R_Preprocessing.R

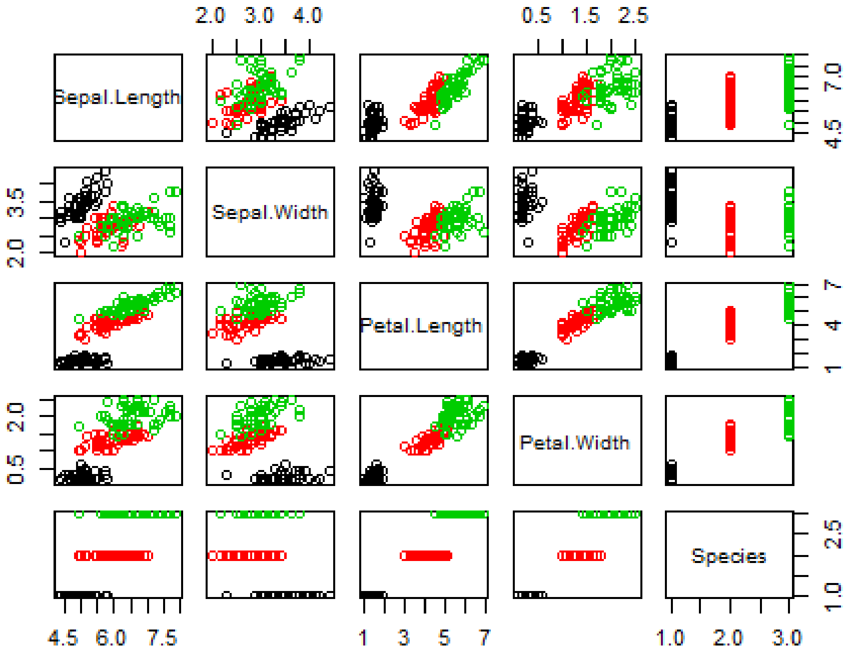
Wow

Tue Aug 21 08:11:05 2018

```
############################################################
#                 Data Preprocessing
############################################################
data(iris)
head(iris)
```

```
##   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
```

```
# inspect data (plot for data.frames actually uses pairs plot)
plot(iris, col=iris$Species)
```



```
# Black dots are setosa
# Red dots are versicolor
# Green dots are virginiga

######################### Sampling #########################
# Simple Random Sampling
#sample size = 20 without replacement
id1 <- sample(1:nrow(iris), 20, replace = FALSE)
id1

##  [1]   9  83 124  57 108  30  85  67  64  15  24 113 149  19 131 100  94
## [18] 128 122  53

#sample size = 20 with replacement
id2 <- sample(1:nrow(iris), 20, replace = TRUE)
id2

##  [1] 144  51 130  42 127  46 135  54   6   1  47 132  98  50 113  23  43
## [18] 122  30   6

#sample size = 20, draw from first 5 flowers with replacement and with
inequivalent probabilities
s1 <- sample(1:5, 20, replace = TRUE, prob = c(0.1,0.1,0.1,0.1,0.6))
s1

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

############################################################
# Stratified Sampling
library(sampling)
#srswor = Simple random sampling without replacement
#srswr = Simple random sampling with replacement
id3 <- strata(iris, stratanames="Species", size=c(5,5,5), method="srswor")
id3

##        Species ID_unit Prob Stratum
## 3       setosa       3  0.1       1
## 8       setosa       8  0.1       1
## 43      setosa      43  0.1       1
## 45      setosa      45  0.1       1
## 50      setosa      50  0.1       1
## 69  versicolor      69  0.1       2
## 71  versicolor      71  0.1       2
## 76  versicolor      76  0.1       2
## 86  versicolor      86  0.1       2
## 100 versicolor     100  0.1       2
## 104  virginica     104  0.1       3
## 110  virginica     110  0.1       3
## 117  virginica     117  0.1       3
## 119  virginica     119  0.1       3
## 133  virginica     133  0.1       3

s2 <- iris[id3$ID_unit,]
s2
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 3            4.7         3.2          1.3         0.2     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 43           4.4         3.2          1.3         0.2     setosa
## 45           5.1         3.8          1.9         0.4     setosa
## 50           5.0         3.3          1.4         0.2     setosa
## 69           6.2         2.2          4.5         1.5 versicolor
## 71           5.9         3.2          4.8         1.8 versicolor
## 76           6.6         3.0          4.4         1.4 versicolor
## 86           6.0         3.4          4.5         1.6 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
## 104          6.3         2.9          5.6         1.8  virginica
## 110          7.2         3.6          6.1         2.5  virginica
## 117          6.5         3.0          5.5         1.8  virginica
## 119          7.7         2.6          6.9         2.3  virginica
## 133          6.4         2.8          5.6         2.2  virginica

####################### Discretization #######################
plot(iris$Sepal.Width, 1:150, ylab="index")

library(arules)

## Loading required package: Matrix

## 
## Attaching package: 'arules'

## The following objects are masked from 'package:base':
## 
##     abbreviate, write

# Equal width approach
ew1 <- discretize(iris$Sepal.Width, method="interval", categories=3)

## Warning in discretize(iris$Sepal.Width, method = "interval", categories
## = 3): Parameter categories is deprecated. Use breaks instead! Also, the
## default method is now frequency!

ew1

##   [1] [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6) [3.6,4.4]
[2.8,3.6)
##   [8] [2.8,3.6) [2.8,3.6) [2.8,3.6) [3.6,4.4] [2.8,3.6) [2.8,3.6)
[2.8,3.6)
##  [15] [3.6,4.4] [3.6,4.4] [3.6,4.4] [2.8,3.6) [3.6,4.4] [3.6,4.4]
[2.8,3.6)
##  [22] [3.6,4.4] [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6)
[2.8,3.6)
##  [29] [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6) [3.6,4.4] [3.6,4.4]
[2.8,3.6)
##  [36] [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6) [2,2.8)
##  [43] [2.8,3.6) [2.8,3.6) [3.6,4.4] [2.8,3.6) [3.6,4.4] [2.8,3.6)
```

```
[3.6,4.4]
##  [50] [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6) [2,2.8)   [2,2.8)   [2,2.8)
##  [57] [2.8,3.6) [2,2.8)   [2.8,3.6) [2,2.8)   [2,2.8)   [2.8,3.6) [2,2.8)
##  [64] [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6) [2,2.8)   [2,2.8)   [2,2.8)
##  [71] [2.8,3.6) [2,2.8)   [2,2.8)   [2,2.8)   [2.8,3.6) [2.8,3.6) [2,2.8)
##  [78] [2.8,3.6) [2.8,3.6) [2,2.8)   [2,2.8)   [2,2.8)   [2,2.8)   [2,2.8)
##  [85] [2.8,3.6) [2.8,3.6) [2.8,3.6) [2,2.8)   [2.8,3.6) [2,2.8)   [2,2.8)
##  [92] [2.8,3.6) [2,2.8)   [2,2.8)   [2,2.8)   [2.8,3.6) [2.8,3.6)
[2.8,3.6)
##  [99] [2,2.8)   [2,2.8)   [2.8,3.6) [2,2.8)   [2.8,3.6) [2.8,3.6)
[2.8,3.6)
## [106] [2.8,3.6) [2,2.8)   [2.8,3.6) [2,2.8)   [2.8,3.6) [2.8,3.6) [2,2.8)
## [113] [2.8,3.6) [2,2.8)   [2,2.8)   [2.8,3.6) [2.8,3.6) [3.6,4.4] [2,2.8)
## [120] [2,2.8)   [2.8,3.6) [2,2.8)   [2,2.8)   [2,2.8)   [2.8,3.6)
[2.8,3.6)
## [127] [2,2.8)   [2.8,3.6) [2,2.8)   [2.8,3.6) [2,2.8)   [3.6,4.4] [2,2.8)
## [134] [2,2.8)   [2,2.8)   [2.8,3.6) [2.8,3.6) [2.8,3.6) [2.8,3.6)
[2.8,3.6)
## [141] [2.8,3.6) [2.8,3.6) [2,2.8)   [2.8,3.6) [2.8,3.6) [2.8,3.6) [2,2.8)
## [148] [2.8,3.6) [2.8,3.6) [2.8,3.6)
## attr(,"discretized:breaks")
## [1] 2.0 2.8 3.6 4.4
## attr(,"discretized:method")
## [1] interval
## Levels: [2,2.8) [2.8,3.6) [3.6,4.4]
```

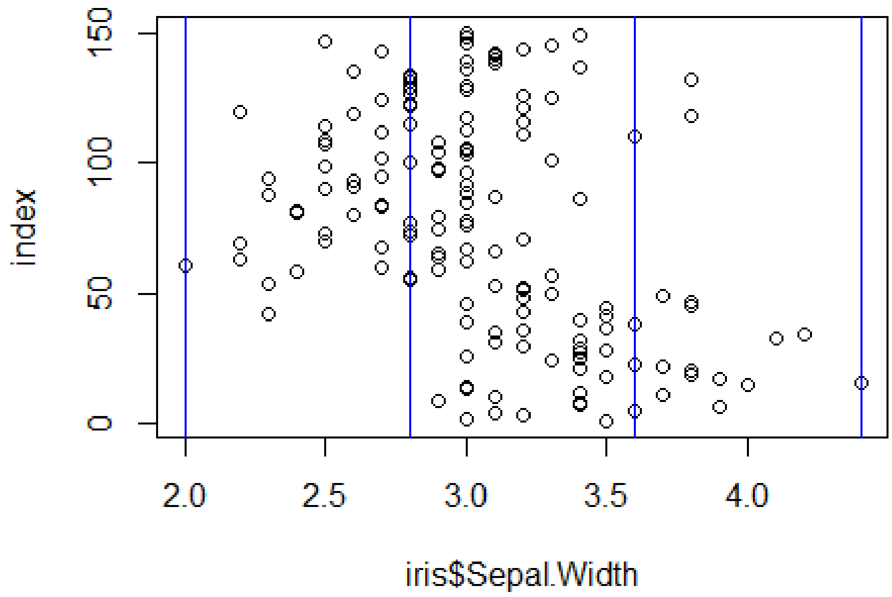
```
#Show only split points
sp1 <- discretize(iris$Sepal.Width, method="interval", categories=3,
onlycuts=TRUE) #get split points
```

```
## Warning in discretize(iris$Sepal.Width, method = "interval", categories
## = 3, : Parameter categories is deprecated. Use breaks instead! Also, the
## default method is now frequency!
```

```
sp1
```

```
## [1] 2.0 2.8 3.6 4.4
```

```
plot(iris$Sepal.Width, 1:150, ylab="index")
abline(v=sp1, col="blue") #add straight lines to the current plot
```



```
# Equal frequency approach
ew2 <- discretize(iris$Sepal.Width, method="frequency", categories=3)

## Warning in discretize(iris$Sepal.Width, method = "frequency", categories
## = 3): Parameter categories is deprecated. Use breaks instead! Also, the
## default method is now frequency!

ew2

##   [1] [3.2,4.4] [2.9,3.2) [3.2,4.4] [2.9,3.2) [3.2,4.4] [3.2,4.4]
[3.2,4.4]
##   [8] [3.2,4.4] [2.9,3.2) [2.9,3.2) [3.2,4.4] [3.2,4.4] [2.9,3.2)
[2.9,3.2)
##  [15] [3.2,4.4] [3.2,4.4] [3.2,4.4] [3.2,4.4] [3.2,4.4] [3.2,4.4]
[3.2,4.4]
##  [22] [3.2,4.4] [3.2,4.4] [3.2,4.4] [3.2,4.4] [2.9,3.2) [3.2,4.4]
[3.2,4.4]
##  [29] [3.2,4.4] [3.2,4.4] [2.9,3.2) [3.2,4.4] [3.2,4.4] [3.2,4.4]
[2.9,3.2)
##  [36] [3.2,4.4] [3.2,4.4] [3.2,4.4] [2.9,3.2) [3.2,4.4] [3.2,4.4] [2,2.9)
##  [43] [3.2,4.4] [3.2,4.4] [3.2,4.4] [2.9,3.2) [3.2,4.4] [3.2,4.4]
[3.2,4.4]
##  [50] [3.2,4.4] [3.2,4.4] [3.2,4.4] [2.9,3.2) [2,2.9)   [2,2.9)   [2,2.9)
##  [57] [3.2,4.4] [2,2.9)   [2.9,3.2) [2,2.9)   [2,2.9)   [2.9,3.2) [2,2.9)
##  [64] [2.9,3.2) [2.9,3.2) [2.9,3.2) [2.9,3.2) [2,2.9)   [2,2.9)   [2,2.9)
##  [71] [3.2,4.4] [2,2.9)   [2,2.9)   [2,2.9)   [2.9,3.2) [2.9,3.2) [2,2.9)
##  [78] [2.9,3.2) [2.9,3.2) [2,2.9)   [2,2.9)   [2,2.9)   [2,2.9)   [2,2.9)
##  [85] [2.9,3.2) [3.2,4.4] [2.9,3.2) [2,2.9)   [2.9,3.2) [2,2.9)   [2,2.9)
##  [92] [2.9,3.2) [2,2.9)   [2,2.9)   [2,2.9)   [2.9,3.2) [2.9,3.2)
[2.9,3.2)
##  [99] [2,2.9)   [2,2.9)   [3.2,4.4] [2,2.9)   [2.9,3.2) [2.9,3.2)
[2.9,3.2)
## [106] [2.9,3.2) [2,2.9)   [2.9,3.2) [2,2.9)   [3.2,4.4] [3.2,4.4] [2,2.9)
## [113] [2.9,3.2) [2,2.9)   [2,2.9)   [3.2,4.4] [2.9,3.2) [3.2,4.4] [2,2.9)
## [120] [2,2.9)   [3.2,4.4] [2,2.9)   [2,2.9)   [2,2.9)   [3.2,4.4]
[3.2,4.4]
## [127] [2,2.9)   [2.9,3.2) [2,2.9)   [2.9,3.2) [2,2.9)   [3.2,4.4] [2,2.9)
## [134] [2,2.9)   [2,2.9)   [2.9,3.2) [3.2,4.4] [2.9,3.2) [2.9,3.2)
[2.9,3.2)
## [141] [2.9,3.2) [2.9,3.2) [2,2.9)   [3.2,4.4] [3.2,4.4] [2.9,3.2) [2,2.9)
## [148] [2.9,3.2) [3.2,4.4] [2.9,3.2)
## attr(,"discretized:breaks")
## [1] 2.0 2.9 3.2 4.4
## attr(,"discretized:method")
## [1] frequency
## Levels: [2,2.9) [2.9,3.2) [3.2,4.4]
```

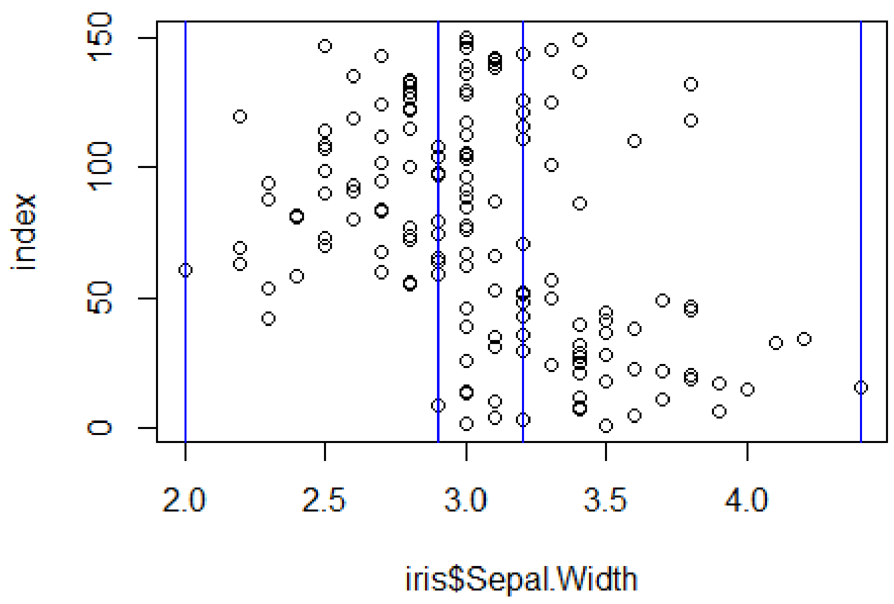
```
sp2 <- discretize(iris$Sepal.Width, method="frequency", categories=3,
onlycuts=TRUE) #get split points
```

```
## Warning in discretize(iris$Sepal.Width, method = "frequency", categories
## = 3, : Parameter categories is deprecated. Use breaks instead! Also, the
## default method is now frequency!
```

```
sp2
```

```
## [1] 2.0 2.9 3.2 4.4
```

```
plot(iris$Sepal.Width, 1:150, ylab="index")
abline(v=sp2, col="blue") #add straight lines to the current plot
```



```
# K-means approach
ew3 <- discretize(iris$Sepal.Width, method="cluster", categories=3)

## Warning in discretize(iris$Sepal.Width, method = "cluster", categories
## = 3): Parameter categories is deprecated. Use breaks instead! Also, the
## default method is now frequency!

ew3

##   [1] [3.35,4.4]  [2.83,3.35) [2.83,3.35) [2.83,3.35) [3.35,4.4]
##   [6] [3.35,4.4]  [3.35,4.4]  [3.35,4.4]  [2.83,3.35) [2.83,3.35)
##  [11] [3.35,4.4]  [3.35,4.4]  [2.83,3.35) [2.83,3.35) [3.35,4.4]
##  [16] [3.35,4.4]  [3.35,4.4]  [3.35,4.4]  [3.35,4.4]  [3.35,4.4]
##  [21] [3.35,4.4]  [3.35,4.4]  [3.35,4.4]  [2.83,3.35) [3.35,4.4]
##  [26] [2.83,3.35) [3.35,4.4]  [3.35,4.4]  [3.35,4.4]  [2.83,3.35)
##  [31] [2.83,3.35) [3.35,4.4]  [3.35,4.4]  [3.35,4.4]  [2.83,3.35)
##  [36] [2.83,3.35) [3.35,4.4]  [3.35,4.4]  [2.83,3.35) [3.35,4.4]
##  [41] [3.35,4.4]  [2,2.83)    [2.83,3.35) [3.35,4.4]  [3.35,4.4]
##  [46] [2.83,3.35) [3.35,4.4]  [2.83,3.35) [3.35,4.4]  [2.83,3.35)
##  [51] [2.83,3.35) [2.83,3.35) [2.83,3.35) [2,2.83)    [2,2.83)
##  [56] [2,2.83)    [2.83,3.35) [2,2.83)    [2.83,3.35) [2,2.83)
##  [61] [2,2.83)    [2.83,3.35) [2,2.83)    [2.83,3.35) [2.83,3.35)
##  [66] [2.83,3.35) [2.83,3.35) [2,2.83)    [2,2.83)    [2,2.83)
##  [71] [2.83,3.35) [2,2.83)    [2,2.83)    [2,2.83)    [2.83,3.35)
##  [76] [2.83,3.35) [2,2.83)    [2.83,3.35) [2.83,3.35) [2,2.83)
##  [81] [2,2.83)    [2,2.83)    [2,2.83)    [2,2.83)    [2.83,3.35)
##  [86] [3.35,4.4]  [2.83,3.35) [2,2.83)    [2.83,3.35) [2,2.83)
##  [91] [2,2.83)    [2.83,3.35) [2,2.83)    [2,2.83)    [2,2.83)
##  [96] [2.83,3.35) [2.83,3.35) [2.83,3.35) [2,2.83)    [2,2.83)
## [101] [2.83,3.35) [2,2.83)    [2.83,3.35) [2.83,3.35) [2.83,3.35)
## [106] [2.83,3.35) [2,2.83)    [2.83,3.35) [2,2.83)    [3.35,4.4]
## [111] [2.83,3.35) [2,2.83)    [2.83,3.35) [2,2.83)    [2,2.83)
## [116] [2.83,3.35) [2.83,3.35) [3.35,4.4]  [2,2.83)    [2,2.83)
## [121] [2.83,3.35) [2,2.83)    [2,2.83)    [2,2.83)    [2.83,3.35)
## [126] [2.83,3.35) [2,2.83)    [2.83,3.35) [2,2.83)    [2.83,3.35)
## [131] [2,2.83)    [3.35,4.4]  [2,2.83)    [2,2.83)    [2,2.83)
## [136] [2.83,3.35) [3.35,4.4]  [2.83,3.35) [2.83,3.35) [2.83,3.35)
## [141] [2.83,3.35) [2.83,3.35) [2,2.83)    [2.83,3.35) [2.83,3.35)
## [146] [2.83,3.35) [2,2.83)    [2.83,3.35) [3.35,4.4]  [2.83,3.35)
## attr(,"discretized:breaks")
## [1] 2.000000 2.826644 3.353010 4.400000
## attr(,"discretized:method")
## [1] cluster
## Levels: [2,2.83) [2.83,3.35) [3.35,4.4]
```

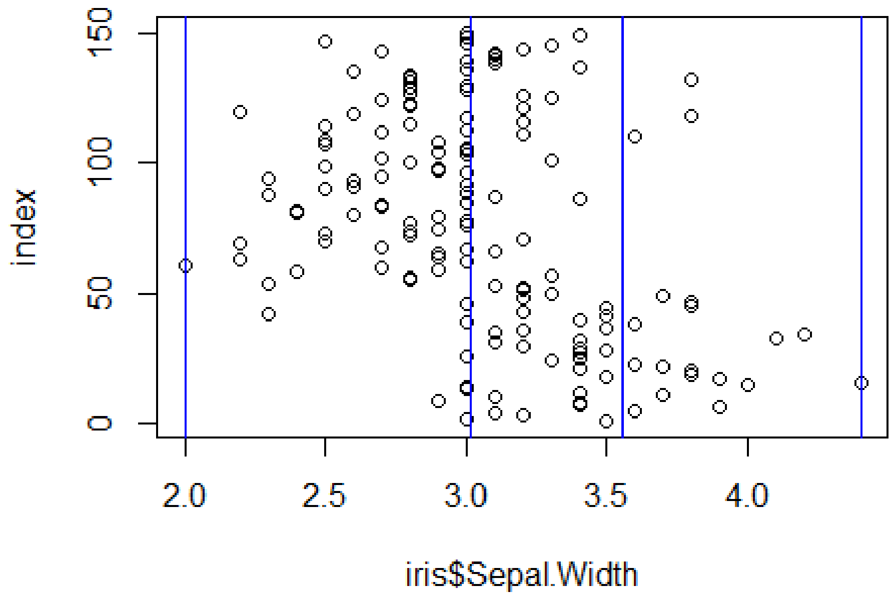
```
sp3 <- discretize(iris$Sepal.Width, method="cluster", categories=3,
onlycuts=TRUE) #get split points
```

```
## Warning in discretize(iris$Sepal.Width, method = "cluster", categories
## = 3, : Parameter categories is deprecated. Use breaks instead! Also, the
## default method is now frequency!
```

```
sp3
```

```
## [1] 2.000000 3.015048 3.554331 4.400000
```

```
plot(iris$Sepal.Width, 1:150, ylab="index")
abline(v=sp3, col="blue") #add straight lines to the current plot
```



```
#Convert continuous attributes to ordinal attributes with cut function
#Cut each attribute into ordered factors with three levels
iris_ord <- data.frame(  # create the new data frame
  cut(iris[,1], 3, labels=c("short", "medium", "long"), ordered=T),
  cut(iris[,2], 3, labels=c("short", "medium", "long"), ordered=T),
  cut(iris[,3], 3, labels=c("short", "medium", "long"), ordered=T),
  cut(iris[,4], 3, labels=c("short", "medium", "long"), ordered=T),
  iris[,5])
colnames(iris_ord) <- colnames(iris) #assign column names
head(iris_ord)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1        short      medium        short       short  setosa
## 2        short      medium        short       short  setosa
## 3        short      medium        short       short  setosa
## 4        short      medium        short       short  setosa
## 5        short      medium        short       short  setosa
## 6        short        long        short       short  setosa

######################## Normalize ########################
# Normalize each column (subtract mean and divide by the standard deviation)
using scale function
iris_scaled <- scale(iris[1:4])
head(iris_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
```

```
###################### Dissimilarity ######################
# R actually only uses dissimilarities
# Calculate distances between the first 5 objects (use only attributes 1-4)
# Note: Don't forget to normalize the data if the ranges are very different!

# Euclidean distance (L2 norm)
d1 <- dist(iris_scaled[1:5, 1:4], method="euclidean") #shows only lower
triangle of the distance matrix
d1
```

```
##           1         2         3         4
## 2 1.1722914
## 3 0.8427840 0.5216255
## 4 1.0999999 0.4325508 0.2829432
## 5 0.2592702 1.3818560 0.9882608 1.2459861
```

```
d1_mtx <- as.matrix(d1) #create full matrix of d1
d1_mtx
```

```
##           1         2         3         4         5
## 1 0.0000000 1.1722914 0.8427840 1.0999999 0.2592702
## 2 1.1722914 0.0000000 0.5216255 0.4325508 1.3818560
## 3 0.8427840 0.5216255 0.0000000 0.2829432 0.9882608
## 4 1.0999999 0.4325508 0.2829432 0.0000000 1.2459861
## 5 0.2592702 1.3818560 0.9882608 1.2459861 0.0000000
```

```
# Manhattan distance (L1 norm)
d2 <- dist(iris_scaled[1:5, 1:4], method="manhattan")
d2
```

```
##           1         2         3         4
## 2 1.3886674
## 3 1.2279853 0.7570306
## 4 1.5781768 0.6483657 0.4634868
## 5 0.3501915 1.4973323 1.3366502 1.6868417
```

```
d2_mtx <- as.matrix(d2) #create full matrix of d2
d2_mtx
```

```
##           1         2         3         4         5
## 1 0.0000000 1.3886674 1.2279853 1.5781768 0.3501915
## 2 1.3886674 0.0000000 0.7570306 0.6483657 1.4973323
## 3 1.2279853 0.7570306 0.0000000 0.4634868 1.3366502
## 4 1.5781768 0.6483657 0.4634868 0.0000000 1.6868417
## 5 0.3501915 1.4973323 1.3366502 1.6868417 0.0000000
```

```
# Supremum (L(inf) norm)
d3 <- dist(iris_scaled[1:5, 1:4], method="maximum")
d3

##           1         2         3         4
## 2 1.1471408
## 3 0.6882845 0.4588563
## 4 0.9177126 0.3622899 0.2294282
## 5 0.2294282 1.3765690 0.9177126 1.1471408

d3_mtx <- as.matrix(d3) #create full matrix of d3
d3_mtx

##           1         2         3         4         5
## 1 0.0000000 1.1471408 0.6882845 0.9177126 0.2294282
## 2 1.1471408 0.0000000 0.4588563 0.3622899 1.3765690
## 3 0.6882845 0.4588563 0.0000000 0.2294282 0.9177126
## 4 0.9177126 0.3622899 0.2294282 0.0000000 1.1471408
## 5 0.2294282 1.3765690 0.9177126 1.1471408 0.0000000

# Create binary matrix (market basket data)
b <- rbind(c(0,0,0,1,1,1,1,0,0), c(0,0,1,1,1,0,0,1,0))
b

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    0    0    0    1    1    1    1    0    0
## [2,]    0    0    1    1    1    0    0    1    0

# Jaccard
d4 <- dist(b, method="binary")
d4

##           1
## 2 0.6666667

# package proxy used to calculate
library(proxy)

##
## Attaching package: 'proxy'

## The following object is masked from 'package:Matrix':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix
```

```r
# You can change method, here is the list of methods available in proxy
package
names(pr_DB$get_entries())
```

```
##  [1] "Jaccard"         "Kulczynski1"     "Kulczynski2"
##  [4] "Mountford"       "Fager"           "Russel"
##  [7] "simple matching" "Hamman"          "Faith"
## [10] "Tanimoto"        "Dice"            "Phi"
## [13] "Stiles"          "Michael"         "Mozley"
## [16] "Yule"            "Yule2"           "Ochiai"
## [19] "Simpson"         "Braun-Blanquet"  "cosine"
## [22] "eJaccard"        "eDice"           "correlation"
## [25] "Chi-squared"     "Phi-squared"     "Tschuprow"
## [28] "Cramer"          "Pearson"         "Gower"
## [31] "Euclidean"       "Mahalanobis"     "Bhjattacharyya"
## [34] "Manhattan"       "supremum"        "Minkowski"
## [37] "Canberra"        "Wave"            "divergence"
## [40] "Kullback"        "Bray"            "Soergel"
## [43] "Levenshtein"     "Podani"          "Chord"
## [46] "Geodesic"        "Whittaker"       "Hellinger"
## [49] "fJaccard"
```

```r
# Simple Matching Coefficient
d5 <- dist(b, method = "simple matching")
d5
```

```
##           1
## 2 0.4444444
```

```r
# Cosine
d6 <- dist(b, method = "cosine")
d6
```

```
##     1
## 2 0.5
```

```r
# Create mixed data
data <- data.frame(
  height=c(160, 185, 170),
  weight=c(52, 90, 75),
  sex=c("female", "male", "male")
)
data
```

```
##   height weight    sex
## 1    160     52 female
## 2    185     90   male
## 3    170     75   male
```

```r
# Gower method is used for mixed type data set
d7 <- dist(data, method = "Gower")
d7
```

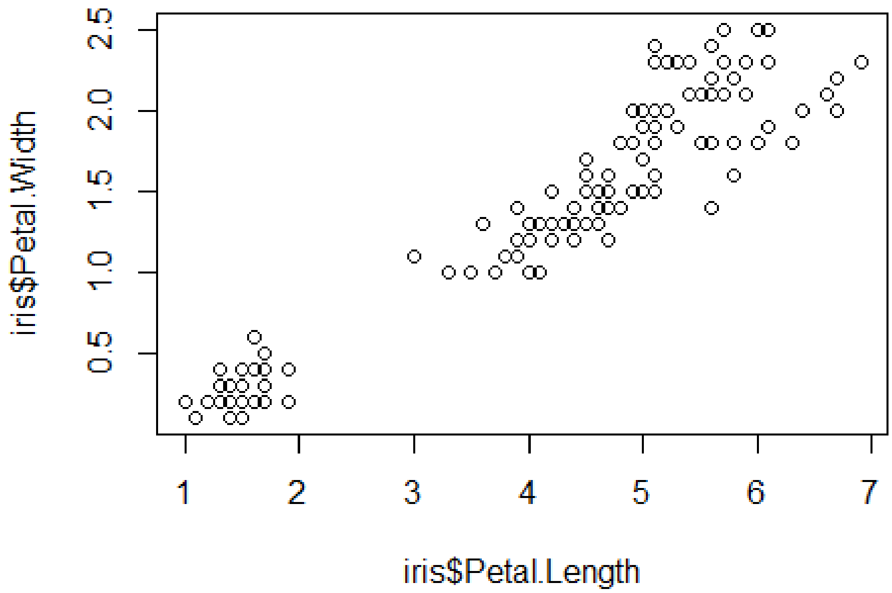
```
##           1         2
## 2 1.0000000          
## 3 0.6684211 0.3315789
```

```
##################### Correlation ######################
# Pearson for ratio/interval scaled features

# Correlation between the first 4 attributes
cr <-cor(iris[,1:4])
cr
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

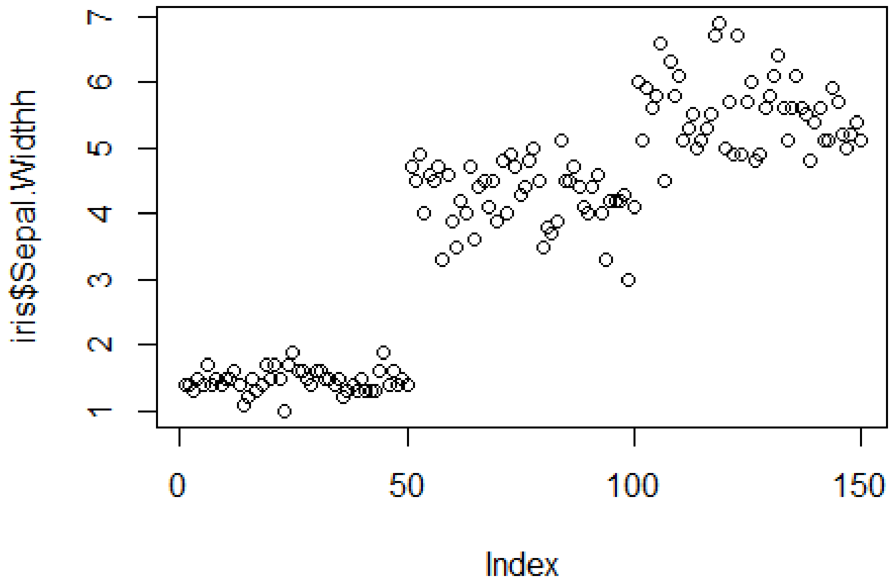
```
plot(iris$Petal.Length, iris$Petal.Width) #positive correlation of 0.9628
```



```
cor(iris$Petal.Length, iris$Petal.Width)
```

```
## [1] 0.9628654
```

```
plot(iris$Petal.Length, iris$Sepal.Widthh) #negative correlation of -0.428
```



```
cor(iris$Petal.Length, iris$Sepal.Width)

## [1] -0.4284401

# Correlation between data objects
# Have to transpose matrix first in order to make objects in columns
irist <- t(iris[,1:4])
irist

##              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
## Sepal.Length  5.1  4.9  4.7  4.6  5.0  5.4  4.6  5.0  4.4   4.9   5.4
## Sepal.Width   3.5  3.0  3.2  3.1  3.6  3.9  3.4  3.4  2.9   3.1   3.7
## Petal.Length  1.4  1.4  1.3  1.5  1.4  1.7  1.4  1.5  1.4   1.5   1.5
## Petal.Width   0.2  0.2  0.2  0.2  0.2  0.4  0.3  0.2  0.2   0.1   0.2
##              [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
## Sepal.Length   4.8   4.8   4.3   5.8   5.7   5.4   5.1   5.7   5.1   5.4
## Sepal.Width    3.4   3.0   3.0   4.0   4.4   3.9   3.5   3.8   3.8   3.4
## Petal.Length   1.6   1.4   1.1   1.2   1.5   1.3   1.4   1.7   1.5   1.7
## Petal.Width    0.2   0.1   0.1   0.2   0.4   0.4   0.3   0.3   0.3   0.2
##              [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31]
## Sepal.Length   5.1   4.6   5.1   4.8   5.0   5.0   5.2   5.2   4.7   4.8
## Sepal.Width    3.7   3.6   3.3   3.4   3.0   3.4   3.5   3.4   3.2   3.1
## Petal.Length   1.5   1.0   1.7   1.9   1.6   1.6   1.5   1.4   1.6   1.6
## Petal.Width    0.4   0.2   0.5   0.2   0.2   0.4   0.2   0.2   0.2   0.2
##              [,32] [,33] [,34] [,35] [,36] [,37] [,38] [,39] [,40] [,41]
## Sepal.Length   5.4   5.2   5.5   4.9   5.0   5.5   4.9   4.4   5.1   5.0
## Sepal.Width    3.4   4.1   4.2   3.1   3.2   3.5   3.6   3.0   3.4   3.5
## Petal.Length   1.5   1.5   1.4   1.5   1.2   1.3   1.4   1.3   1.5   1.3
## Petal.Width    0.4   0.1   0.2   0.2   0.2   0.2   0.1   0.2   0.2   0.3
##              [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50] [,51]
## Sepal.Length   4.5   4.4   5.0   5.1   4.8   5.1   4.6   5.3   5.0   7.0
## Sepal.Width    2.3   3.2   3.5   3.8   3.0   3.8   3.2   3.7   3.3   3.2
## Petal.Length   1.3   1.3   1.6   1.9   1.4   1.6   1.4   1.5   1.4   4.7
## Petal.Width    0.3   0.2   0.6   0.4   0.3   0.2   0.2   0.2   0.2   1.4
##              [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59] [,60] [,61]
## Sepal.Length   6.4   6.9   5.5   6.5   5.7   6.3   4.9   6.6   5.2   5.0
## Sepal.Width    3.2   3.1   2.3   2.8   2.8   3.3   2.4   2.9   2.7   2.0
## Petal.Length   4.5   4.9   4.0   4.6   4.5   4.7   3.3   4.6   3.9   3.5
## Petal.Width    1.5   1.5   1.3   1.5   1.3   1.6   1.0   1.3   1.4   1.0
##              [,62] [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71]
## Sepal.Length   5.9   6.0   6.1   5.6   6.7   5.6   5.8   6.2   5.6   5.9
## Sepal.Width    3.0   2.2   2.9   2.9   3.1   3.0   2.7   2.2   2.5   3.2
## Petal.Length   4.2   4.0   4.7   3.6   4.4   4.5   4.1   4.5   3.9   4.8
## Petal.Width    1.5   1.0   1.4   1.3   1.4   1.5   1.0   1.5   1.1   1.8
##              [,72] [,73] [,74] [,75] [,76] [,77] [,78] [,79] [,80] [,81]
## Sepal.Length   6.1   6.3   6.1   6.4   6.6   6.8   6.7   6.0   5.7   5.5
## Sepal.Width    2.8   2.5   2.8   2.9   3.0   2.8   3.0   2.9   2.6   2.4
## Petal.Length   4.0   4.9   4.7   4.3   4.4   4.8   5.0   4.5   3.5   3.8
## Petal.Width    1.3   1.5   1.2   1.3   1.4   1.4   1.7   1.5   1.0   1.1
##              [,82] [,83] [,84] [,85] [,86] [,87] [,88] [,89] [,90] [,91]
## Sepal.Length   5.5   5.8   6.0   5.4   6.0   6.7   6.3   5.6   5.5   5.5
## Sepal.Width    2.4   2.7   2.7   3.0   3.4   3.1   2.3   3.0   2.5   2.6
## Petal.Length   3.7   3.9   5.1   4.5   4.5   4.7   4.4   4.1   4.0   4.4
## Petal.Width    1.0   1.2   1.6   1.5   1.6   1.5   1.3   1.3   1.3   1.2
##              [,92] [,93] [,94] [,95] [,96] [,97] [,98] [,99] [,100] [,101]
## Sepal.Length   6.1   5.8   5.0   5.6   5.7   5.7   6.2   5.1    5.7    6.3
## Sepal.Width    3.0   2.6   2.3   2.7   3.0   2.9   2.9   2.5    2.8    3.3
## Petal.Length   4.6   4.0   3.3   4.2   4.2   4.2   4.3   3.0    4.1    6.0
## Petal.Width    1.4   1.2   1.0   1.3   1.2   1.3   1.3   1.1    1.3    2.5
##              [,102] [,103] [,104] [,105] [,106] [,107] [,108] [,109]
## Sepal.Length    5.8    7.1    6.3    6.5    7.6    4.9    7.3    6.7
## Sepal.Width     2.7    3.0    2.9    3.0    3.0    2.5    2.9    2.5
## Petal.Length    5.1    5.9    5.6    5.8    6.6    4.5    6.3    5.8
## Petal.Width     1.9    2.1    1.8    2.2    2.1    1.7    1.8    1.8
##              [,110] [,111] [,112] [,113] [,114] [,115] [,116] [,117]
## Sepal.Length    7.2    6.5    6.4    6.8    5.7    5.8    6.4    6.5
## Sepal.Width     3.6    3.2    2.7    3.0    2.5    2.8    3.2    3.0
## Petal.Length    6.1    5.1    5.3    5.5    5.0    5.1    5.3    5.5
## Petal.Width     2.5    2.0    1.9    2.1    2.0    2.4    2.3    1.8
##              [,118] [,119] [,120] [,121] [,122] [,123] [,124] [,125]
## Sepal.Length    7.7    7.7    6.0    6.9    5.6    7.7    6.3    6.7
## Sepal.Width     3.8    2.6    2.2    3.2    2.8    2.8    2.7    3.3
## Petal.Length    6.7    6.9    5.0    5.7    4.9    6.7    4.9    5.7
## Petal.Width     2.2    2.3    1.5    2.3    2.0    2.0    1.8    2.1
##              [,126] [,127] [,128] [,129] [,130] [,131] [,132] [,133]
## Sepal.Length    7.2    6.2    6.1    6.4    7.2    7.4    7.9    6.4
## Sepal.Width     3.2    2.8    3.0    2.8    3.0    2.8    3.8    2.8
## Petal.Length    6.0    4.8    4.9    5.6    5.8    6.1    6.4    5.6
## Petal.Width     1.8    1.8    1.8    2.1    1.6    1.9    2.0    2.2
##              [,134] [,135] [,136] [,137] [,138] [,139] [,140] [,141]
## Sepal.Length    6.3    6.1    7.7    6.3    6.4    6.0    6.9    6.7
## Sepal.Width     2.8    2.6    3.0    3.4    3.1    3.0    3.1    3.1
## Petal.Length    5.1    5.6    6.1    5.6    5.5    4.8    5.4    5.6
## Petal.Width     1.5    1.4    2.3    2.4    1.8    1.8    2.1    2.4
##              [,142] [,143] [,144] [,145] [,146] [,147] [,148] [,149]
## Sepal.Length    6.9    5.8    6.8    6.7    6.7    6.3    6.5    6.2
## Sepal.Width     3.1    2.7    3.2    3.3    3.0    2.5    3.0    3.4
## Petal.Length    5.1    5.1    5.9    5.7    5.2    5.0    5.2    5.4
## Petal.Width     2.3    1.9    2.3    2.5    2.3    1.9    2.0    2.3
##              [,150]
## Sepal.Length    5.9
## Sepal.Width     3.0
## Petal.Length    5.1
## Petal.Width     1.8
```

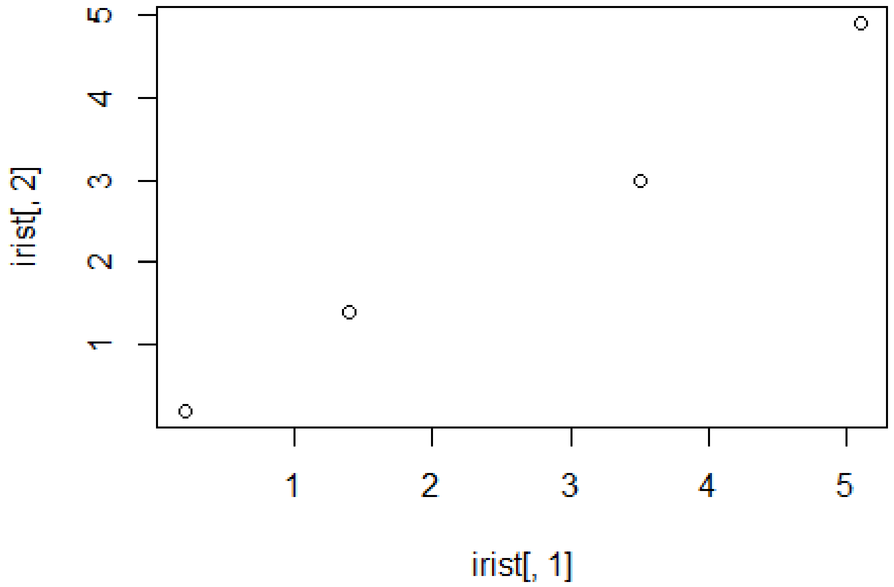
```
cc <- cor(irist)
dim(cc) # matrix of size n x n
```

```
## [1] 150 150
```

```
cc[1:5,1:5] #takes only the correlations between object 1 to 5
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 1.0000000 0.9959987 0.9999739 0.9981685 0.9993473
## [2,] 0.9959987 1.0000000 0.9966071 0.9973966 0.9922327
## [3,] 0.9999739 0.9966071 1.0000000 0.9983335 0.9990611
## [4,] 0.9981685 0.9973966 0.9983335 1.0000000 0.9967188
## [5,] 0.9993473 0.9922327 0.9990611 0.9967188 1.0000000
```

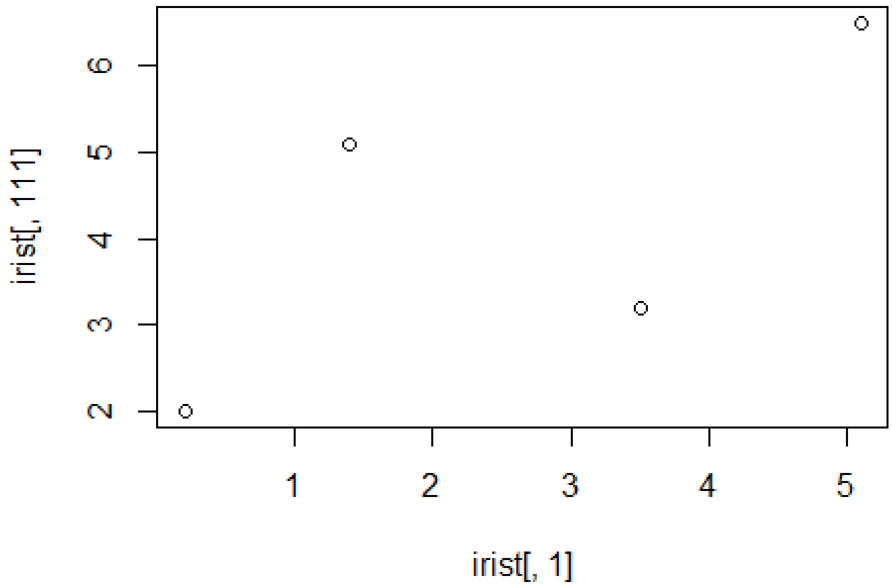
```
#correlation plot between object 1 and 2 (same type = highly correlated)
plot(irist[,1],irist[,2])
```



```
cc[1,2]

## [1] 0.9959987

#correlation plot between object 1 and 111 (different type = less correlated)
plot(irist[,1],irist[,111])
```



```
cc[1,111]

## [1] 0.6938075

# make correlation into a distance (dissimilarity) for the fist 5 objects
d <- as.dist(1-abs(cc[1:5,1:5]))
d

##              1            2            3            4
## 2 4.001339e-03
## 3 2.608895e-05 3.392914e-03
## 4 1.831548e-03 2.603370e-03 1.666521e-03
## 5 6.526685e-04 7.767321e-03 9.388680e-04 3.281179e-03

###############################################################
# From iris_ord data frame created above
head(iris_ord)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1        short      medium        short       short  setosa
## 2        short      medium        short       short  setosa
## 3        short      medium        short       short  setosa
## 4        short      medium        short       short  setosa
## 5        short      medium        short       short  setosa
## 6        short        long        short       short  setosa

# Is sepal length and species related?
# Create table that has sepal length levels in rows and iris species in
```

```
column
tbl <- table(Sepal.Length=iris_ord$Sepal.Length, iris_ord$Species)
tbl
```

```
##
## Sepal.Length setosa versicolor virginica
##       short      47         11         1
##       medium      3         36        32
##       long        0          3        17
```