Homework_Week2

HT

5/24/2019

Question 4.2 : Clustering Anaysis

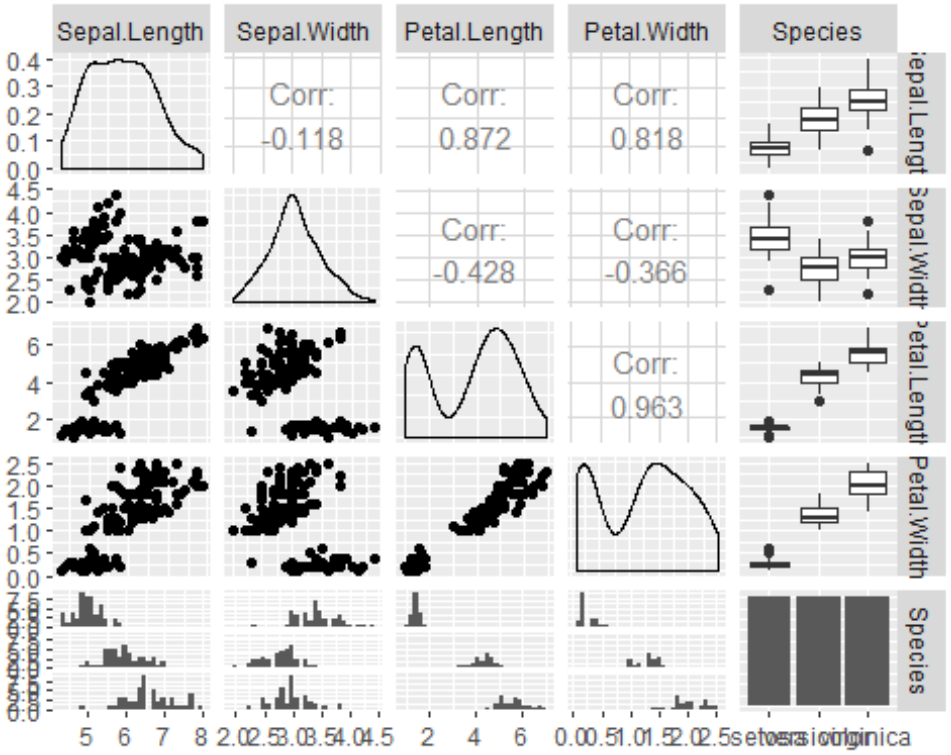
Loading libraries and dataset

```
library(ggplot2) # for generating plots
library(GGally) # for exploring data correlations (Ref: https://www.r-bloggers.com/plot-matrix-with-the-r-package-ggally/)
library(purrr) # for map_dbl function (Ref: https://uc-r.github.io/kmeans_clustering)


# Reading dataset and removing any na's
iris_df <- iris
iris_df <- na.omit(iris_df)

head(iris_df)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
# Plotting data to see the distribution of data points

ggpairs(iris_df)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#Looking at the GGpair plots histograms, the overlap between species
characteristics is minimum for Petal length and Petal width. So those will be
best combinations of predictors

# Scaling the data to minimize the effect of aboslute values, also dropping
the Species Column to make it all numerical for further analysis

iris_df_scaled <- scale(iris_df[,-5])

head(iris_df_scaled)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1   -0.8976739  1.01560199    -1.335752   -1.311052
## 2   -1.1392005 -0.13153881    -1.335752   -1.311052
## 3   -1.3807271  0.32731751    -1.392399   -1.311052
## 4   -1.5014904  0.09788935    -1.279104   -1.311052
## 5   -1.0184372  1.24503015    -1.335752   -1.311052
## 6   -0.5353840  1.93331463    -1.165809   -1.048667
```

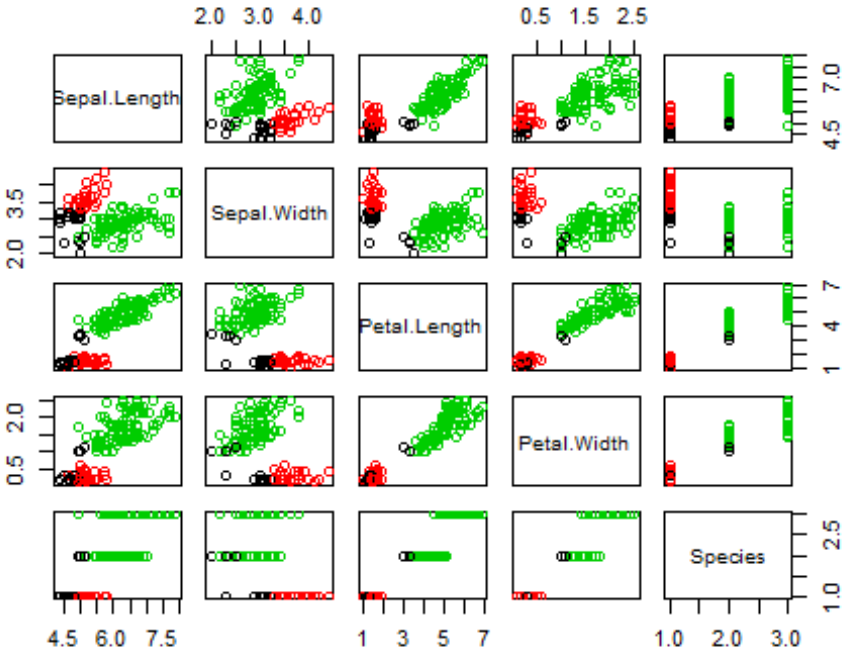
Analysis with Kmeans

```
# Setting seed as part of general best practise for improving repeatability
of analysis
set.seed(456)

# Running K-means on the scaled data with 3 known centers as raw data
```

```r
(Species)
iris_k <- kmeans(iris_df_scaled, 3)

# Exploring the k-means components (Cluster, Centers, Totss, Withinss,
tot.withninss,betweenss,size etc.)
iris_k
```

```
## K-means clustering with 3 clusters of sizes 21, 33, 96
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1   -1.3232208  -0.3718921   -1.1334386  -1.1111395
## 2   -0.8135055   1.3145538   -1.2825372  -1.2156393
## 3    0.5690971  -0.3705265    0.6888118   0.6609378
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   2   1   1   1   2   2   2   2   1   1   2   2   1   1   2   2   2   2
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   2   2   2   2   2   2   2   1   2   2   2   1   1   2   2   2   1   1
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   2   2   1   2   2   1   1   2   2   1   2   1   2   2   3   3   3   3
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   3   3   3   1   3   3   1   3   3   3   3   3   3   3   3   3   3   3
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   3   3   3   1   3   3   3   3   1   3   3   3   3   3   3   3   3   3
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## 145 146 147 148 149 150
##   3   3   3   3   3   3
##
## Within cluster sum of squares by cluster:
## [1]  23.15862  17.33362 149.25899
##  (between_SS / total_SS =  68.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
# Plotting the original dataframe using the cluster classification from K-
means components, this will serve as a baseline for this analysis.
plot(iris_df, col = iris_k$cluster)
```




```
# Expanding the analysis to higher K values, I plan to explore up to K= 15.
k_selection <- list()

for(i in 1:15){
  k_selection[[i]] <- kmeans(iris_df_scaled,i)
}

# Exploring the K means components for changing K values
k_selection

## [[1]]
## K-means clustering with 1 clusters of sizes 150
##
## Cluster means:
##    Sepal.Length  Sepal.Width Petal.Length   Petal.Width
## 1 -9.793092e-16 4.503805e-16 5.107026e-17 -6.217249e-17
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
```

```
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 145 146 147 148 149 150
##   1   1   1   1   1   1
##
## Within cluster sum of squares by cluster:
## [1] 596
##  (between_SS / total_SS =   0.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[2]]
## K-means clustering with 2 clusters of sizes 100, 50
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1    0.5055957  -0.4252069     0.650315   0.6253518
## 2   -1.0111914   0.8504137    -1.300630  -1.2507035
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   1   1   1   1
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 145 146 147 148 149 150
##   1   1   1   1   1   1
##
## Within cluster sum of squares by cluster:
## [1] 173.52867  47.35062
```

```
##  (between_SS / total_SS =  62.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[3]]
## K-means clustering with 3 clusters of sizes 53, 50, 47
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1  -0.05005221 -0.88042696    0.3465767   0.2805873
## 2  -1.01119138  0.85041372   -1.3006301  -1.2507035
## 3   1.13217737  0.08812645    0.9928284   1.0141287
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   3   3   3   1
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   1   1   3   1   1   1   1   1   1   1   1   3   1   1   1   1   3   1
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   1   1   1   3   3   3   1   1   1   1   1   1   1   3   3   1   1   1
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   1   1   1   1   1   1   1   1   1   1   3   1   3   3   3   3   1   3
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   3   3   3   3   3   1   1   3   3   3   3   1   3   1   3   1   3   3
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   1   3   3   3   3   3   3   1   1   3   3   3   1   3   3   3   1   3
## 145 146 147 148 149 150
##   3   3   1   3   3   1
##
## Within cluster sum of squares by cluster:
## [1] 44.08754 47.35062 47.45019
##  (between_SS / total_SS =  76.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[4]]
## K-means clustering with 4 clusters of sizes 22, 50, 49, 29
##
```

```
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1   -0.4201099  -1.4246794   0.03924137 -0.05279511
## 2    0.3558492  -0.3930869   0.58460377  0.54663615
## 3   -0.9987207   0.9032290  -1.29875725 -1.25214931
## 4    1.3926646   0.2323817   1.15674505  1.21327591
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   3   3   3   3   3   1   3   3   3   3   3   3   3   3   4   2   4   1
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   2   2   2   1   2   1   1   2   1   2   2   2   2   1   1   1   2   2
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   2   2   2   2   2   2   2   1   1   1   1   2   2   2   2   1   2   1
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   1   2   1   1   1   2   2   2   1   2   4   2   4   2   4   4   1   4
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   2   4   4   2   4   2   2   4   2   4   4   1   4   2   4   2   4   4
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   2   2   2   4   4   4   2   2   2   4   4   2   2   4   4   4   2   4
## 145 146 147 148 149 150
##   4   4   2   2   4   2
##
## Within cluster sum of squares by cluster:
## [1] 17.04641 29.59039 40.12172 26.89129
##  (between_SS / total_SS =  80.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[5]]
## K-means clustering with 5 clusters of sizes 25, 22, 49, 13, 41
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1    1.3340319  -0.2324872   1.05931109  0.95070960
## 2   -0.4201099  -1.4246794   0.03924137 -0.05279511
## 3   -0.9987207   0.9032290  -1.29875725 -1.25214931
## 4    1.1460128   0.7155805   1.14803003  1.43390232
## 5    0.2422139  -0.4001376   0.52118603  0.49044729
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
```

```
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   3   3   3   3   3   2   3   3   3   3   3   3   3   3   1   5   1   2
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   5   5   5   2   5   2   2   5   2   5   5   5   5   2   2   2   5   5
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   5   5   5   5   1   1   5   2   2   2   2   5   5   5   1   2   5   2
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   2   5   2   2   2   5   5   5   2   5   4   5   1   5   1   1   2   1
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   1   4   4   5   1   5   5   4   1   4   1   2   4   5   1   5   4   1
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   5   5   1   1   1   4   1   5   5   1   4   1   5   1   4   1   5   4
## 145 146 147 148 149 150
##   4   1   5   1   4   5
##
## Within cluster sum of squares by cluster:
## [1] 16.48474 17.04641 40.12172  9.53827 21.79555
##  (between_SS / total_SS =  82.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[6]]
## K-means clustering with 6 clusters of sizes 12, 13, 36, 22, 29, 38
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1    1.9704545   0.1552464   1.43998331  1.11601237
## 2   -0.5725419   1.9509630  -1.26603145 -1.20004296
## 3   -1.1526186   0.5248806  -1.31057490 -1.27096550
## 4   -0.4201099  -1.4246794   0.03924137 -0.05279511
## 5    0.8596404   0.1928251   0.85201975  1.05041603
## 6    0.2527555  -0.5360569   0.54703743  0.49112094
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   3   3   3   3   3   2   3   3   3   3   2   3   3   3   2   2   2   3
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   2   2   3   2   3   3   3   3   3   3   3   3   3   3   2   2   3   3
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   3   3   3   3   3   4   3   3   2   3   2   3   2   3   5   5   5   4
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   6   6   5   4   6   4   4   6   4   6   6   5   6   4   4   4   5   6
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
```

```
##   6   6   6   6   6   5   6   4   4   4   4   6   6   5   5   4   6   4
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   4   6   4   4   4   6   6   6   4   6   5   6   1   6   5   1   4   1
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   6   1   5   6   5   6   6   5   5   1   1   4   5   6   1   6   5   1
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   6   6   5   1   1   1   5   6   6   1   5   5   6   5   5   5   6   5
## 145 146 147 148 149 150
##   5   5   6   5   5   6
##
## Within cluster sum of squares by cluster:
## [1] 12.013666  4.340012 13.076892 17.046407 14.596105 19.109637
##  (between_SS / total_SS =  86.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[7]]
## K-means clustering with 7 clusters of sizes 11, 16, 19, 21, 16, 34, 33
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length   Petal.Width
## 1    2.0006454  0.05617514    1.4502829  1.0623426376
## 2    0.3476977 -0.84850181    0.8239401  0.8864251849
## 3    0.9455555  0.29109201    1.0106537  1.3404208948
## 4   -0.3628650 -1.40978142    0.1074147  0.0008746178
## 5   -0.6259564  1.80426129   -1.2826445 -1.2290567253
## 6   -1.1924784  0.40154427   -1.3090939 -1.2608902424
## 7    0.3831490 -0.16630065    0.4374915  0.3387951485
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   6   6   6   6   5   5   6   6   6   6   5   6   6   6   5   5   5   6
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   5   5   6   5   6   6   6   6   6   6   6   6   6   6   5   5   6   6
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   5   5   6   6   6   6   6   6   5   6   5   6   5   6   7   7   7   4
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   7   7   7   4   7   4   4   7   4   7   7   7   7   4   4   4   7   7
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   2   7   7   7   7   7   7   4   4   4   4   2   7   7   7   4   7   4
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   4   7   4   4   4   7   7   7   4   7   3   2   1   2   3   1   4   1
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   2   3   3   2   3   2   2   3   3   1   1   4   3   2   1   2   3   1
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   2   7   2   1   1   1   2   7   2   1   3   3   7   3   3   3   2   3
```

```
## 145 146 147 148 149 150
##   3   3   2   3   3   7
##
## Within cluster sum of squares by cluster:
## [1] 10.203534  5.606072  6.326077 11.951942  6.457580 15.974847 15.231839
##  (between_SS / total_SS =  88.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[8]]
## K-means clustering with 8 clusters of sizes 49, 17, 20, 11, 21, 18, 9, 5
##
## Cluster means:
##   Sepal.Length  Sepal.Width Petal.Length Petal.Width
## 1   -0.9987207  0.903229014   -1.2987572  -1.2521493
## 2    0.3312703 -0.765840197    0.8268558   0.8806373
## 3    0.7386689 -0.005353324    0.5732742   0.4928472
## 4   -0.7988675 -1.570679092   -0.1564505  -0.1541713
## 5   -0.1098371 -0.535769381    0.2476851   0.1258200
## 6    1.1351750  0.505761634    1.0875090   1.4002632
## 7    1.9201365 -0.309982937    1.4211008   1.0358391
## 8    0.3824171 -1.783421568    0.4543142   0.2107829
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   1   1   1   1   1   4   1   1   1   1   1   1   1   1   3   3   3   4
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   3   5   3   4   3   5   4   5   8   5   5   3   5   5   8   4   3   5
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   8   5   3   3   3   3   5   5   4   4   5   2   5   3   3   8   5   4
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   5   3   5   4   5   5   5   5   4   5   6   2   7   2   6   7   4   7
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   2   6   6   2   6   2   2   6   3   6   7   8   6   2   7   2   6   7
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   2   3   2   7   7   6   2   2   2   7   6   3   3   6   6   6   2   6
## 145 146 147 148 149 150
##   6   6   2   3   6   2
##
## Within cluster sum of squares by cluster:
## [1] 40.121722  5.944668  7.069615  7.182709  5.284445 12.220084  3.091184
## [8]  1.031789
```

```
##  (between_SS / total_SS =  86.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[9]]
## K-means clustering with 9 clusters of sizes 18, 16, 19, 9, 12, 13, 13, 31,
19
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1   -0.4884205 -1.32966365   0.04267456 -0.05014476
## 2   -1.4184656 -0.07418177  -1.32867068 -1.31925169
## 3    0.3544509 -1.01302596   0.77213731  0.79493556
## 4   -1.1794549  1.06658603  -1.30428072 -1.20901340
## 5   -0.5454476  1.99067167  -1.26494207 -1.21265764
## 6   -0.8233580  0.78617383  -1.29653403 -1.23031819
## 7    1.1460128  0.71558055   1.14803003  1.43390232
## 8    0.3099591 -0.18334517   0.43677167  0.34366839
## 9    1.4794564 -0.15568914   1.11202322  1.00208189
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   6   2   2   2   4   5   4   6   2   2   5   4   2   2   5   5   5   6
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   5   5   6   4   4   6   4   2   6   6   6   2   2   6   5   5   2   6
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   6   4   2   6   4   2   2   4   5   2   5   2   5   6   9   8   9   1
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   8   8   8   1   8   1   1   8   1   8   8   8   8   1   3   1   8   8
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   3   8   8   8   8   9   8   1   1   1   1   3   8   8   8   3   8   1
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   1   8   1   1   1   8   8   8   1   8   7   3   9   3   9   9   1   9
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   3   7   7   3   9   3   3   7   9   7   9   3   7   3   9   3   7   9
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   3   8   3   9   9   7   3   8   3   9   7   8   8   9   7   9   3   7
## 145 146 147 148 149 150
##   7   9   3   9   7   8
##
## Within cluster sum of squares by cluster:
## [1]  8.085437  4.847831  9.689095  1.339057  3.954505  1.342484  9.538270
## [8] 12.915283 11.114640
##  (between_SS / total_SS =  89.5 %)
##
## Available components:
```

```
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[10]]
## K-means clustering with 10 clusters of sizes 14, 7, 9, 3, 49, 21, 11, 16,
16, 4
##
## Cluster means:
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1    0.62049335 -0.80343557    0.8128129   0.6943216
## 2   -0.05233076 -0.62317058    0.7116564   1.0129324
## 3    1.92013650 -0.30998294    1.4211008   1.0358391
## 4    2.12140867  1.55093437    1.4966310   1.3565323
## 5   -0.99872072  0.90322901   -1.2987572  -1.2521493
## 6   -0.10983710 -0.53576938    0.2476851   0.1258200
## 7   -0.79886754 -1.57067909   -0.1564505  -0.1541713
## 8    0.74017841  0.08355009    0.5619447   0.4600490
## 9    0.92887107  0.26996047    0.9938831   1.3865973
## 10   0.34014997 -1.90960706    0.4061637   0.1648655
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   5   5   5   5   5   7   5   5   5   5   5   5   5   5   8   8   8   7
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   1   6   8   7   8   6   7   6  10   6   6   8   6   6  10   7   8   6
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   1   6   8   8   1   8   6   6   7   7   6   1   6   8   8  10   6   7
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   6   8   6   7   6   6   6   6   7   6   9   2   3   1   9   3   7   3
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   1   4   9   1   9   2   2   9   8   4   3  10   9   2   3   1   9   3
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   1   8   1   3   3   4   1   1   1   3   9   8   2   9   9   9   2   9
## 145 146 147 148 149 150
##   9   9   1   9   9   2
##
## Within cluster sum of squares by cluster:
##  [1]  4.0896197  1.6045130  3.0911844  0.7953180 40.1217221  5.2844452
##  [7]  7.1827088  5.1964555  3.7267717  0.5890659
##  (between_SS / total_SS =  88.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
```

```
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[11]]
## K-means clustering with 11 clusters of sizes 8, 4, 9, 14, 9, 24, 16, 4,
26, 18, 18
##
## Cluster means:
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1   -0.71652893  0.90088791  -1.30034685 -1.26185489
## 2   -1.16939131  1.18767311  -1.40656120 -1.31105215
## 3   -0.66956542  1.67839445  -1.23504470 -1.16528251
## 4   -0.59576562 -1.45894460   0.00760697 -0.05535081
## 5   -1.11236419  0.70969778  -1.25392725 -1.16528251
## 6    0.69740807 -0.03594375   0.61623204  0.53657805
## 7   -1.41846562 -0.07418177  -1.32867068 -1.31925169
## 8   -0.35423902  2.56424207  -1.33575163 -1.27825398
## 9    1.42005257  0.23025175   1.21557146  1.28757202
## 10  -0.05233076 -0.46293504   0.27241226  0.14664426
## 11   0.34350450 -1.04925145   0.75706440  0.79531916
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   1   7   7   7   2   3   5   5   7   7   3   5   7   7   8   8   3   1
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   3   3   1   3   2   5   5   7   5   1   1   7   7   1   8   8   7   5
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   1   2   7   1   2   7   7   5   3   7   3   7   3   5   6   6   6   4
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   6  10   6   4   6   4   4  10   4  10  10   6  10  10  11   4   6  10
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  11  10   6   6   6   6  10   4   4   4  10  11  10   6   6  11  10   4
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   4   6  10   4  10  10  10  10   4  10   9  11   9   6   9   9   4   9
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##  11   9   6  11   9  11  11   9   6   9   9  11   9  11   9  11   9   9
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##  11   6  11   9   9   9  11   6  11   9   9   6   6   9   9   9  11   9
## 145 146 147 148 149 150
##   9   9  11   6   9   6
##
## Within cluster sum of squares by cluster:
##  [1]  0.4661656  0.2651729  0.9603065  6.0463245  1.0295237  8.8544169
##  [7]  4.8478312  0.8678439 23.8238976  3.5796668  9.1215628
##  (between_SS / total_SS =  90.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
```

```
## [9] "ifault"
##
## [[12]]
## K-means clustering with 12 clusters of sizes 5, 16, 11, 9, 16, 16, 12, 12,
14, 8, 16, 15
##
## Cluster means:
##    Sepal.Length  Sepal.Width Petal.Length   Petal.Width
## 1  -1.139200483 -1.737535936   -0.4973664 -0.4189419473
## 2   0.981705010  0.355996029    1.0257474  1.4275949759
## 3   2.000645372  0.056175137    1.4502829  1.0623426376
## 4  -1.179454918  1.066586026   -1.3042807 -1.2090133997
## 5  -1.380727088  0.054871568   -1.3322112 -1.3274512328
## 6  -0.007044526 -0.289270672    0.3318136  0.2386613442
## 7  -0.807101403  0.824411857   -1.2838246 -1.2235903638
## 8  -0.545447581  1.990671673   -1.2649421 -1.2126576408
## 9  -0.336987119 -1.213128712    0.1330411  0.0008746178
## 10  0.430722444 -1.479429255    0.7177258  0.5420444088
## 11  0.325054555 -0.533038093    0.7814544  0.9110238118
## 12  0.873521219  0.006118084    0.5562799  0.4119450045
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   7   5   5   5   4   8   4   7   5   5   8   4   5   5   8   8   8   7
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   8   8   7   4   4   7   4   5   7   7   7   5   5   7   8   8   5   5
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   7   4   5   7   4   1   5   4   8   5   8   5   8   7  12  12  12   9
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  12   6  12   1  12   9   1   6   9   6   6  12   6   9  10   9   6   6
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  10   6  12  12  12  12   6   9   9   9   9  11   6  12  12  10   6   9
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   9   6   9   1   9   6   6   6   1   6   2  11   3  11   2   3   9   3
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##  10   2   2  11   2  10  11   2  12   3   3  10   2  11   3  11   2   3
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##  11  11  11   3   3   3  11  11  10   3   2  12  11   2   2   2  11   2
## 145 146 147 148 149 150
##   2   2  10  11   2  11
##
## Within cluster sum of squares by cluster:
##  [1]  2.836687  4.863671 10.203534  1.339057  2.061305  3.265036  1.040902
##  [8]  3.954505  4.286788  3.024726  4.330994  4.696717
##  (between_SS / total_SS =  92.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
```

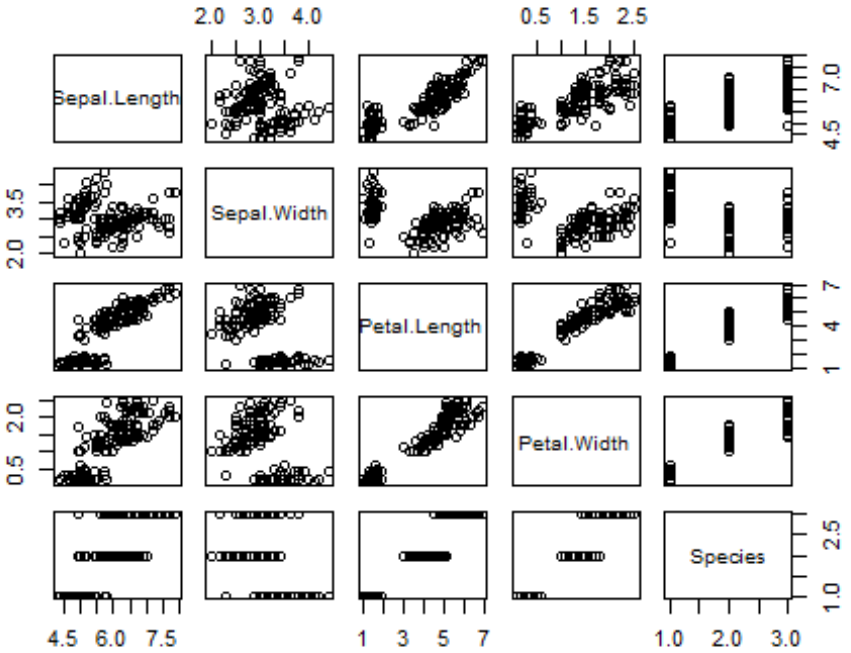
```
## [9] "ifault"
##
## [[13]]
## K-means clustering with 13 clusters of sizes 11, 24, 11, 25, 5, 9, 15, 6,
6, 3, 15, 10, 10
##
## Cluster means:
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1    -0.7988675 -1.57067909   -0.1564505  -0.1541713
## 2    -0.7165289  1.53181535   -1.2956262  -1.2126576
## 3     1.1772592  0.05617514    1.0280004   1.2770216
## 4    -1.2696248  0.29978613   -1.3017630  -1.2900613
## 5     0.3824171 -1.78342157    0.4543142   0.2107829
## 6     1.1553023 -0.08055478    0.5965627   0.3507218
## 7    -0.2616538 -0.59039513    0.2012880   0.0883364
## 8     0.7125035  0.59498370    1.0528910   1.5095904
## 9     2.0811542 -0.47568105    1.5249548   1.1378778
## 10    2.1214087  1.55093437    1.4966310   1.3565323
## 11    0.3341118 -0.81982329    0.8357417   0.9192234
## 12    0.3944935 -0.40685260    0.4203256   0.2501407
## 13    0.4307224  0.18966061    0.6695752   0.7224343
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   2   4   4   4   2   2   4   4   4   4   2   4   4   4   2   2   2   2
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   2   2   2   2   2   4   4   4   4   2   4   4   4   2   2   2   4   4
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   2   2   4   4   2   1   4   2   2   4   2   4   2   4   6  13   6   1
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  12   7  13   1   6   7   1  12   5  12   7   6   7   7   5   1  13  12
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   5  12  12   6   6   6  12   7   1   1   7  11   7  13   6   5   7   1
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   7  12   7   1   7   7   7  12   1   7   8  11   3  11   3   9   1   9
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##  11  10  13  11   3  11  11   8  13  10   9   5   3  11   9  11   8   3
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##  11  13  11   6   9  10  11  12  11   9   8  13  13   3   3   3  11   3
## 145 146 147 148 149 150
##   8   3  11   3   8  13
##
## Within cluster sum of squares by cluster:
##  [1]  7.1827088 11.5994552  1.9376294  6.1160617  1.0317894  1.7399963
##  [7]  3.2494748  0.7863023  1.3742228  0.7953180  5.0727841  1.2037341
## [13]  2.4479854
##  (between_SS / total_SS =  92.5 %)
##
## Available components:
##
```

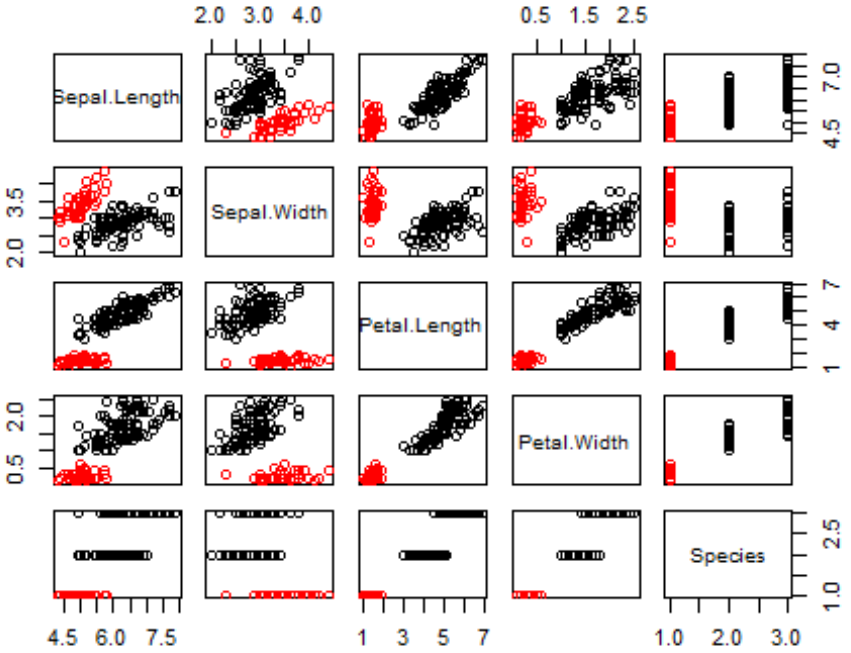
```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[14]]
## K-means clustering with 14 clusters of sizes 14, 3, 11, 13, 4, 4, 11, 12,
14, 16, 5, 16, 8, 19
##
## Cluster means:
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1   0.620493348 -0.80343557  0.812812881  0.69432162
## 2   2.121408674  1.55093437  1.496630966  1.35653228
## 3  -1.128222001  1.05731620 -1.310002702 -1.21563929
## 4  -0.656147275 -1.41986618 -0.002353056 -0.03949236
## 5   0.340149968 -1.90960706  0.406163665  0.16486546
## 6   1.638355466 -0.07418177  1.270040358  0.82082885
## 7  -0.809846023  0.74445962 -1.289403555 -1.22756590
## 8  -0.545447581  1.99067167 -1.264942069 -1.21265764
## 9   0.801638301  0.11427707  0.561944708  0.45067808
## 10 -1.418465620 -0.07418177 -1.328670678 -1.31925169
## 11  2.145561335 -0.49862387  1.541949088  1.20784724
## 12  0.928871066  0.26996047  0.993883054  1.38659726
## 13 -0.007044526 -0.56171661  0.703563838  0.98481969
## 14 -0.033262874 -0.44549314  0.283178631  0.15278193
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   3  10  10  10   3   8   3   7  10  10   8   3  10  10   8   8   8   3
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   8   8   7   3   3   7   3  10   7   7   7  10  10   7   8   8  10   7
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   7   3  10   7   3  10  10   3   8  10   8  10   8   7   9   9   9   4
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   1  14   9   4   9   4   4  14   5  14  14   9  14  14   5   4   9  14
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   1  14   9   9   1   9  14   4   4   4  14   1  14   9   9   5  14   4
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   4  14  14   4  14  14  14  14   4  14  12  13   6   1  12  11   4   6
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   1   2  12   1  12  13  13  12   9   2  11   5  12  13  11   1  12   6
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   1  13   1   6  11   2   1   1   1  11  12   9  13  12  12  12  13  12
## 145 146 147 148 149 150
##  12  12   1  12  12  13
##
## Within cluster sum of squares by cluster:
##  [1] 4.0896197 0.7953180 1.4856387 5.0410002 0.5890659 0.5435669 1.1922318
##  [8] 3.9545047 4.5056836 4.8478312 1.0784476 3.7267717 1.9787749 3.8605823
##  (between_SS / total_SS =  93.7 %)
##
```

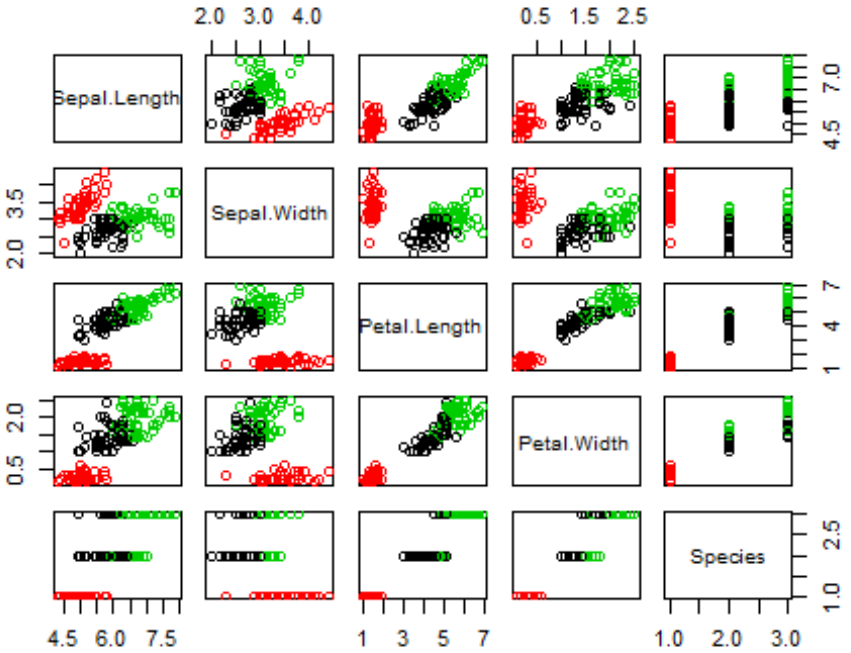
```
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## [[15]]
## K-means clustering with 15 clusters of sizes 3, 16, 16, 9, 6, 13, 12, 5,
5, 7, 15, 20, 5, 14, 4
##
## Cluster means:
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1    2.12140867  1.55093437    1.4966310  1.35653228
## 2   -1.38072709  0.05487157   -1.3322112 -1.32745123
## 3    0.34769767 -0.80548403    0.8310211  0.88642518
## 4   -0.66956542  1.67839445   -1.2350447 -1.16528251
## 5   -1.28009100  1.01560199   -1.3074278 -1.31105215
## 6   -0.37746273 -1.15514137    0.1327298  0.02105811
## 7    0.95403009 -0.05506276    0.4958558  0.33978903
## 8    0.38241712 -1.78342157    0.4543142  0.21078290
## 9    0.64809639  0.60263130    1.0434497  1.57518674
## 10   2.01789727 -0.42651788    1.4723534  1.05041603
## 11   1.08284428  0.09788935    1.0094612  1.19035489
## 12   0.03220355 -0.21183867    0.3863370  0.33541594
## 13  -1.13920048 -1.73753594   -0.4973664 -0.41894195
## 14  -0.83729223  0.85172473   -1.2871965 -1.18923038
## 15  -0.35423902  2.56424207   -1.3357516 -1.27825398
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  14   2   2   2   5   4   5  14   2   2   4   5   2   2  15  15   4  14
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   4   4  14   4   5  14   5   2  14  14  14   2   2  14  15  15   2   2
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  14   5   2  14  14  13   2  14   4   2   4   2   4  14   7   7   7   6
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   7  12   7  13   7   6  13  12   8  12  12   7  12   6   8   6  12  12
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   8  12   7   7   7   7  12   6   6   6   6   3  12  12   7   8  12   6
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   6  12   6  13   6  12  12  12  13  12   9   3  11   3  11  10   6  10
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   3   1  11   3  11   3   3   9  11   1  10   8  11   3  10   3  11  11
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   3  12   3  10  10   1   3   3   3  10   9  11  12  11  11  11   3  11
## 145 146 147 148 149 150
##   9  11   3  11   9  12
##
## Within cluster sum of squares by cluster:
##  [1] 0.7953180 2.0613055 5.4299508 0.9603065 0.6769948 3.3024815 2.6315868
```

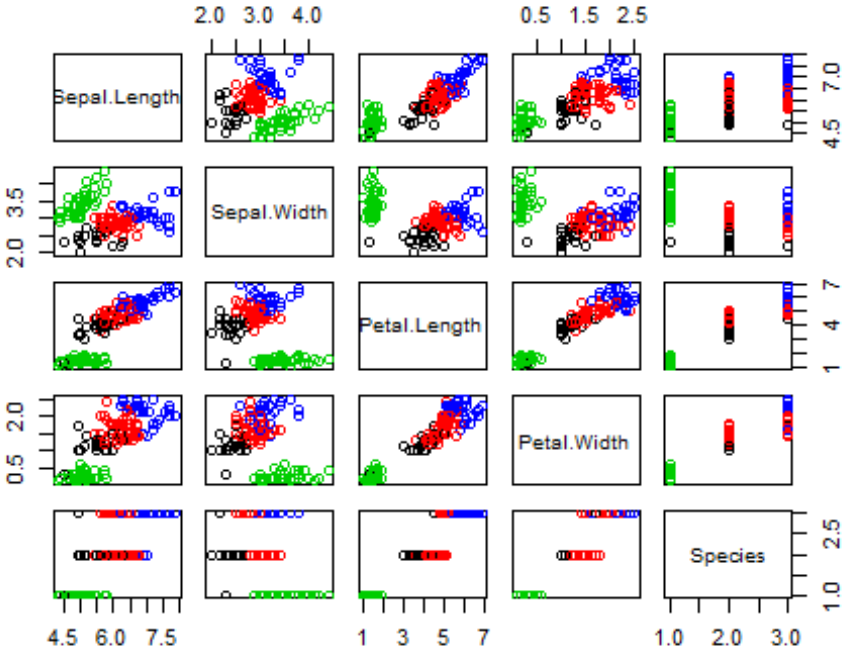
```
##  [8] 1.0317894 0.5283390 2.0812899 3.2107360 5.6272830 2.8366870 1.3721486
## [15] 0.8678439
##  (between_SS / total_SS =  94.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

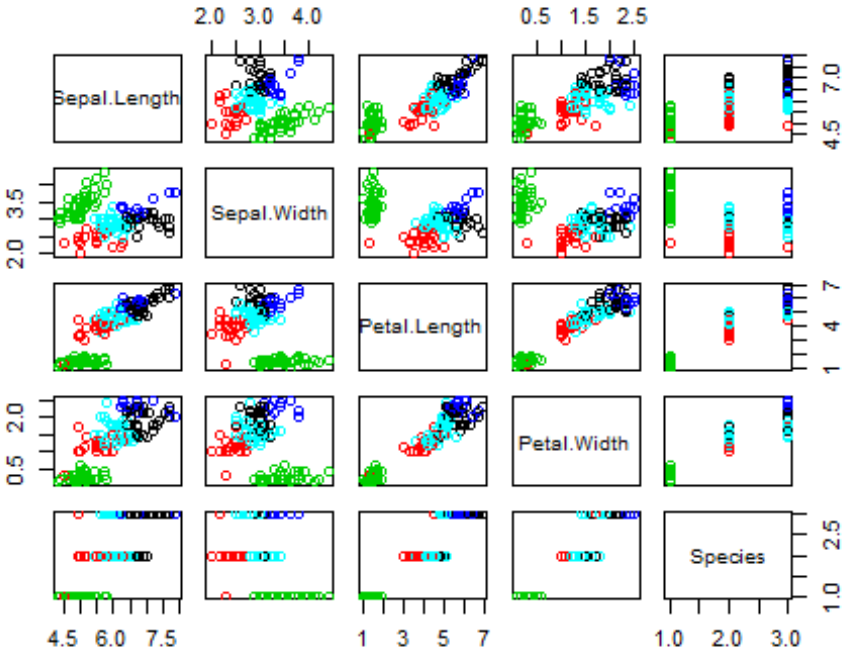
```r
# Plotting the data set with newly created clusters to observe how it changes
for (i in 1:15){
  plot(iris_df, col = k_selection[[i]]$cluster)
}
```

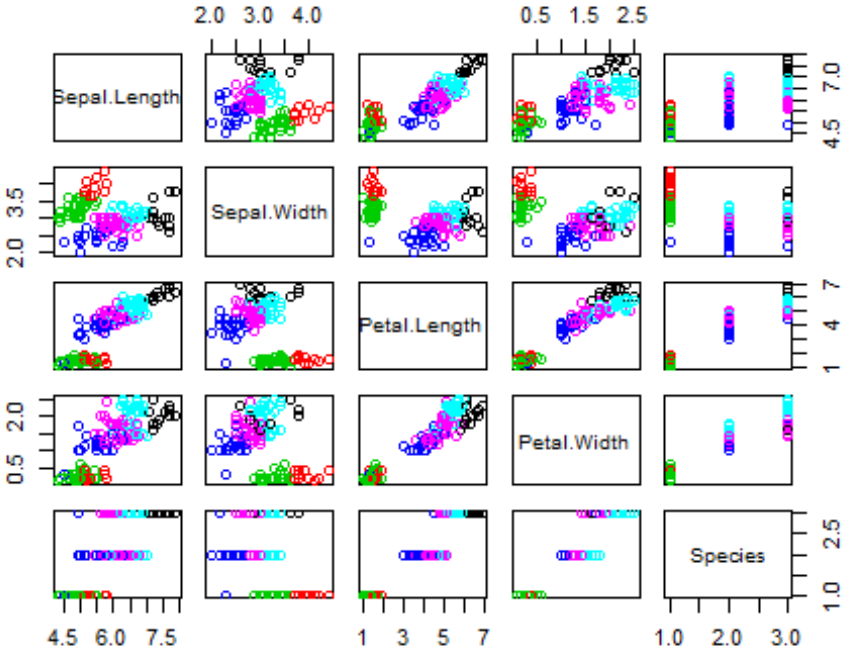

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

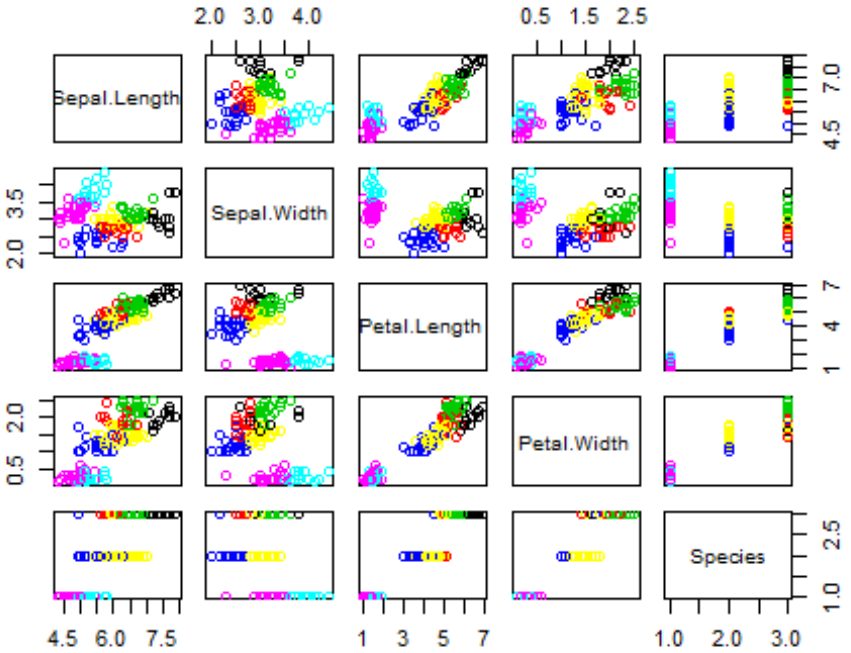

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

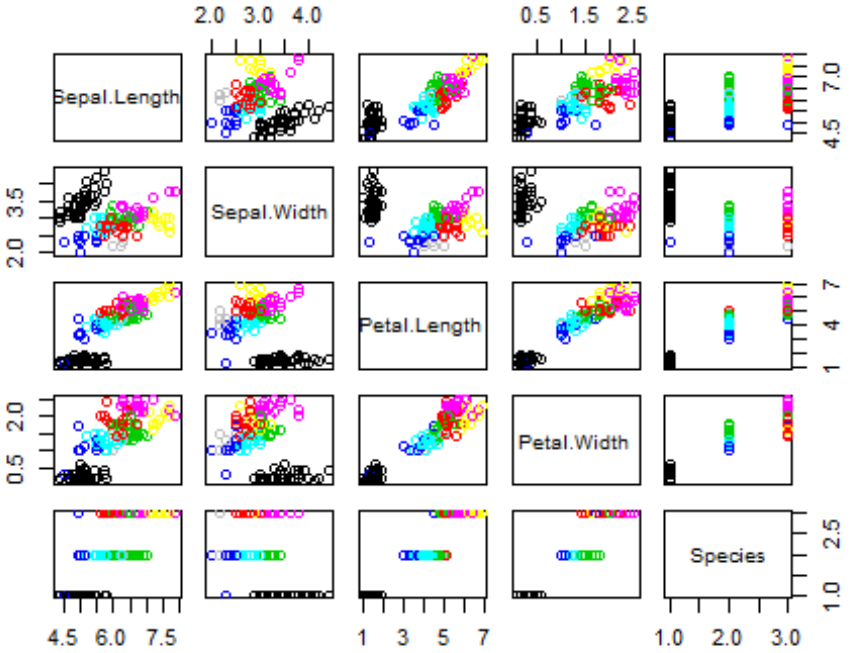

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

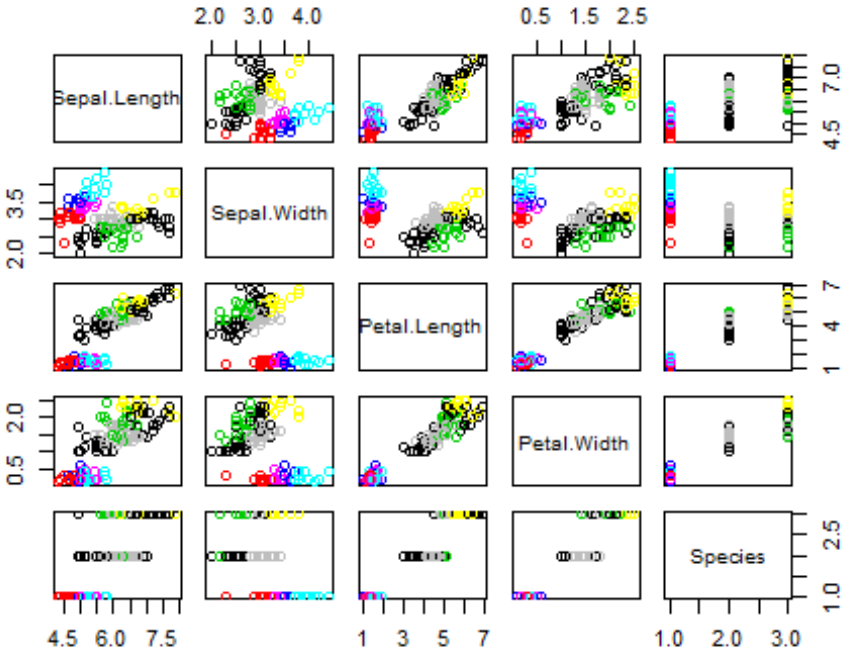

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

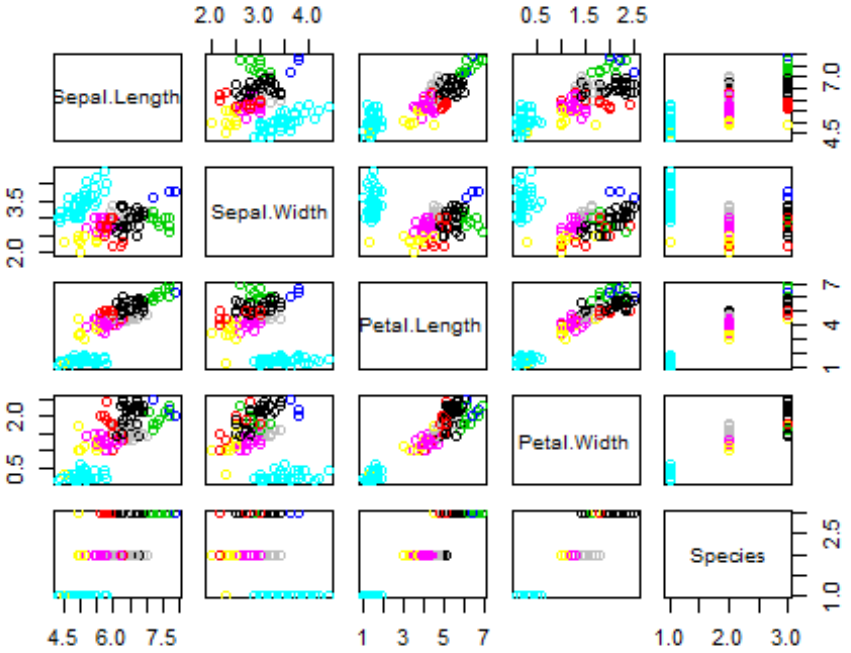

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

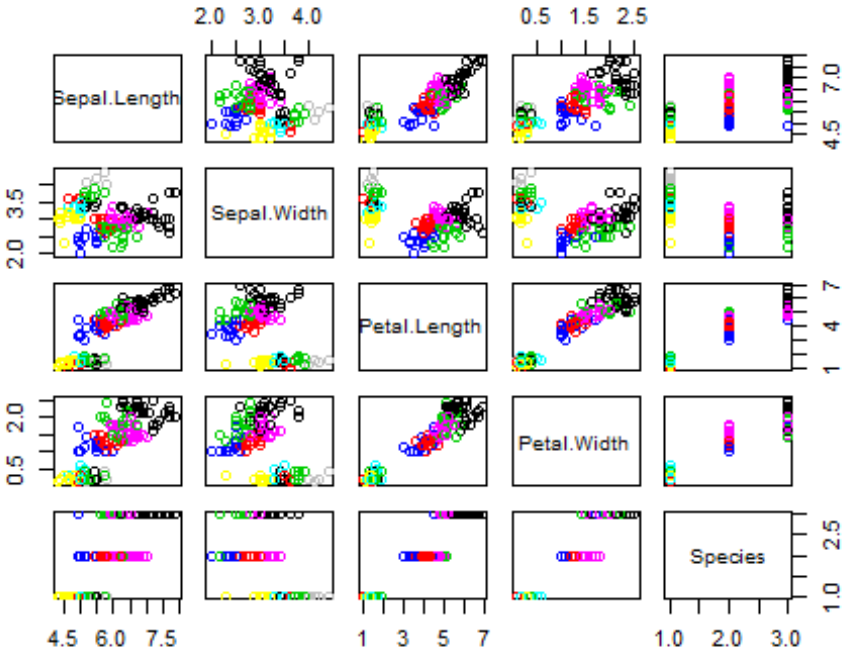

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

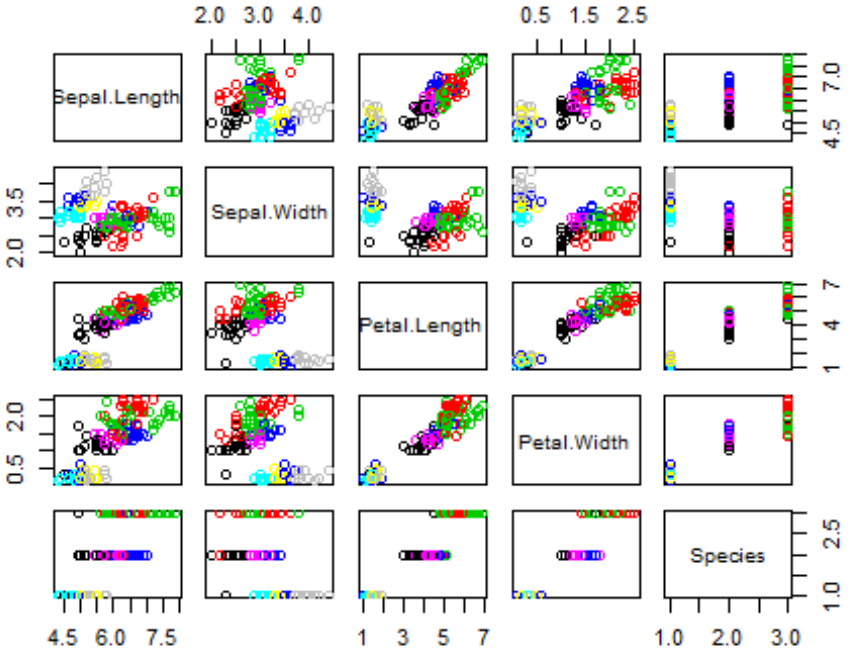

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

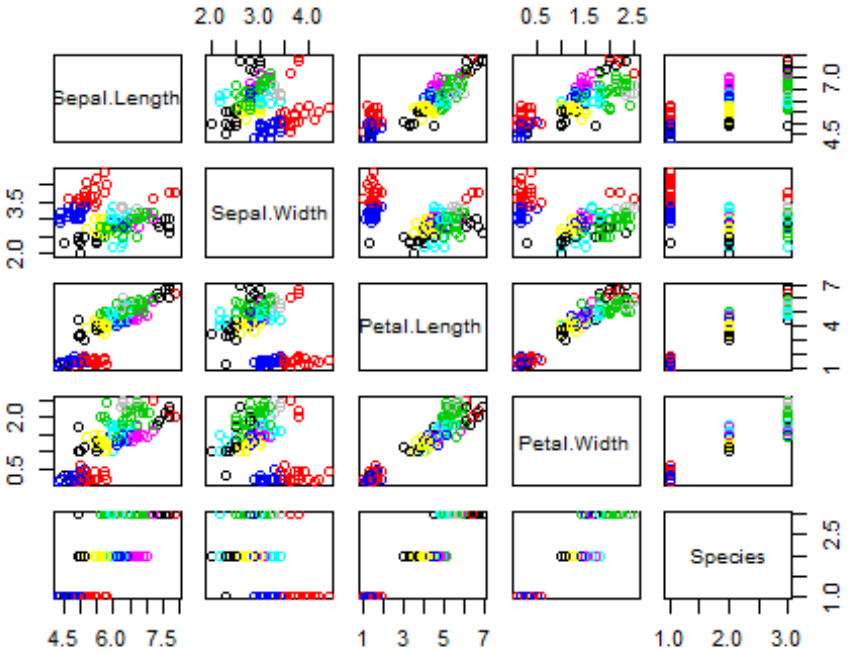

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

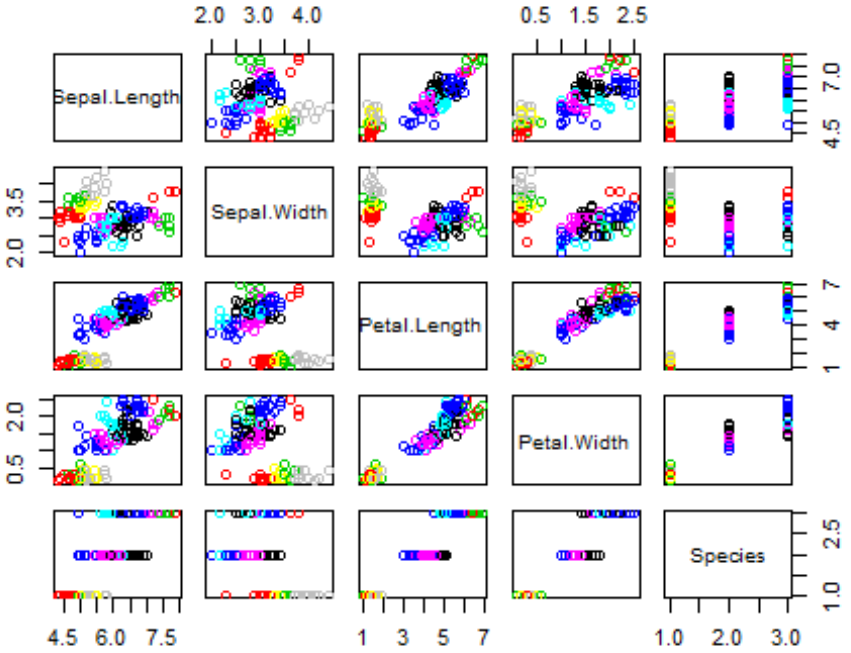


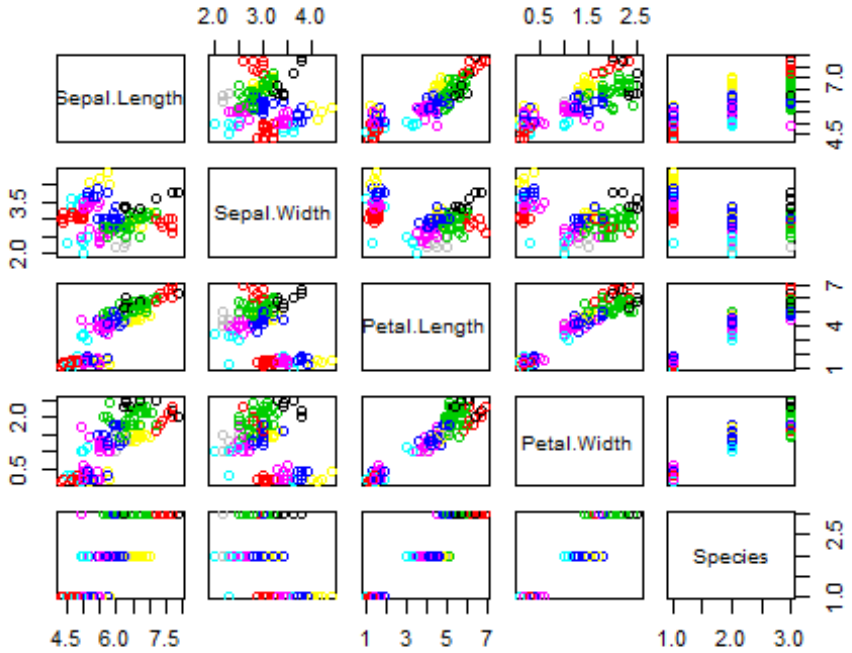




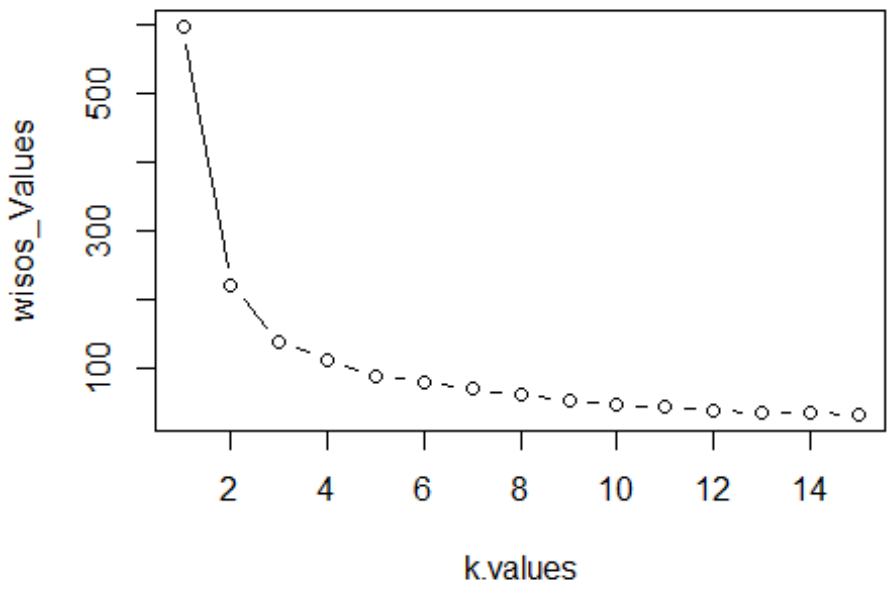



Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species
2.0 3.0 4.0
0.5 1.5 2.5
4.5 6.0 7.5
1 3 5 7
1.0 2.0 3.0

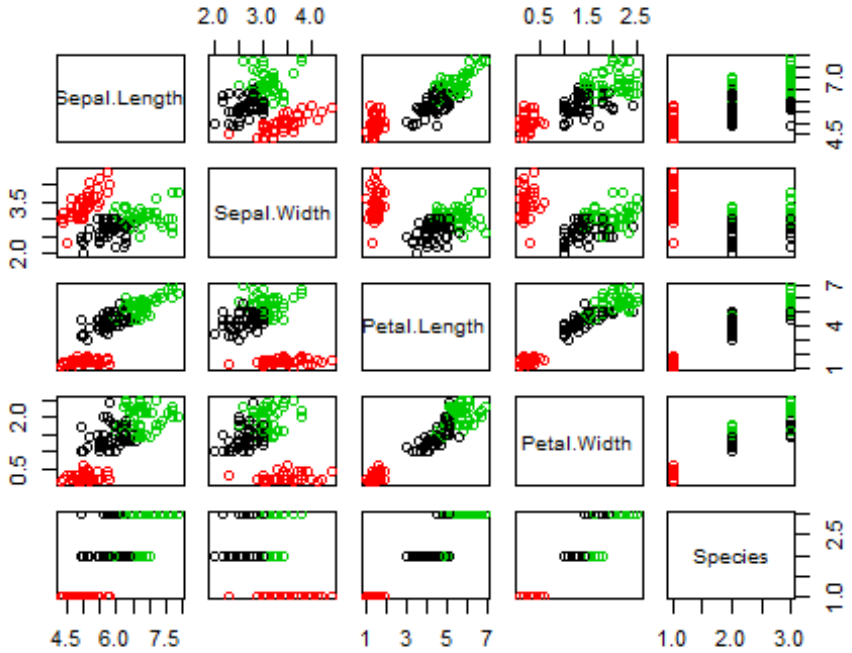

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species
2.0 3.0 4.0
0.5 1.5 2.5
4.5 6.0 7.5
1 3 5 7
1.0 2.0 3.0




```
# Upon plotting it was very clear that we can split the data into 15
clusters, but at this point it doesn't tell us the optimum number of
clusters.

# Using Within cluster sum of square to determine optimum number of clusters

wisos <- function(k){
  kmeans(iris_df_scaled, k, nstart = 10)$tot.withinss
}

# Computing and plotting wisos for K = 1 to 15

k.values <- 1:15

wisos_Values <- map_dbl(k.values,wisos)

# Visualizing the values of clusters wrt Sum of Squares
plot(k.values,wisos_Values,
     type = "b")
```

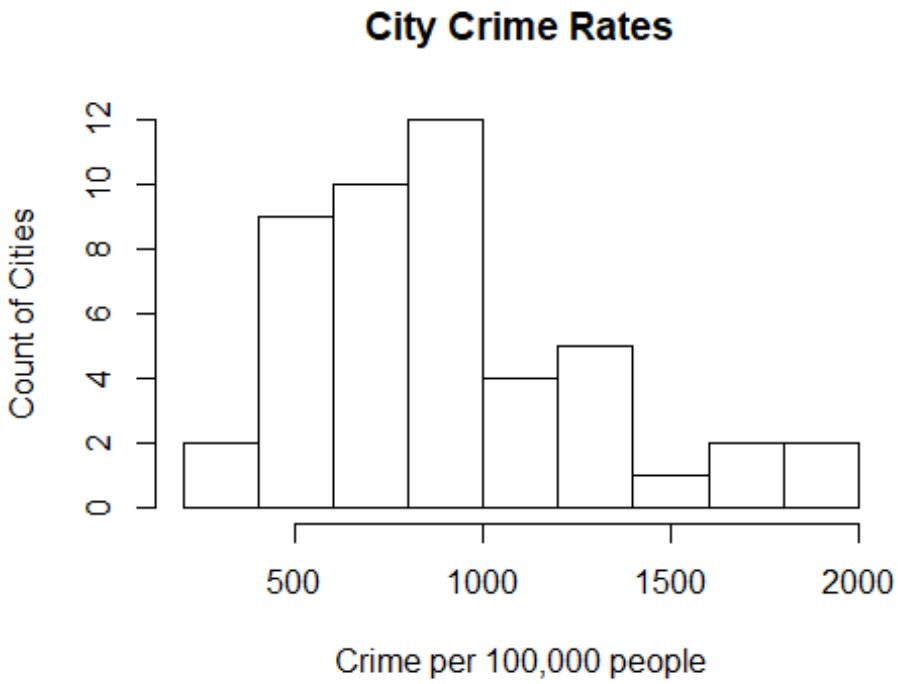



```
# From this visualization it is clear that 3 clusters will suffice and there
is not much significant improvement in center distance unless data is broken
into too many clusters.


# Plotting the data with final 3 clusters

plot(iris_df, col = k_selection[[3]]$cluster)
```

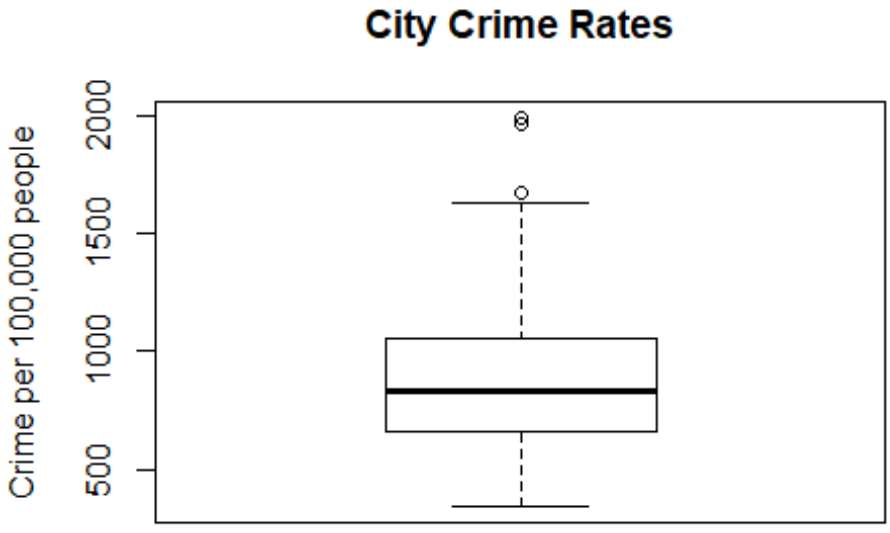


QUESTION 5.1 : Outlier Test

```
library(outliers) # To identify the outlier in dataset

# Reading the data and setting header
crime_df<- read.table('uscrime.txt', header = TRUE)

# create histogram for last column Crime (Crime per 100,000 people), this
will give an idea of distribution
hist(crime_df$Crime, breaks = 10, xlab = "Crime per 100,000 people", ylab =
"Count of Cities", main = "City Crime Rates")
```




```r
# Create boxplot to visualize outliers and in which direction they exist
boxplot(crime_df$Crime, ylab= "Crime per 100,000 people", main = "City Crime Rates")
```



```r
# Checking for the value of outlier
Crime_df_Outlier <- outlier(crime_df$Crime)

# Running grubbs test to confirm the outlier identified with Outlier
function. For Grubbs test I decided to run Type 10 as it was evident in Box
plot that the outlier is on top tail only and nothing on the bottom tail end.

grubbs.test(crime_df$Crime, type = 10)

##
##  Grubbs test for one outlier
##
## data:  crime_df$Crime
## G = 2.81290, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1993 is an outlier
```