Multivariate Statistics - Exercise 2

Philipp Satlawa - h0640348

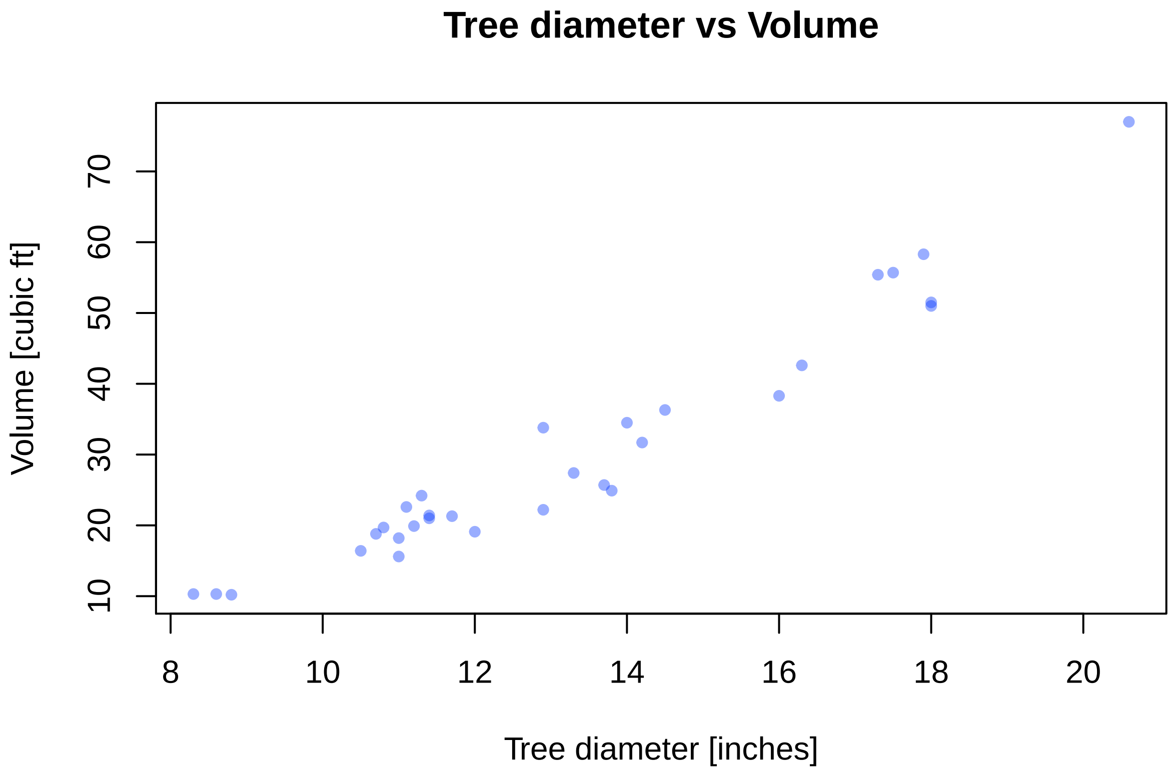
06/05/2021

This document contains the answered questions of exercise 2 of the course "Multivariate Statistics".

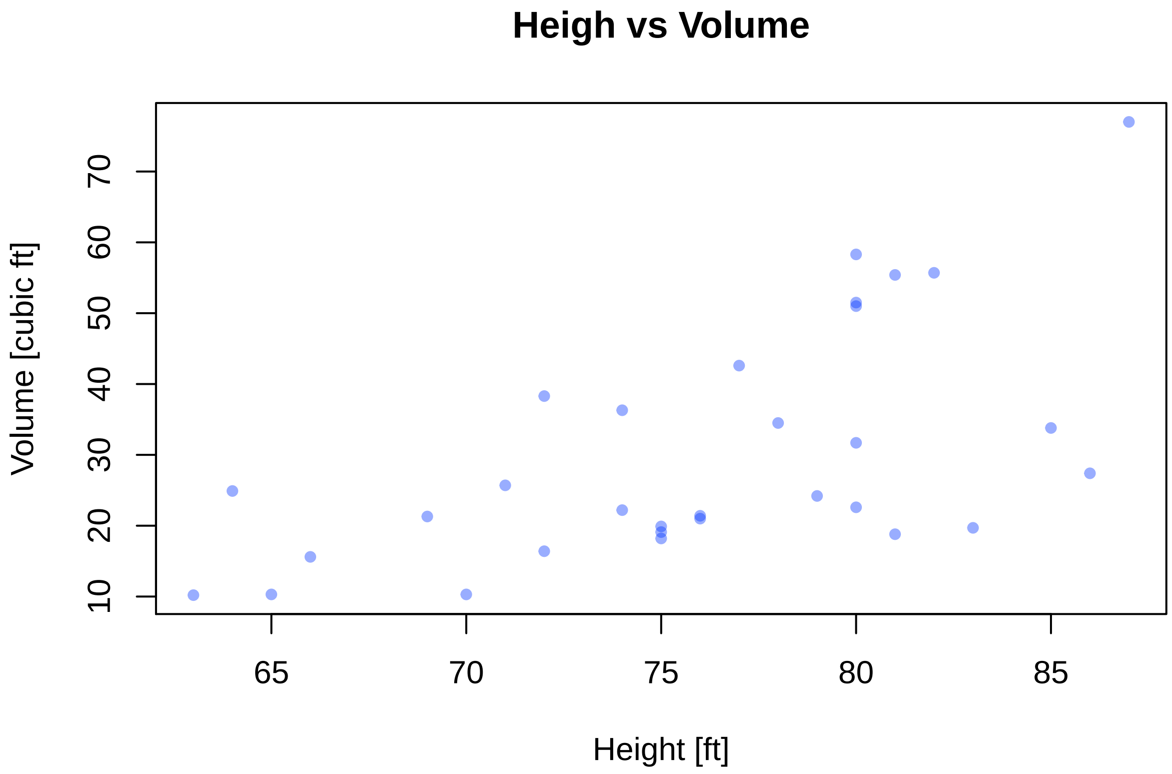
Linear regression

1.install and load the packages - load and summerize the data

```
# load dataset "trees"
data(trees)

# examine help page
help(trees)

# scatterplot "Girth" vs "Volume"
plot(trees[c("Girth","Volume")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "Tree diameter vs Volume",
      xlab = "Tree diameter [inches]", ylab = "Volume [cubic ft]")
```



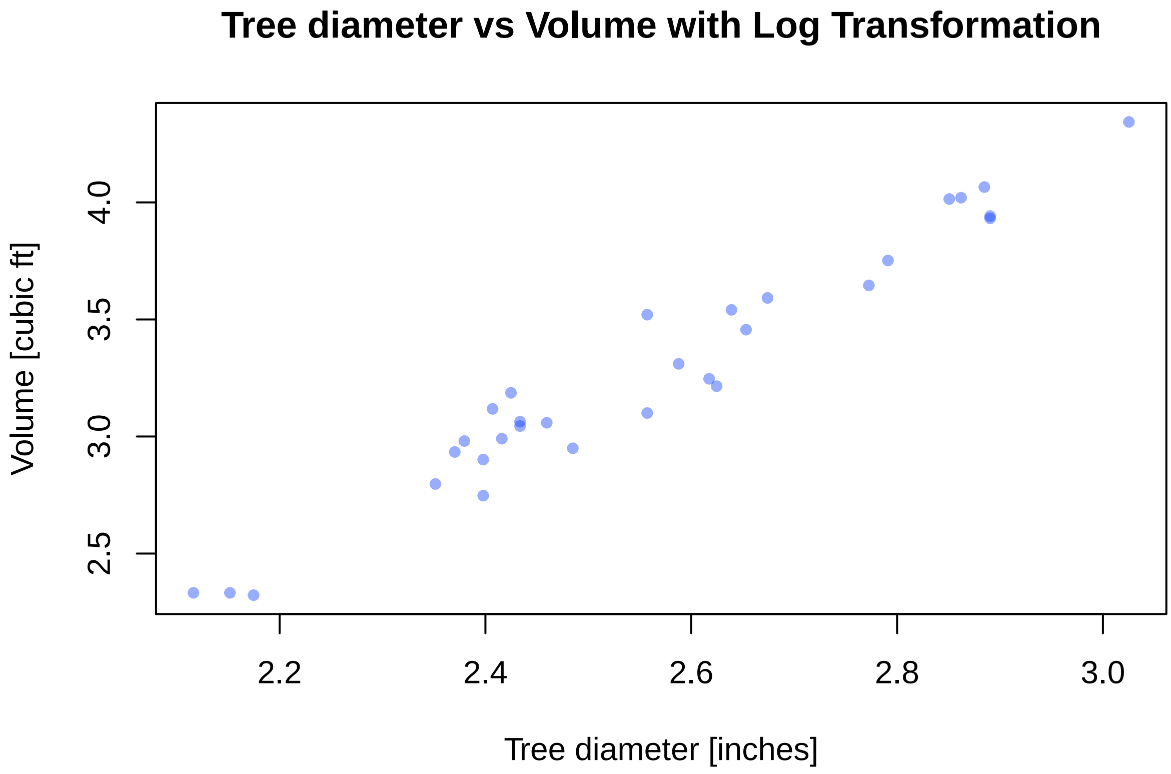
```r
# scatterplot "Height" vs "Volume"
plot(trees[c("Height","Volume")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "Heigh vs Volume",
      xlab = "Height [ft]", ylab = "Volume [cubic ft]")
```



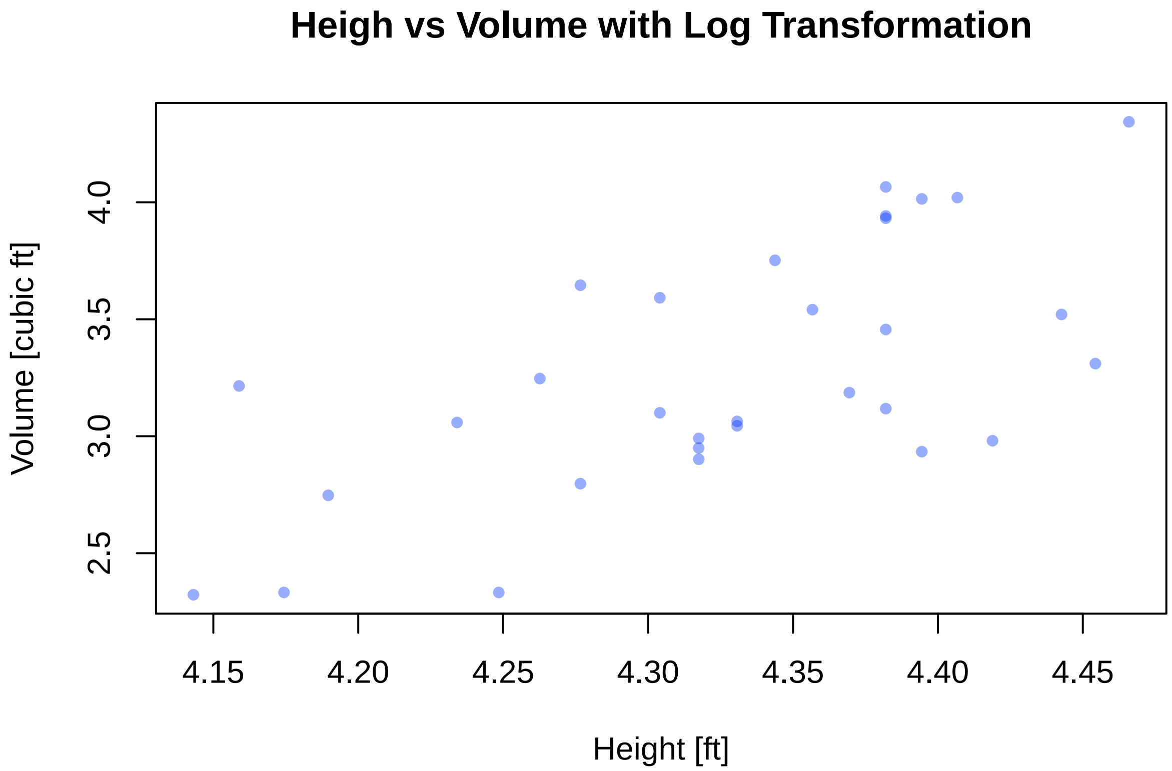
In the fist scatterplot there seems to be a positive correlation between the variables `Volume` and `Girth` (Tree diameter). This correlation is almost linear. However, the second scatterplot showing the variables `Volume` and `Height` indicates a very vague positive relationship for the two variables. The correlation between the variables `Volume` and `Girth` have a constant variance (variance of `Volume` is nearly constant independent of the `Girth`) whereas the relationship between `Volume` and `Height` seems to have an increasing variance (variance of `Volume` is increasing with the increase in `Height`). That makes sense, since young trees have to grow in height to increase their volume, however depending on how much concurrence they experienced from other trees they might have a small or big tree crown (tree crown diameter and tree diameter have a high positive correlation).

2. log transformation of the data

```r
# logarithmic transform
trees_log <- log(trees)

# scatterplot "Girth" vs "Volume with Log Transformation"
plot(trees_log[c("Girth","Volume")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "Tree diameter vs Volume with Log Transformation",
      xlab = "Tree diameter [inches]", ylab = "Volume [cubic ft]")
```



```
# scatterplot "Height" vs "Volume with Log Transformation"
plot(trees_log[c("Height","Volume")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "Heigh vs Volume with Log Transformation",
      xlab = "Height [ft]", ylab = "Volume [cubic ft]")
```



After the log transformation the relationship between the attributes `Girth` and `Volume` seems to be clearly linear, as shown in scatterplot "Tree diameter vs Volume with Log Transformation". The second scatterplot shows the positive but noisy correlation between the variables `Volume` and `Height`.

3. bivariate regression

```r
# create linear model using "Volume"|"Girth"
model_girth <- lm(Volume ~ Girth, data = trees)
summary(model_girth)
```

```
##
## Call:
## lm(formula = Volume ~ Girth, data = trees)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.065 -3.107  0.152  3.495  9.587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
## Girth         5.0659     0.2474   20.48  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.252 on 29 degrees of freedom
## Multiple R-squared:  0.9353, Adjusted R-squared:  0.9331
## F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

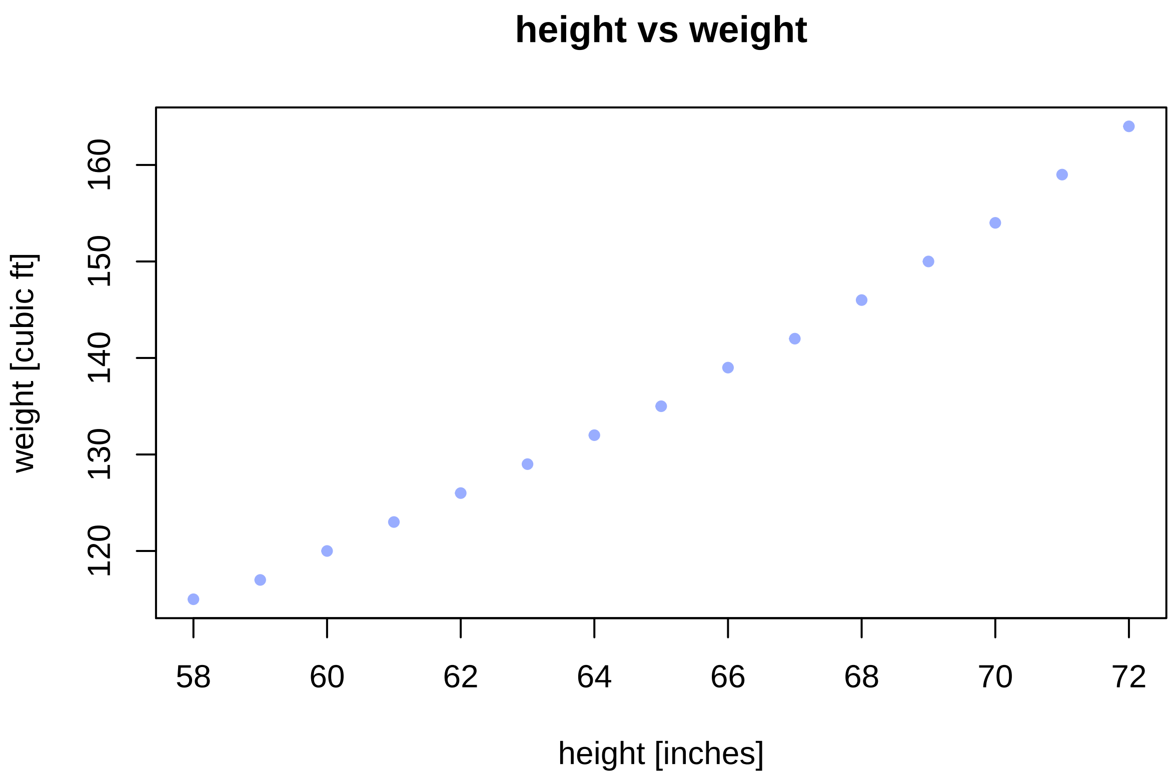
```r
# create linear model using "Volume"|"Height"
model_height <- lm(Volume ~ Height, data = trees)
summary(model_height)
```

```
##
## Call:
## lm(formula = Volume ~ Height, data = trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.274  -9.894  -2.894  12.068  29.852
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -87.1236    29.2731  -2.976 0.005835 **
## Height        1.5433     0.3839   4.021 0.000378 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 13.4 on 29 degrees of freedom
## Multiple R-squared:  0.3579, Adjusted R-squared:  0.3358
## F-statistic: 16.16 on 1 and 29 DF,  p-value: 0.0003784
```

Comparing the two bivariate linear regression models `model_girth` and `model_height` we can clearly identify the preeminence of `model_girth` with a R^2 of 0.935 over `model_height` with a R^2 of 0.358. Hence, Tree diameter is highly correlated with wood volume.

4. bivariate polynomial regression

```r
# load dataset
data(women)

# Visualize the relationship between "height" vs "weight"
plot(women[c("height","weight")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "height vs weight",
      xlab = "height [inches]", ylab = "weight [cubic ft]")
```



```
# compute polynomial regression and visualize the fits
plot(women[c("height","weight")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "polynomial regression",
      xlab = "height [inches]", ylab = "weight [cubic ft]")

# fit polynomial models up to order 5
model_weight_2 <- lm(weight ~ poly(height,2), data = women)
print(summary(model_weight_2))
```

```
##
## Call:
## lm(formula = weight ~ poly(height, 2), data = women)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50941 -0.29611 -0.00941  0.28615  0.59706
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      136.73333    0.09917 1378.85  < 2e-16 ***
## poly(height, 2)1  57.72954    0.38407  150.31  < 2e-16 ***
## poly(height, 2)2   5.33510    0.38407   13.89 9.32e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3841 on 12 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9994
## F-statistic: 1.139e+04 on 2 and 12 DF,  p-value: < 2.2e-16
```

```
lines(women$height, predict(model_weight_2), col=3)

model_weight_3 <- lm(weight ~ poly(height,3), data = women)
print(summary(model_weight_3))
```

```
##
## Call:
## lm(formula = weight ~ poly(height, 3), data = women)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40677 -0.17391  0.03091  0.12051  0.42191
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      136.7333     0.0667 2049.86  < 2e-16 ***
## poly(height, 3)1  57.7295     0.2583  223.46  < 2e-16 ***
## poly(height, 3)2   5.3351     0.2583   20.65 3.79e-10 ***
## poly(height, 3)3   1.0178     0.2583    3.94  0.00231 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.2583 on 11 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9997
## F-statistic: 1.679e+04 on 3 and 11 DF,  p-value: < 2.2e-16
```

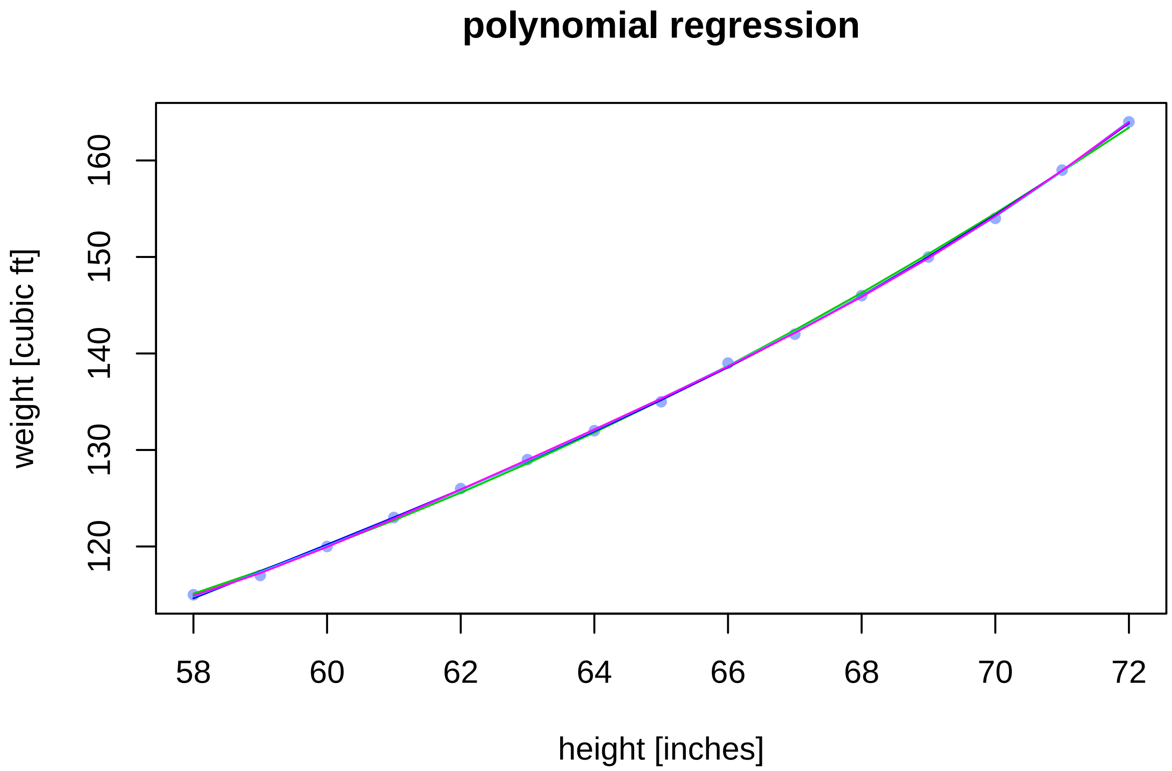
```
lines(women$height, predict(model_weight_3), col=4)

model_weight_4 <- lm(weight ~ poly(height,4), data = women)
print(summary(model_weight_4))
```

```
##
## Call:
## lm(formula = weight ~ poly(height, 4), data = women)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32577 -0.11881  0.07792  0.13334  0.30466
##
## Coefficients:
##                   Estimate Std. Error  t value Pr(>|t|)
## (Intercept)      136.73333    0.05794 2359.814  < 2e-16 ***
## poly(height, 4)1  57.72954    0.22441  257.250  < 2e-16 ***
## poly(height, 4)2   5.33510    0.22441   23.774 3.94e-10 ***
## poly(height, 4)3   1.01780    0.22441    4.535  0.00108 **
## poly(height, 4)4   0.48016    0.22441    2.140  0.05807 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.2244 on 10 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9998
## F-statistic: 1.669e+04 on 4 and 10 DF,  p-value: < 2.2e-16
```

```
lines(women$height, predict(model_weight_4), col=5)

model_weight_5 <- lm(weight ~ poly(height,5), data = women)
print(summary(model_weight_5))
```

```
##
## Call:
## lm(formula = weight ~ poly(height, 5), data = women)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32555 -0.13048  0.04343  0.10887  0.35110
##
## Coefficients:
##                   Estimate Std. Error  t value Pr(>|t|)
## (Intercept)      136.73333    0.05796 2359.101  < 2e-16 ***
## poly(height, 5)1  57.72954    0.22448  257.173  < 2e-16 ***
## poly(height, 5)2   5.33510    0.22448   23.767 1.97e-09 ***
## poly(height, 5)3   1.01780    0.22448    4.534  0.00142 **
## poly(height, 5)4   0.48016    0.22448    2.139  0.06112 .
## poly(height, 5)5  -0.22380    0.22448   -0.997  0.34482
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2245 on 9 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9998
## F-statistic: 1.335e+04 on 5 and 9 DF,  p-value: < 2.2e-16
```

```
lines(women$height, predict(model_weight_5), col=6)
```



```
# preform anova on fitted models
anova(model_weight_2, model_weight_3, model_weight_4, model_weight_5)
```

```
## Analysis of Variance Table
##
## Model 1: weight ~ poly(height, 2)
## Model 2: weight ~ poly(height, 3)
## Model 3: weight ~ poly(height, 4)
## Model 4: weight ~ poly(height, 5)
##   Res.Df     RSS Df Sum of Sq       F   Pr(>F)
## 1     12 1.77007
## 2     11 0.73415  1   1.03592 20.5580 0.001418 **
## 3     10 0.50360  1   0.23055  4.5753 0.061121 .
## 4      9 0.45351  1   0.05009  0.9940 0.344824
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looking at the fitted lines in the "polynomial regression" plot, the green (polynomial 2) and blue (polynomial 3) fit the data best. Considering that simpler models that have similar performance to more complex ones are preferred, the model `model_weight_2` is the best model to be selected. The results of anova show again that `model_weight_2` (polynomial 2) is the best fitting model.

Clustering

5. load and explore the milk dataset

```
# import necessary libraries
library(grid)
library(lattice)
library(modeltools)
```

```
## Loading required package: stats4
```

```
library(stats4)
library(flexclust)

library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:modeltools':
##
##     Predict
```

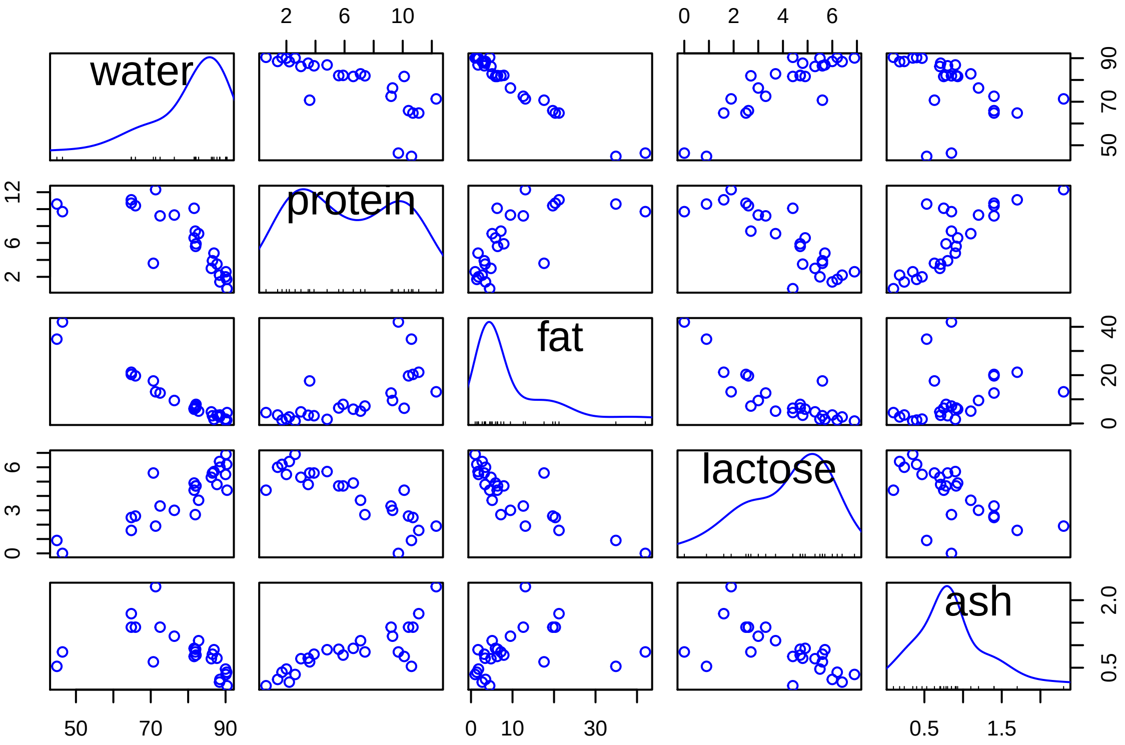
```
# load dataset milk
data(milk)

help(milk)
str(milk)
```

```
## 'data.frame':    25 obs. of  5 variables:
##  $ water  : num  90.1 88.5 88.4 90.3 90.4 87.7 86.9 82.1 81.9 81.6 ...
##  $ protein: num  2.6 1.4 2.2 1.7 0.6 3.5 4.8 5.9 7.4 10.1 ...
##  $ fat    : num  1 3.5 2.7 1.4 4.5 3.4 1.7 7.9 7.2 6.3 ...
##  $ lactose: num  6.9 6 6.4 6.2 4.4 4.8 5.7 4.7 2.7 4.4 ...
##  $ ash    : num  0.35 0.24 0.18 0.4 0.1 0.71 0.9 0.78 0.85 0.75 ...
```

```
head(milk)
```

```
##           water protein fat lactose  ash
## HORSE      90.1     2.6 1.0     6.9 0.35
## ORANGUTAN  88.5     1.4 3.5     6.0 0.24
## MONKEY     88.4     2.2 2.7     6.4 0.18
## DONKEY     90.3     1.7 1.4     6.2 0.40
## HIPPO      90.4     0.6 4.5     4.4 0.10
## CAMEL      87.7     3.5 3.4     4.8 0.71
```

```
scatterplotMatrix(milk, smooth = FALSE, regLine = FALSE)
```



The dataset `milk` contains 25 records and 5 attributes. The records represent tree species and all attributes are stored as continuous numbers. Examining the scatterplot matrix we can see the correlations shown in table 1.

Table 1: Correlations between variables.

variable 1	variable 2	correlation
water	protein	-
water	fat	-
water	lactose	+
water	ash	-
protein	fat	+
protein	lactose	-
protein	ash	+
fat	lactose	-
fat	ash	+
lactose	ash	-

6. data preperation

```r
# convert data from a data.frame to a matrix
milk_matrix <- as.matrix(milk)
# distance measure
milk_dist <- dist(milk_matrix, method = "euclidean")

# check classes of objects
print(class(milk))
```

```
## [1] "data.frame"
```

```r
print(class(milk_matrix))
```

```
## [1] "matrix"
```

```r
print(class(milk_dist))
```

```
## [1] "dist"
```

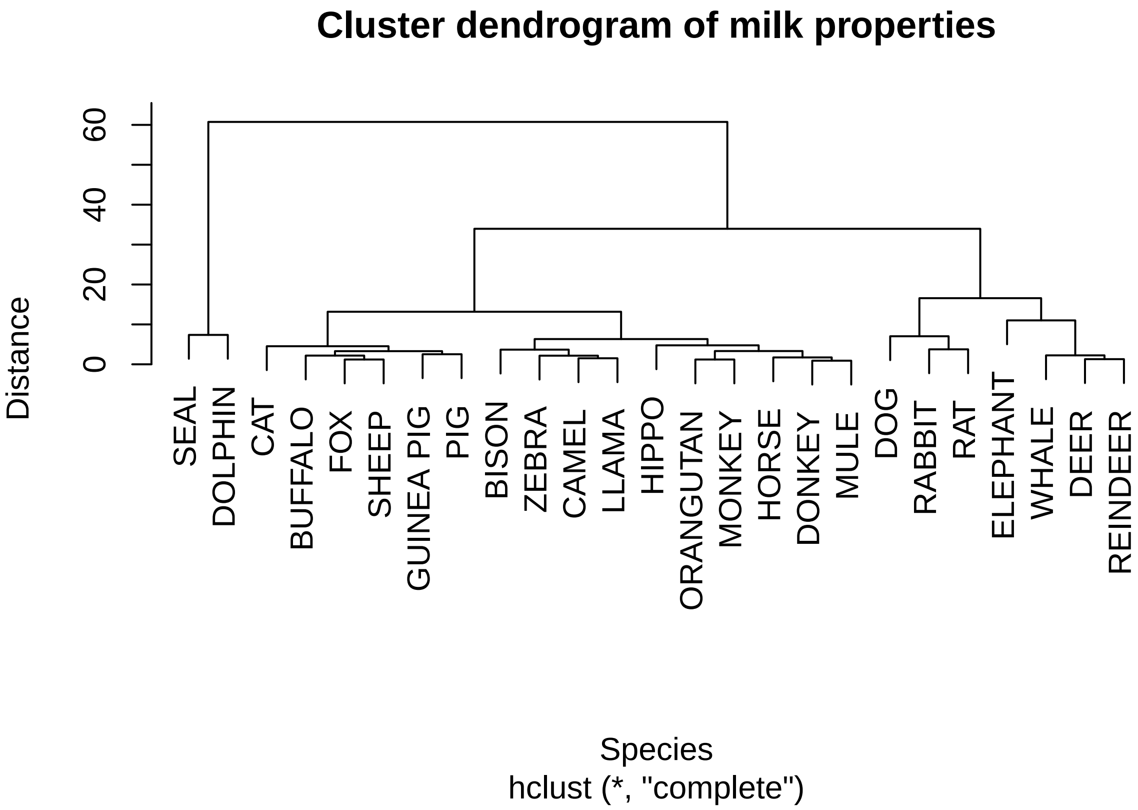
Table 2 shows the classes of the objects.

Table 2: Classes of the objects.

object	**class**
`milk`	data.frame
`milk_matrix`	matrix
`milk_dist`	dist

7. hierarchical clustering

```r
# hierarchical clustering
hc1 <- hclust(milk_dist)
# plot
plot(hc1, main = "Cluster dendrogram of milk properties",
      xlab = "Species", ylab = "Distance")
```



The default linkage method of the function `hclust` is "complete". The cluster hierarchical dendogram shows that the animals living in the sea (seal and dolphin) (not whale!) have very different milk properties (high distance) to the milk of land living animals. In the matrix scatterplot (*5. load and explore the milk dataset*) in the first column (`water`), they are most probably the outliers not far away from the rest of the data points. The land living animals (exept whale) can be further divided into at least three subgroups where there is a clear distance between the subgroups.

8. cut tree

```
# cut the tree to get 3 clusters
table(cutree(hc1, k = 3))
```

```
##
##  1  2  3
## 16  7  2
```

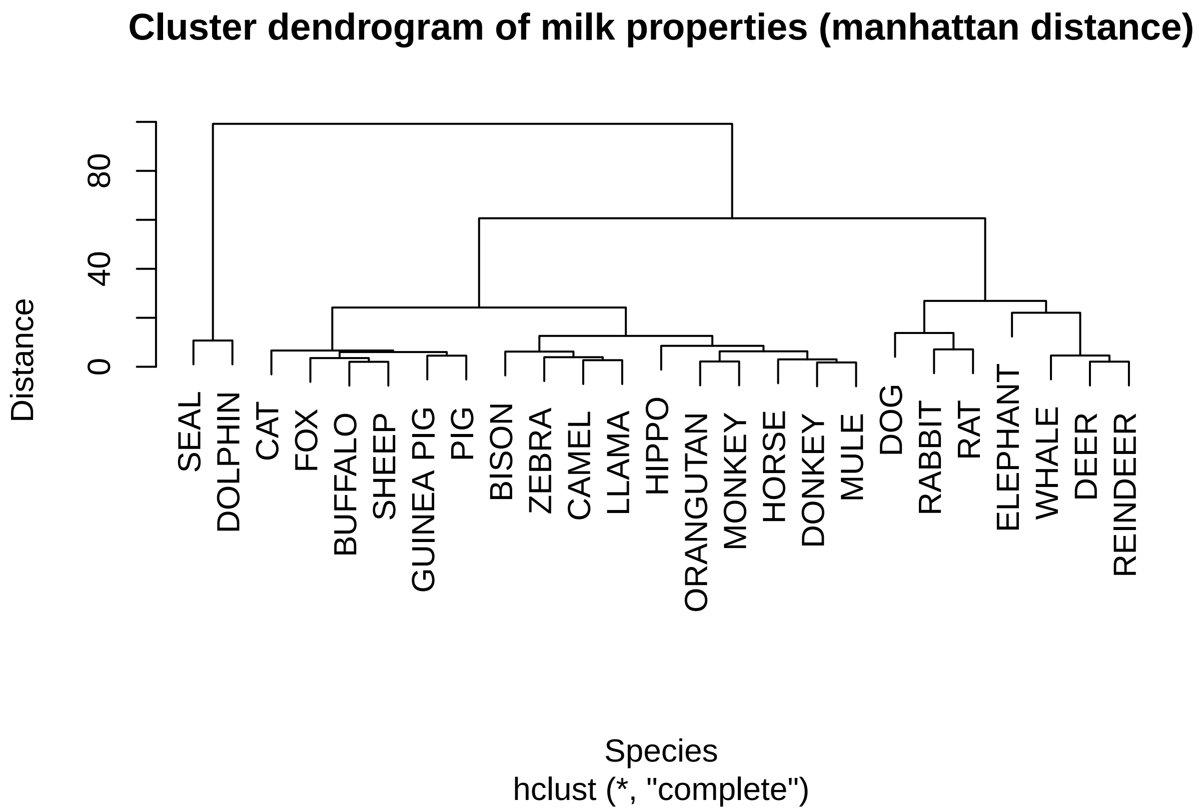
```
# cut the tree at height 10
table(cutree(hc1, h = 10))
```

```
##
##  1  2  3  4  5  6
## 10  6  3  1  3  2
```

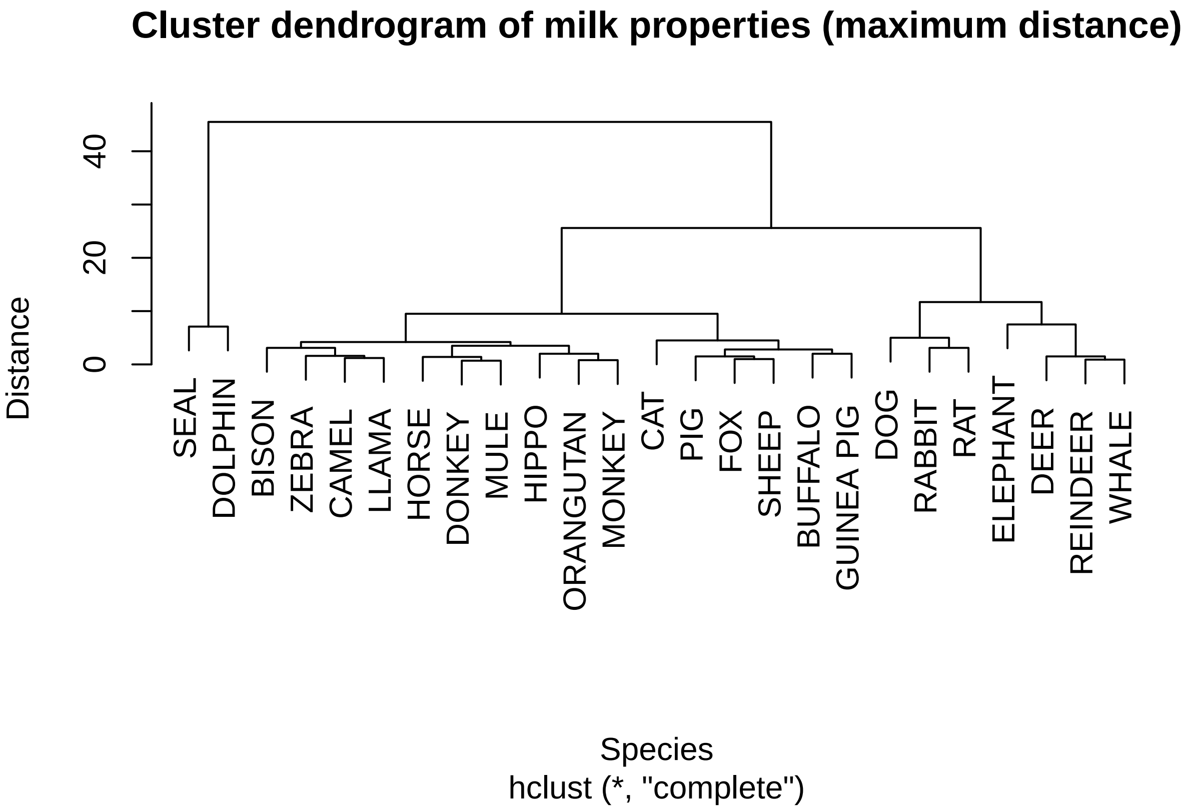
We get 3 clusters by using the function `cutree` with the argument "k = 3". Cutting the tree at height 10 results in 6 clusters.

9. use two more distance measures and two more linkage methods

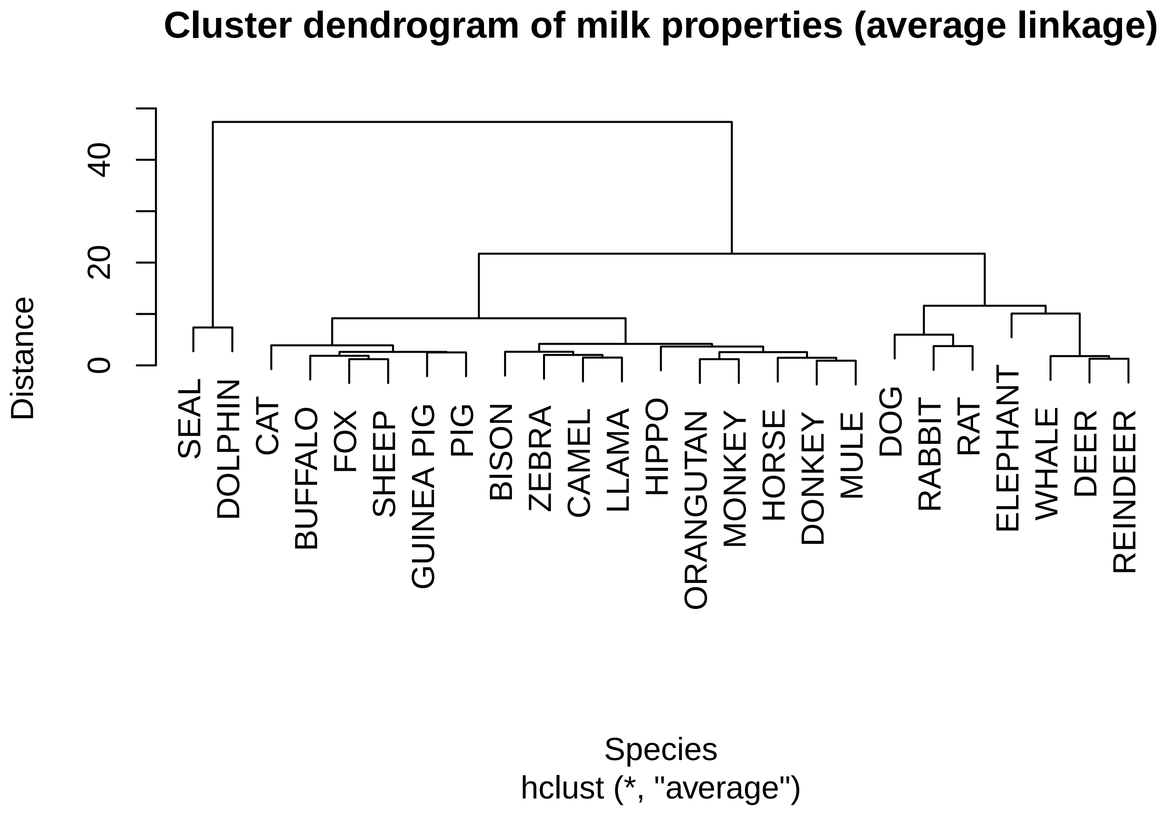
```
# distance measure "manhattan"
milk_dist <- dist(milk_matrix, method = "manhattan")
hc1 <- hclust(milk_dist)
plot(hc1, main = "Cluster dendrogram of milk properties (manhattan distance)",
      xlab = "Species", ylab = "Distance")
```



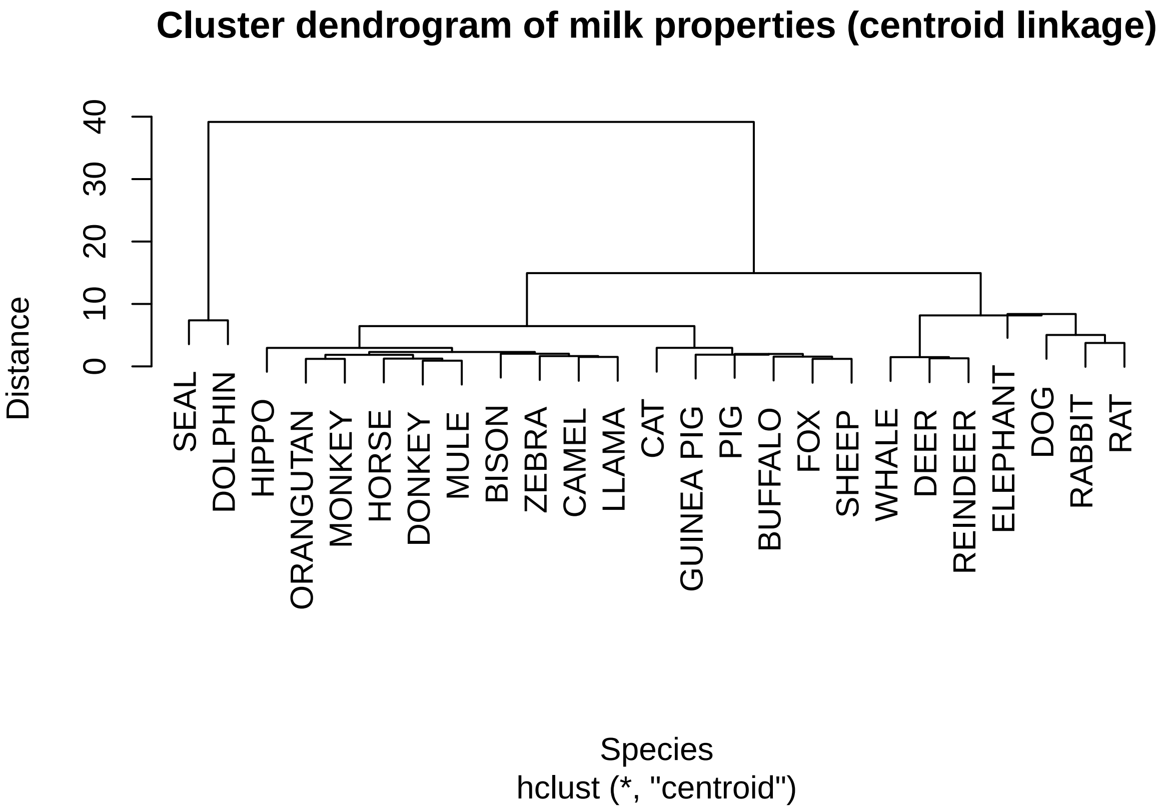
```
# distance measure "binary" "nominal"
milk_dist <- dist(milk_matrix, method = "maximum")
hc1 <- hclust(milk_dist)
plot(hc1, main = "Cluster dendrogram of milk properties (maximum distance)",
      xlab = "Species", ylab = "Distance")
```



```
# set distance measure back to "euclidean"
milk_dist <- dist(milk_matrix, method = "euclidean")
hc1 <- hclust(milk_dist)

# linkage method "average"
hc1 <- hclust(milk_dist, method = "average")
plot(hc1, main = "Cluster dendrogram of milk properties (average linkage)",
      xlab = "Species", ylab = "Distance")
```



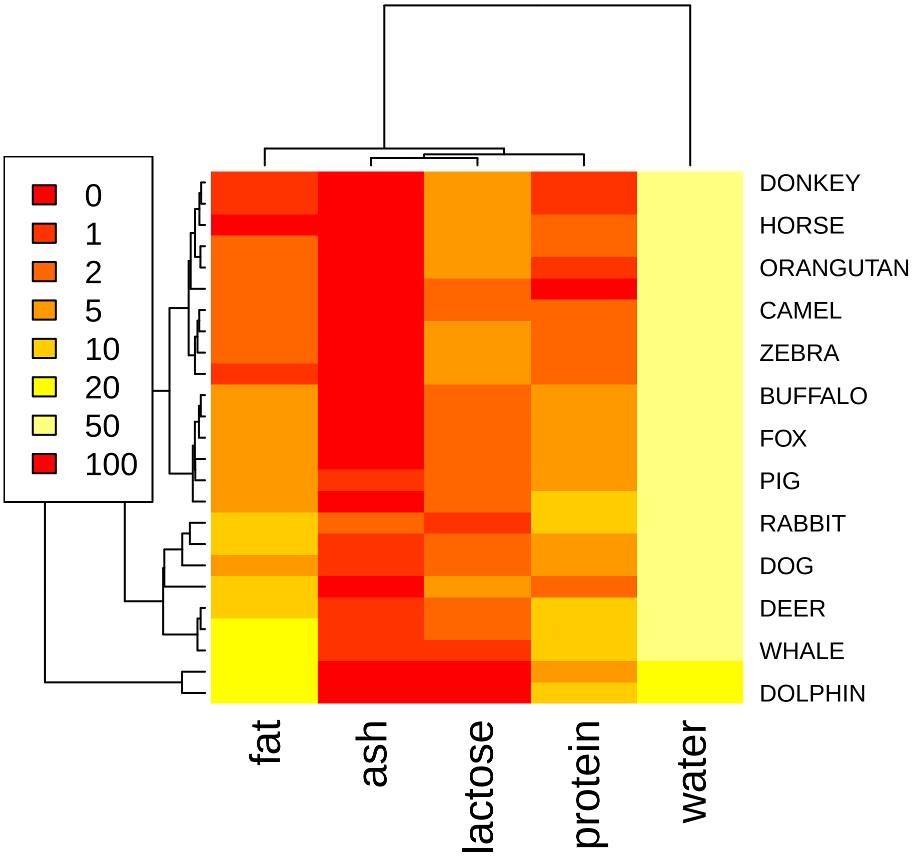
```r
# linkage method "centroid"
hc1 <- hclust(milk_dist, method = "centroid")
plot(hc1, main = "Cluster dendrogram of milk properties (centroid linkage)",
      xlab = "Species", ylab = "Distance")
```



After calculating the dendrograms for the distances "manhattan" and "maximum" and the linkages "average" and "centroid" the absolute distances change, however relative distances between the species change just slightly. Most importantly the 4 main clusters as described in *7. hierarchical clustering* are basically the same and contain the same species.

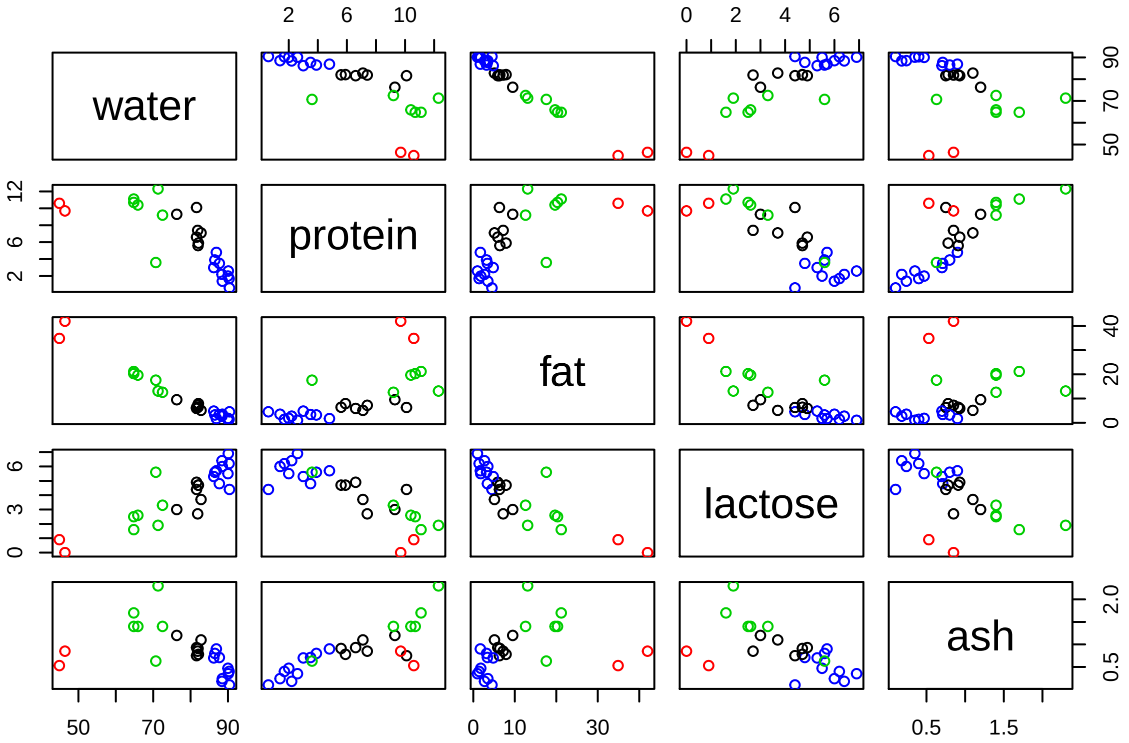
10. heatmap

```
# heatmap

breaks <-c(0, 1, 2, 5, 10, 20, 50, 100)
heatmap(milk_matrix,
hclustfun = function(x) hclust(x, "average"),
distfun = function(x) dist(x, "manhattan"),
scale = "none", breaks=breaks,
col = heat.colors(7))
legend(0,1, legend = c(breaks), fill = heat.colors(7),
bg = "white")
```



11. k-means

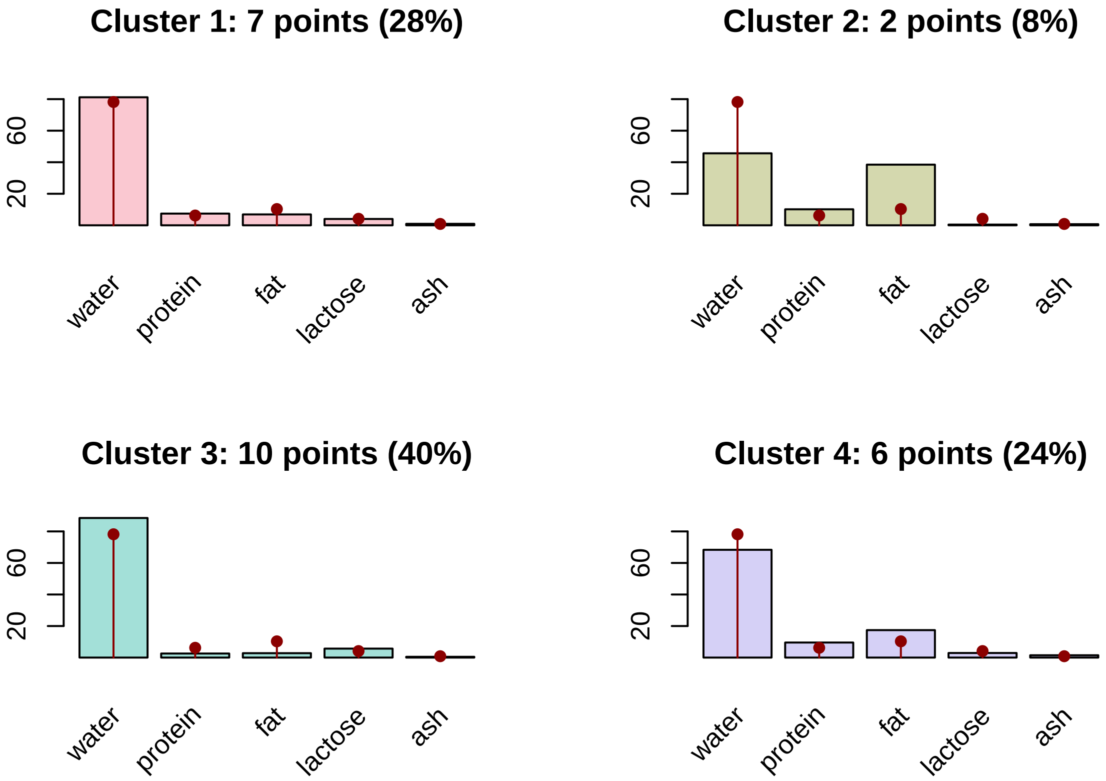
```
# calculate k-means
km1 <- kmeans(milk_matrix, centers=4)

# plot the result of the k-means algorithm
pairs(milk_matrix, col = km1$cluster)
```



After trying different numbers of clusters "k", I would suggest 4 numbers of clusters for this dataset. Four clusters seem to capture best the groups in the dataset. The result is partly biased by the previous result of the hierarchical clustering. However, 2 clusters seem to less (it is just capturing the dolphin and the seal). The number of clusters between 3-4 seem to work fine it captures groups that have their center points in a certain distance form each other. More than 4 clusters are too many for this small dataset.

12. bivariate regression

```
# load dataset trees
cl1 <- kcca(milk_matrix, k=4)

# plot the result
barplot(cl1)
```



The algorithm seems to have found:

- one cluster with milk having almost as much `water` content as content of `fat`
- one cluster with milk with high `water` content, however an clearly visible `fat` and `protein` content
- one cluster with milk having very high `water` content with low but similar content of `protein`, `fat` and `lactose`
- one cluster with milk having almost just `water` with some `lactose` content