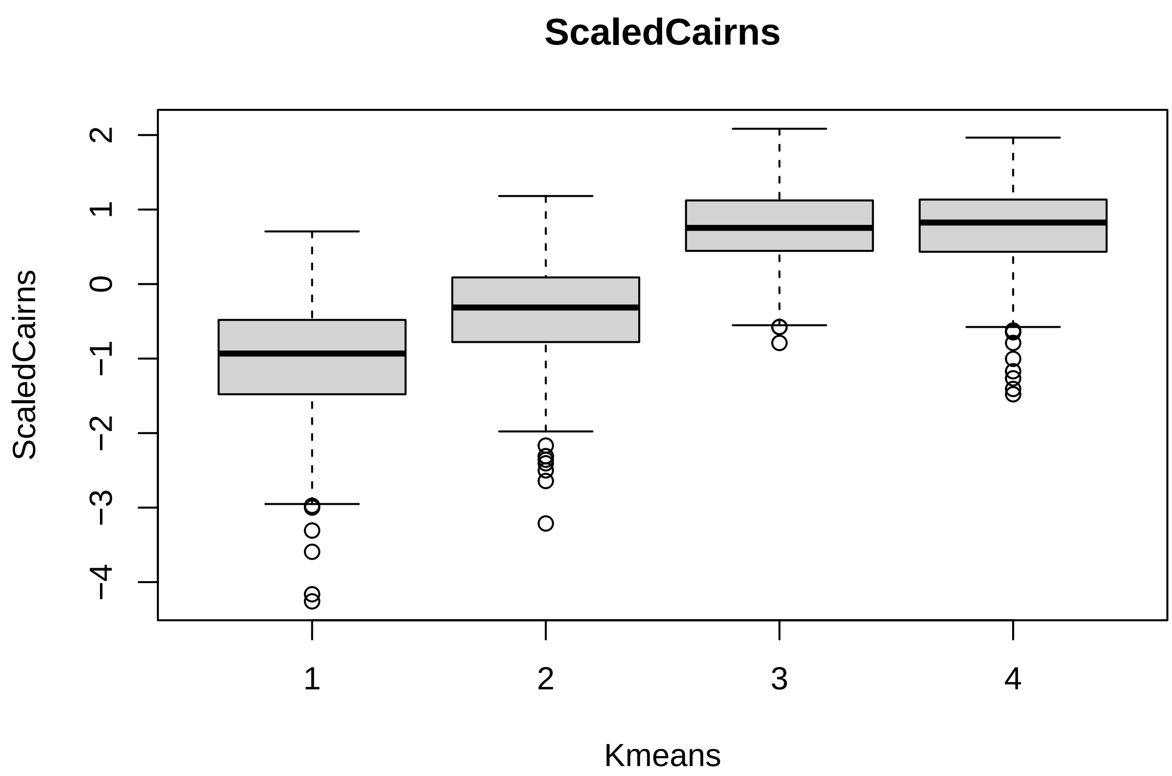

ScaledCairns
ScaledCairns
1.0
0.5
−0.5
−1.5
1
2
3
4
Kmeans


ScaledCairns
ScaledCairns
0
2
4
6
8
10
12
14
1
2
3
4
Kmeans


ScaledCairns
ScaledCairns
2
1
0
−1
−2
−3
−4
1
2
3
4
Kmeans


ScaledCairns
ScaledCairns
2
1
0
−1
−2
−3
1
2
3
4
Kmeans


ScaledCairns
ScaledCairns
2
1
0
−1
−2
−3
−4
1
2
3
4
Kmeans




```
fun04<-function(x) print(names(x))
lapply(Kmeans, fun04)

fun05<-function(x){x[,ncol(x)]
                   x$KMCluster<-x[,ncol(x)]
                   return(x$KMCluster)}

clusters<-lapply(Kmeans,fun05)

BrisbaneClusters<-as.data.frame(cbind(originaldata[[1]],as.factor(clusters[[1]]),
                                      as.factor(clusters[[2]])))
names(BrisbaneClusters)<-c(names(originaldata[[1]]),"KmeansDF","KmeansScaled")

PerthClusters<-as.data.frame(cbind(originaldata[[2]],as.factor(clusters[[3]]),
                                   as.factor(clusters[[4]])))

names(PerthClusters)<-c(names(originaldata[[3]]),"KmeansDF","KmeansScaled")

CairnsClusters<-as.data.frame(cbind(originaldata[[3]],as.factor(clusters[[5]]),
                                    as.factor(clusters[[6]])))
names(CairnsClusters)<-c(names(originaldata[[3]]),"KmeansDF","KmeansScaled")



funMetrics<-function(i){ tmp_df = listall[[i]]
```

```r
                      }
lapply(1:length(listall), funMetrics)
```

```r
write.csv(BrisbaneClusters,"BrisbaneClusters.csv")
write.csv(PerthClusters,"PerthClusters.csv")
write.csv(CairnsClusters,"CairnsClusters.csv")

DFClusters<-list(BrisbaneClusters,PerthClusters,CairnsClusters)


fun06<-function(x){tmpdf=DFClusters[[x]]
                   levels(tmpdf[,24])<-list(C1="1",C2="2",C3="3",C4="4")
                   levels(tmpdf[,25])<-list(G1="1",G2="2",G3="3",G4="4")
                   return(tmpdf)}
data<-lapply(1:length(DFClusters),fun06)
```

```r
funtableKmeans<-function(x){table(x$KmeansDF,x$KmeansScaled)}
funtabseason<-function(x){table(x$KmeansDF,x$Season) }
funtabseason2<-function(x){table(x$KmeansScaled,x$Season) }
funtabseason2<-function(x){table(x$KmeansScaled,x$RainTomorrow) }

lapply(data, funtableKmeans)
```

```
## [[1]]
##
##       G1  G2  G3  G4
##   C1  66   8 228   0
##   C2 534  20  44 113
##   C3   4  98   1 179
##   C4  36 190 263  13
##
## [[2]]
##
##       G1  G2  G3  G4
##   C1  54   0 259   0
##   C2   9 204 178 216
##   C3   2 109   2 327
##   C4 309 110  17   1
##
## [[3]]
##
##       G1  G2  G3  G4
##   C1 153  17   0 170
##   C2 186 280  70  99
##   C3   0 203 334  95
##   C4  33   0   0   4
```

```r
lapply(data,funtabseason)
```

```
## [[1]]
##
##      autumn spring summer winter
```

```
##   C1     48    104     15    135
##   C2    149    204    339     19
##   C3     97     48     96     41
##   C4    166     99      1    236
##
## [[2]]
##
##      autumn spring summer winter
##   C1     57     84     21    151
##   C2    149    198    231     29
##   C3    127     97    198     18
##   C4    127     76      1    233
##
## [[3]]
##
##      dry wet
##   C1 155 185
##   C2 268 367
##   C3 492 140
##   C4   3  34
```

```
lapply(data,funtabseason2)
```

```
## [[1]]
##
##       No Yes
##   G1 526 114
##   G2 182 134
##   G3 521  15
##   G4 117 188
##
## [[2]]
##
##       No Yes
##   G1 316  58
##   G2 397  26
##   G3 222 234
##   G4 520  24
##
## [[3]]
##
##       No Yes
##   G1 133 239
##   G2 381 119
##   G3 390  14
##   G4 200 168
```

```
#boxplot(BrisbaneClusters$Pressure9am ~ BrisbaneClusters$KmeansDF)

funtableKmeans<-function(x){table(x$KmeansDF,x$KmeansScaled)}
funtabseason<-function(x){table(x$KmeansDF,x$Season) }
funtabseason2<-function(x){table(x$KmeansScaled,x$Season) }
funtabseason2<-function(x){table(x$KmeansScaled,x$Season) }
```

```
lapply(data, funtableKmeans)
```

```
## [[1]]
##
##       G1  G2  G3  G4
##   C1  66   8 228   0
##   C2 534  20  44 113
##   C3   4  98   1 179
##   C4  36 190 263  13
##
## [[2]]
##
##       G1  G2  G3  G4
##   C1  54   0 259   0
##   C2   9 204 178 216
##   C3   2 109   2 327
##   C4 309 110  17   1
##
## [[3]]
##
##       G1  G2  G3  G4
##   C1 153  17   0 170
##   C2 186 280  70  99
##   C3   0 203 334  95
##   C4  33   0   0   4
```

```
lapply(data,funtabseason)
```

```
## [[1]]
##
##      autumn spring summer winter
##   C1     48    104     15    135
##   C2    149    204    339     19
##   C3     97     48     96     41
##   C4    166     99      1    236
##
## [[2]]
##
##      autumn spring summer winter
##   C1     57     84     21    151
##   C2    149    198    231     29
##   C3    127     97    198     18
##   C4    127     76      1    233
##
## [[3]]
##
##      dry wet
##   C1 155 185
##   C2 268 367
##   C3 492 140
##   C4   3  34
```

```r
lapply(data,funtabseason2)
```

```
## [[1]]
##
##      autumn spring summer winter
##   G1    147    198    289      6
##   G2    105     66      0    145
##   G3    117    141      0    278
##   G4     91     50    162      2
##
## [[2]]
##
##      autumn spring summer winter
##   G1     77     42      1    254
##   G2    153    186     33     51
##   G3    108    158     64    126
##   G4    122     69    353      0
##
## [[3]]
##
##      dry wet
##   G1  56 316
##   G2 141 359
##   G3 400   4
##   G4 321  47
```

```r
lapply(1:length(Kmeans), function(x){

  # Get the dataframe and the name
  tmp_df = Kmeans[[x]]
  tmp_name = names(Kmeans)[x]

  for (i in 1:17) {
                  kruskal.test(tmp_df[,13] ~ tmp_df[,18], data = tmp_df)
                  pairwise.wilcox.test(tmp_df[,13], tmp_df[,18], p.adjust.method = "BH")
                  #boxplot(tmp_df[,i] ~ tmp_df[,18], xlab = 'Kmeans', ylab = names(a)[i], main = tmp_
                }
  })
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
```

```
##
## [[6]]
## NULL
```

```
#kruskal.test(Murder ~ res.km.cluster, data = results)
# As the p-value is less than the significance level 0.05, we can conclude that there are significant d

#From the output of the Kruskal-Wallis test, we know that there is a significant difference between gro

#pairwise.wilcox.test(results$Murder, results$res.km.cluster, p.adjust.method = "BH")




funProfile<-function(x){catdes(x, num.var=18, prob = 0.01)
                        catdes(x, num.var=19, prob = 0.01)}


#lapply(temp,funProfile)
```