Multivariate Statistical Analysis

Domenica Melone

Table of contents

Exploratory Data Analysis: Outliers and Normality **2**
- Correlation Matrix . . . 2
- Univariate Outliers . . . 5
- Boxplots . . . 6
- Univariate Normality . . . 8
- Scatter Plot of Area vs Population: Bivariate Outliers . . . 12
- Multivariate Normality . . . 14
- Multivariate Outliers . . . 15

Linear Algebra for Multivariate Statistics **17**
- 1 - Inverse . . . 17
- 2 - Eigenvalues . . . 18
- 3 - Total variance . . . 18
- 4 - Conditional distribution . . . 18
- 5 - The ellipse . . . 19

Principal Component Analysis **20**
- PCs on adjusted data . . . 20
- How many PCs to retain? . . . 23
- Interpretation of the first two PCs . . . 24
- Univariate Outliers with respect to the first three PCs . . . 26
- 3D Scatter Plot . . . 31
- Multivariate Normality . . . 31
- Multivariate Outliers . . . 32

```
library(corrplot)
library(RColorBrewer)
library(ellipse)
```

Exploratory Data Analysis: Outliers and Normality

Analyze the data set 'state.x77' containing 8 variables (and an added ninth variable for the population density) recorded to the 50 states of the United States of America in year 1977.

```
rm(list=ls())
st = as.data.frame(state.x77)
names(st)[4] = "Life.Exp"
names(st)[6] = "HS.Grad"
st[,9] = st$Population * 1000 / st$Area
colnames(st)[9] = "Density"
head(st)
```

```
           Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
California      21198   5114        1.1    71.71   10.3    62.6    20 156361
Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
              Density
Alabama     71.2905261
Alaska       0.6443845
Arizona     19.5032491
Arkansas    40.6198864
California 135.5708904
Colorado    24.4877898
```

Correlation Matrix

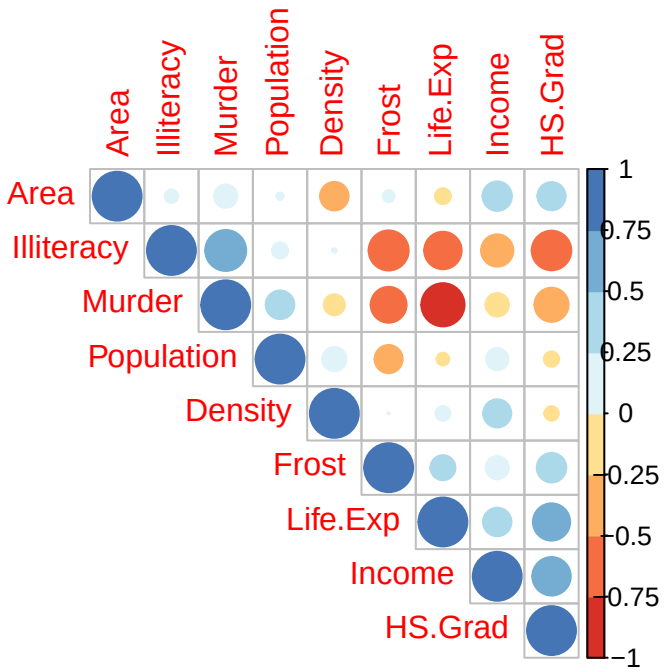
```
R = round(cor(st),3); R
```

```
           Population Income Illiteracy Life.Exp Murder HS.Grad  Frost   Area
Population      1.000  0.208      0.108   -0.068  0.344  -0.098 -0.332  0.023
Income          0.208  1.000     -0.437    0.340 -0.230   0.620  0.226  0.363
Illiteracy      0.108 -0.437      1.000   -0.588  0.703  -0.657 -0.672  0.077
Life.Exp       -0.068  0.340     -0.588    1.000 -0.781   0.582  0.262 -0.107
Murder          0.344 -0.230      0.703   -0.781  1.000  -0.488 -0.539  0.228
HS.Grad        -0.098  0.620     -0.657    0.582 -0.488   1.000  0.367  0.334
Frost          -0.332  0.226     -0.672    0.262 -0.539   0.367  1.000  0.059
Area            0.023  0.363      0.077   -0.107  0.228   0.334  0.059  1.000
Density         0.246  0.330      0.009    0.091 -0.185  -0.088  0.002 -0.341
```

```
            Density
Population    0.246
Income        0.330
Illiteracy    0.009
Life.Exp      0.091
Murder       -0.185
HS.Grad      -0.088
Frost         0.002
Area         -0.341
Density       1.000
```

This is the correlation matrix of the 9 variables. Let's check more visually their magnitude.

```
corrplot(R, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
```



With the visual aid it is obvious that some pairs of variables are way more correlated than others. Specifically, the six pairs of variables with the highest correlation coefficients are the following:

```
R[!lower.tri(R)]<-NA
p<-dim(st)[2]
stack.m<-matrix(1:p^2,ncol=p)
order.cor = order(abs(R),decreasing=T,na.last = NA)
my.fun<-function(x) which(stack.m==x,arr.ind=T)
my.fun<-Vectorize(my.fun)
```

```
Order.cor<-t(my.fun(order.cor))
colnames(Order.cor)<-c("row","col")
out<-matrix(names(st)[Order.cor],ncol=2)
out = cbind(out, round(R[Order.cor],3))
colnames(out)<-c("FirstVar","SecondVar", "correlation")
head(out)
```

```
     FirstVar   SecondVar    correlation
[1,] "Murder"   "Life.Exp"   "-0.781"
[2,] "Murder"   "Illiteracy" "0.703"
[3,] "Frost"    "Illiteracy" "-0.672"
[4,] "HS.Grad"  "Illiteracy" "-0.657"
[5,] "HS.Grad"  "Income"     "0.62"
[6,] "Life.Exp" "Illiteracy" "-0.588"
```

While the four pairs of variables with the lowest correlation coefficients are the following:

```
tail(out, 4)
```

```
      FirstVar  SecondVar    correlation
[33,] "Area"    "Frost"      "0.059"
[34,] "Area"    "Population" "0.023"
[35,] "Density" "Illiteracy" "0.009"
[36,] "Density" "Frost"      "0.002"
```

The two most (negatively) correlated variables are 'Murder' and 'Life.Exp' (Life Expectancy) with a correlation coefficient of -0.781. This is not too surprising since several studies, a lot of which made in the USA, have highlighted this possible connection. In particular, the impact of murder on life expectancy appears to be higher in low-income neighborhoods and on black individuals.

The couple 'Murder' and 'Illiteracy' follows with a correlation coefficient of 0.703. Here the interpretation is clear-cut. Indeed, several studies found that low literacy is significantly correlated with higher levels of criminal activity. Low literacy may contribute to poor decision-making and impulse control, as well as reduced opportunities for employment and social mobility, which could lead individuals to engage in violent criminal activity such as murder.

The variables 'Frost' and 'Illiteracy' show a correlation coefficient of -0.672. However, also 'Frost' and 'Murder' show an important correlation coefficient (-0.539). It may be a possibility that 'Frost' is a lurking variable causing a spurious association.

Two couples of variables in the list are not unforeseen: "HS.Grad" (the percentage of high-school graduates in 1970) and "Illiteracy" have a correlation coefficient of -0.657 and "HS.Grad" and "Income" of 0.62. One of the determinant of illiteracy is lack of schooling in

an individual's life, so the higher the analphabetism the lower must be graduates from high school. For the second couple we can notice that economists have been trying to estimate accurately how much the increase in the level of education makes the income raise, but even if they cannot come out with a unique estimate they can prove that for sure this has to be a positive number since a higher level of education (like high school graduating) is on average better remunerated.

Lastly, it is interesting to take a look at the two less correlated pairs of variables: "Density" and "Illiteracy" together with "Density" and "Frost". Both pairs have a correlation coefficient of nearly 0. For the first pair: one of the determinants of the level of illiteracy is the presence of schools in a town, the low correlation between density and illiteracy might indicate that even if a town is more crowded the USA can still adapt and provide a sufficient school service (there are enough schools for all citizens). As concerns frost and density, this correlation may suggest, among others, that people tend not to choose where to live depending on the weather (which may be due to not too extreme weathers in the USA and to well functioning heating/cooling systems).

Univariate Outliers

Outliers are very small or very large observations relative to the others and do not seem to belong to the pattern of variability produced by the other observations. They can be categorized into univariate, bivariate and, more generally, multivariate outliers. The three categories difference in the sense that, for example, a multivariate outlier might not show off as unusual when looking at one variable at the time (each of its component has a typical value). In the next part of the exercise we are going to follow the general steps for detecting outliers. Notice that, unlike univariate and bivariate, multivariate outliers cannot be seen visually (for the first two we can make boxplots and scatter plots).

```
a = as.data.frame(which(abs(scale(st)[,])>qnorm(0.99),arr.ind=T))
names(a) = c('Obs', 'Variable')
a$Variable = names(st)[a$Variable]; a
```

```
              Obs   Variable
California      5 Population
New.York       32 Population
Alaska          2     Income
Louisiana      18 Illiteracy
Alaska.1        2       Area
Massachusetts  21    Density
New.Jersey     30    Density
Rhode.Island   39    Density
```

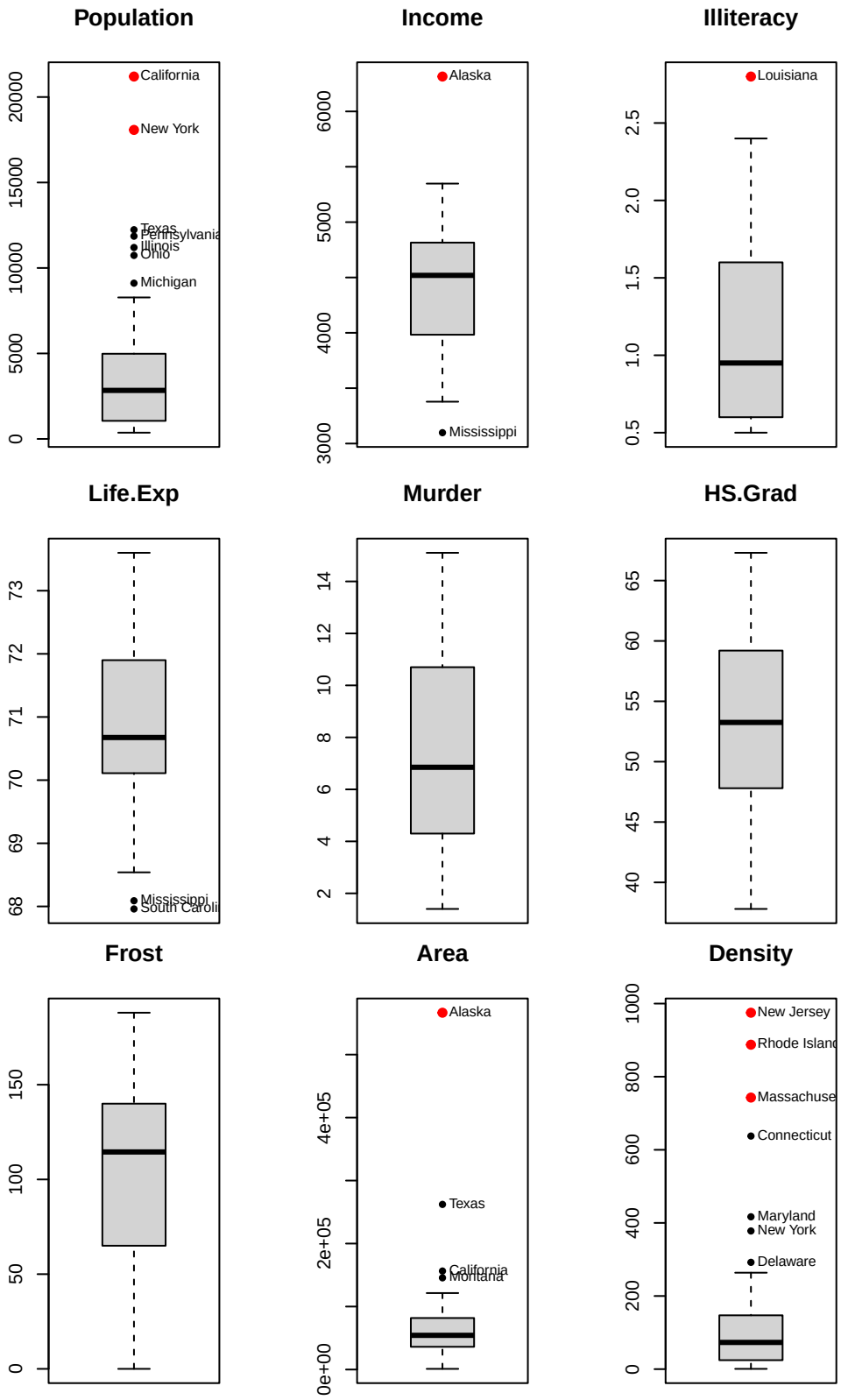
By assuming that the data are normally distributed, we can check for univariate outilers by standardizing (centering and scaling) each observation (so that they present zero mean and unit variance) and comparing them to the quantiles of the standard normal distribution.

In the table, the 8 observations that exceed the 99-th quantile of the standard normal distribution (which also corresponds to the quantile computed with the formula: (obs num - 0.5)/obs num) are presented.

It is good practice to combine this assessment method with the boxplot one. Boxplots are graphical methods that do not assumes any underlying distribution.

Boxplots

```
par(mfrow = c(1,3))
a=as.data.frame(which(abs(scale(st)[,]) > qnorm(0.99), arr.ind = TRUE))

for (j in 1:p) {
  values = st[,j]
  my.fun <- function(x) which(values == x, arr.ind = TRUE)
  my.fun <- Vectorize(my.fun)

  main.lab = names(st)[j]
  m = boxplot(values, main = main.lab, outpch = 19, range = 1)

  t = m[["out"]]

  if (length(t) != 0) {
    t_names <- my.fun(t)
    b = cbind(t_names, rep(j, length(t)))
    palla = which(a[,2] == b[,2])
    text(1, t, labels = as.character(rownames(st)[t_names]),
         pos = 4, cex = 0.75, offset = 0.3)
    if (length(palla) != 0) {
      cc = a[palla,1]
      points(rep(1, length(cc)), values[cc], pch = 19, col = 'red')
    }
  }
}
```



These are the 9 boxplots (one per variable) which identify several outliers: in red the eight

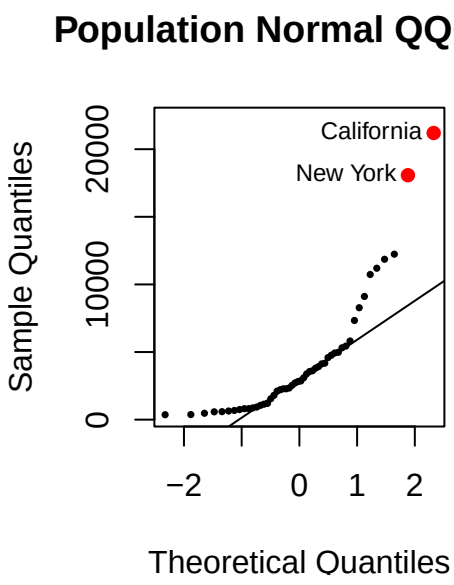
outliers highlighted in point 2. Notice that in the code for the boxplot there is the argument 'range' set equal to 1. The default value for it is 1.5, this determines how far the plot whiskers extend out from the box. Specifically, the whiskers extend to the most extreme data point which is no more than range times the interquartile range from the box. With 'range' equal to 1.5, Louisiana is not an outlier for Illiteracy (this may be due to the fact that Illitercay is pretty positively skewed while the Gaussian-which is the criteria used in point 2-is symmetric around the mean and therefore has skewness equal to zero and Louisiana is a high positive value). As concerns the other seven outliers found in point 2, they all appear to be outliers also in the boxplot even when 'range' is equal to 1.5. No observation is detected as outlier in all boxplots.

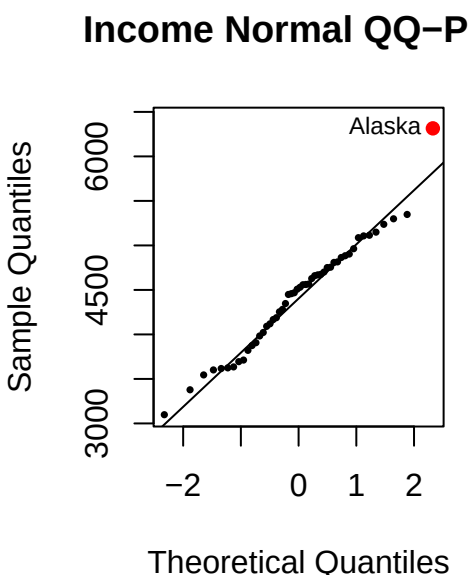
Finally, notice that all the observations found in point 2 (have to) correspond to the most extreme values, namely to the "farthest" outliers detected by the boxplot.

Univariate Normality

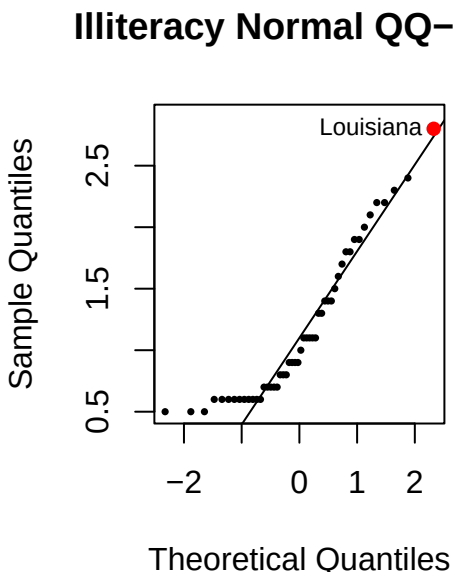
To assess univariate normality, a comparison between the empirical quantiles of the data and the theoretical normal quantiles (visually through a QQ-plot) is usually recommended (this method is used when normality of the data is assumed).

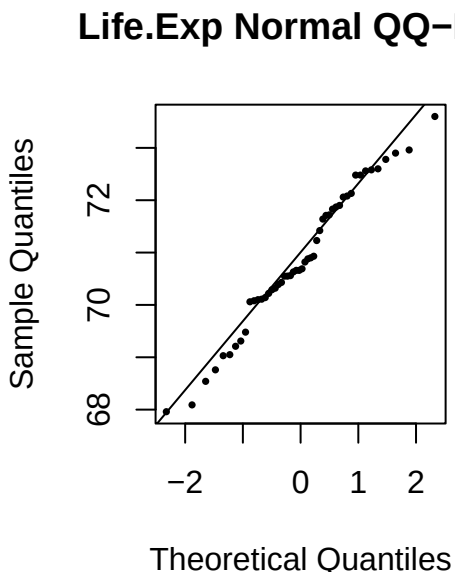
```
par(mfrow = c(1,2))

#Population
qqnorm(st[,1], main = "Population Normal QQ-Plot",
       ylim = c(min(st[,1]), max(st[,1])+1000), pch = 16, cex = 0.5)
qqline(st[,1])
qnor = qnorm(ppoints(st[,1]))
out.qq = match(c(5, 32), order(st[,1]))
points(qnor[out.qq], sort(st[,1])[out.qq], pch = 16, col = 'red')
text(qnor[out.qq], sort(st[,1])[out.qq], labels = c("California", "New York"),
     pos = 2, cex = 0.75, offset = 0.3)

#Income
qqnorm(st[,2], main = "Income Normal QQ-Plot", ylim = c(min(st[,2]), max(st[,2])+100),
       pch = 16, cex = 0.5)
qqline(st[,2])
qnor = qnorm(ppoints(st[,2]))
out.qq = match(2, order(st[,2]))
points(qnor[out.qq], sort(st[,2])[out.qq], pch = 16, col = 'red')
text(qnor[out.qq], sort(st[,2])[out.qq], labels = "Alaska", pos = 2,
     cex = 0.75, offset = 0.3)
```

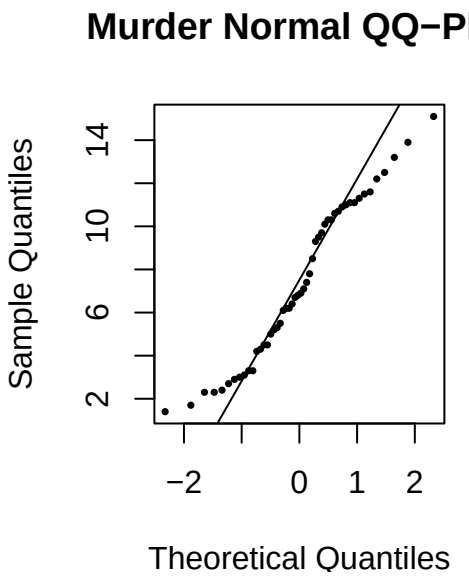


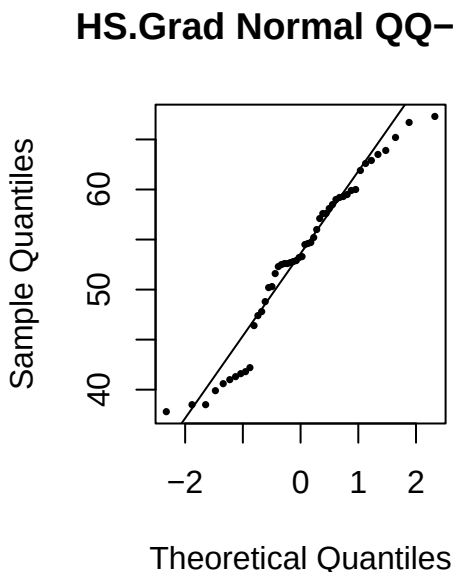



```
#Illiteracy
qqnorm(st[,3], main = "Illiteracy Normal QQ-Plot",
       ylim = c(min(st[,3]), max(st[,3])+0.1), pch = 16, cex = 0.5)
qqline(st[,3])
qnor = qnorm(ppoints(st[,3]))
out.qq = match(18, order(st[,3]))
points(qnor[out.qq], sort(st[,3])[out.qq], pch = 16, col = 'red')
text(qnor[out.qq], sort(st[,3])[out.qq], labels = "Louisiana",
     pos = 2, cex = 0.75, offset = 0.3)

#from Life.Exp to Frost
for (j in 4:7) {
       values = st[,j]
       main.lab = names(st)[j]
       qqnorm(values, main = paste(main.lab, "Normal QQ-Plot"),cex=0.5, pch=16)
       qqline(values)
}
```

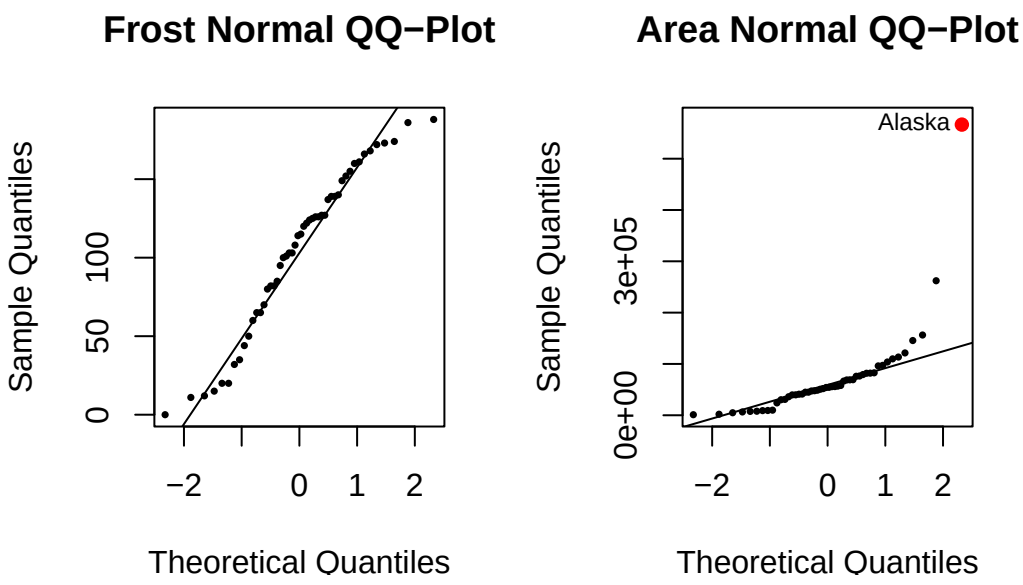




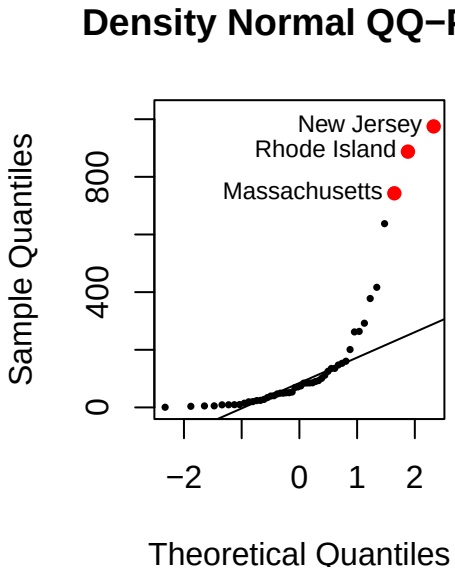





```
#Area
qqnorm(st[,8], main = "Area Normal QQ-Plot", ylim = c(min(st[,8]), max(st[,8])+10000),
       pch = 16, cex = 0.5)
qqline(st[,8])
qnor = qnorm(ppoints(st[,8]))
out.qq = match(2, order(st[,8]))
points(qnor[out.qq], sort(st[,8])[out.qq], pch = 16, col = 'red')
text(qnor[out.qq], sort(st[,8])[out.qq], labels = "Alaska",
     pos = 2, cex = 0.75, offset = 0.3)
```




```
#Density
qqnorm(st[,9], main = "Density Normal QQ-Plot", ylim = c(min(st[,9]), max(st[,9])+50),
       pch = 16, cex = 0.5)
qqline(st[,9])
qnor = qnorm(ppoints(st[,9]))
out.qq = match(c(21,30,39), order(st[,9]))
points(qnor[out.qq], sort(st[,9])[out.qq], pch = 16, col = 'red')
text(qnor[out.qq], sort(st[,9])[out.qq],
     labels = c("Massachusetts", "New Jersey", "Rhode Island"),
     pos = 2, cex = 0.75, offset = 0.3)
```



In red are highlighted outliers from point 2.

For the variable 'Population', the bulk of the distribution appears quite normal, while the tails appear to be a bit too fat. Specifically, the left tail is more concentrated around a small interval of values (the smallest nine obs range from 365 to 812), while the right tail reaches higher values and shows more variability (indeed the range of values for the largest nine obs goes from 7333 to 21198). Hence, the right tail looks fatter and we can conclude about positive skewness. A very similar comment can be done with respect to 'Density', last QQ-Plot.

The normality distribution assumption for the variables 'Income' and 'Area' seems satisfied but for 1 observation (the outlier 'Alaska' from point 2).

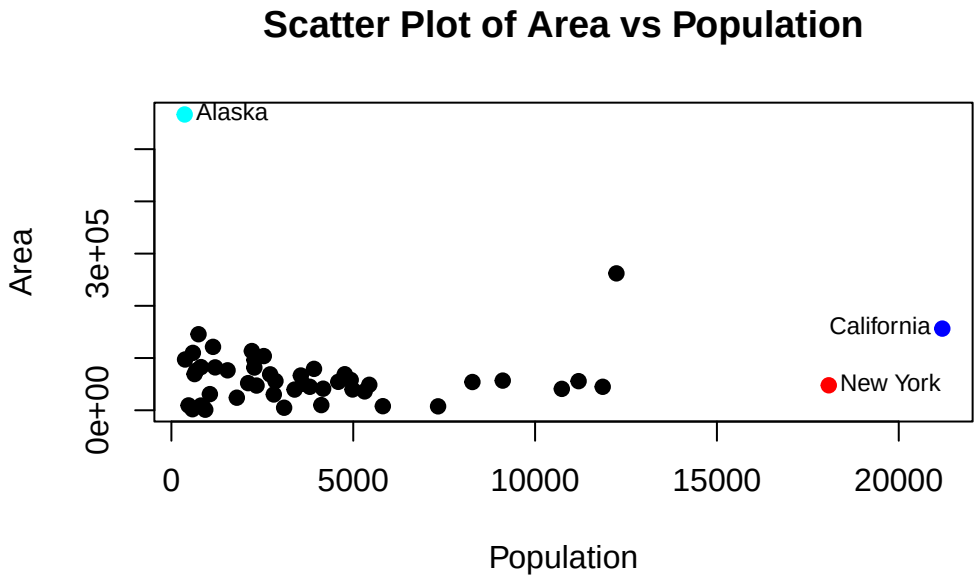
For the variable 'Illiteracy', we can notice that Louisiana is above the 99-th quantile of the Gaussian, but it still appears to be normally distributed. Moreover, in the left tail we can notice a "spike" of identical values.

For 'Life.Exp', 'Murder', 'HS.Grad' and 'Frost' the distributions appear quite normal (maybe with the possible exception of some values in the tails).

Scatter Plot of Area vs Population: Bivariate Outliers

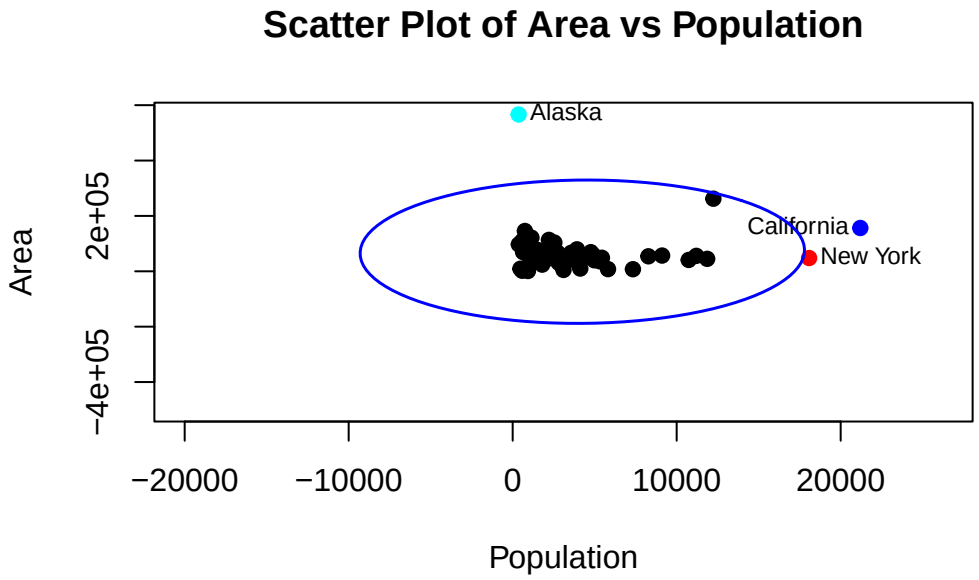
To check bivariate outliers, scatter plots of pairs of variables can be used.

```
col = rep('black', dim(st)[1])
col[5] = 'blue'
col[32] = 'red'
col[2] = 'cyan'
plot(st$Population, st$Area, xlab = "Population", ylab = "Area", pch = 19,
     col = col, main = "Scatter Plot of Area vs Population")

text(st$Population[5], st$Area[5], labels = rownames(st)[5],
     pos = 2, cex = 0.75, offset = 0.3)
text(st$Population[32], st$Area[32], labels = rownames(st)[32],
     pos = 4, cex = 0.75, offset = 0.3)
text(st$Population[2], st$Area[2], labels = rownames(st)[2],
     pos = 4, cex = 0.75, offset = 0.3)
```



It is clear from the plot that the three univariate outliers found in point 2 ('Alaska' for 'Area', 'California' and 'New York' for 'Population') differ from the compact pattern formed by the rest of the data. To statistically check if they can be considered not only univariate outliers, but bivariate outliers too, one can compute the Mahalanobis distances of this subset of data points. These should approximately distribute (since normality of the underlying data is assumed) as a chi-squared with 2 degrees of freedom. Hence, we can draw an ellipse (which is a level set of the multivariate normal density) that should contain 0.99% of the observations (which is equivalent to say that Mahalanobis distances are below the 99th quantile of the chi-squared distribution).

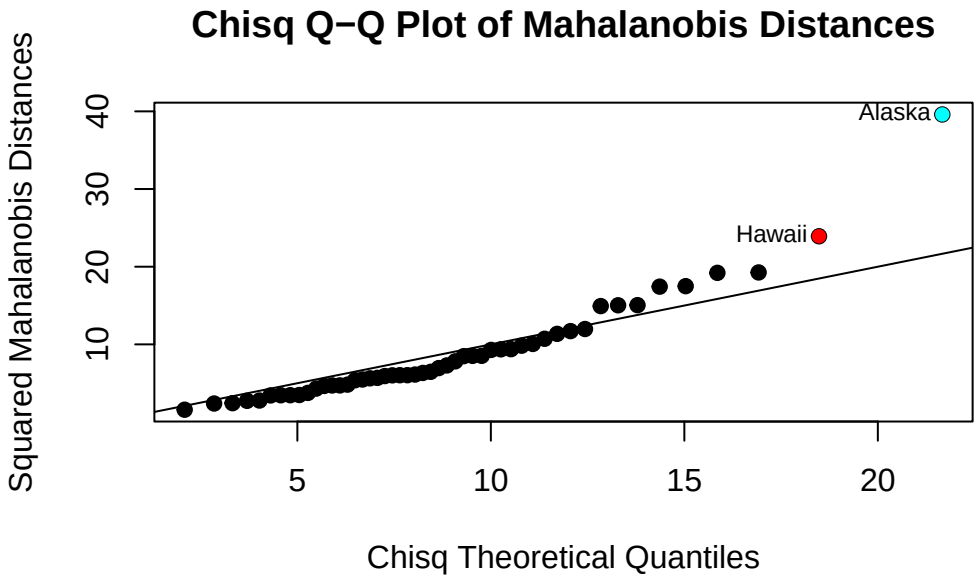
```
col = rep('black', dim(st)[1])
col[5] = 'blue'
col[32] = 'red'
col[2] = 'cyan'
plot(st$Population, st$Area, xlab = "Population", ylab = "Area", pch = 19,
     col = col, main = "Scatter Plot of Area vs Population",
     ylim = c(-500000, max(st$Area)) ,xlim = c(-20000, max(st$Population)+5000))

text(st$Population[5], st$Area[5], labels = rownames(st)[5],
     pos = 2, cex = 0.75, offset = 0.3)
text(st$Population[32], st$Area[32], labels = rownames(st)[32],
     pos = 4, cex = 0.75, offset = 0.3)
text(st$Population[2], st$Area[2], labels = rownames(st)[2],
     pos = 4, cex = 0.75, offset = 0.3)
lines(ellipse(x=cov(st[,c(1,8)]), centre=colMeans(st[,c(1,8)]),level=0.99),
      col="blue",lwd=1.5)
```



All the three univariate outliers fall outside the ellipse, meaning that the value of their Mahalanobis distances exceeds the 99th quantile of the chi-squared distribution and therefore we can conclude that they can be considered bivariate outliers too.

Multivariate Normality

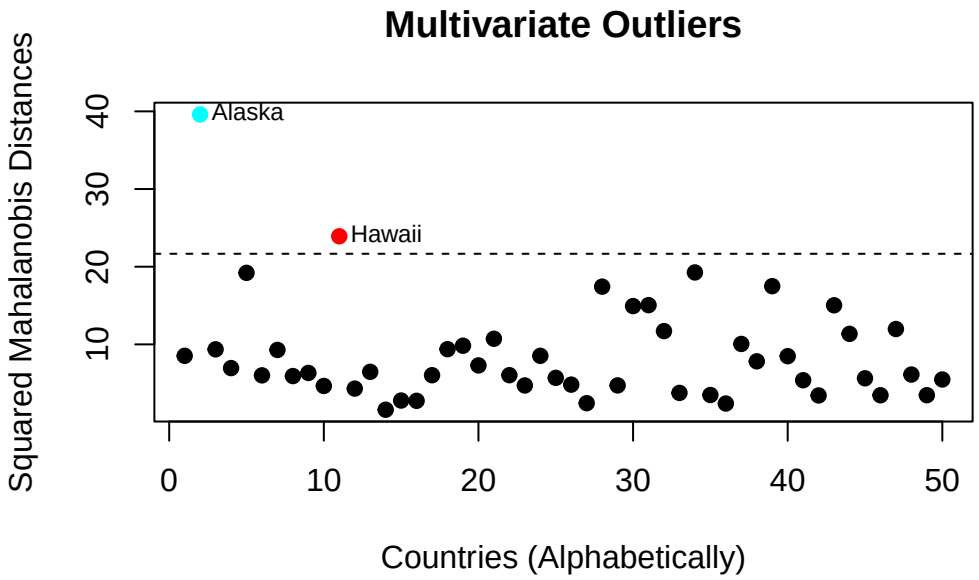
```
d = mahalanobis(st, center = colMeans(st), cov = cov(st))
plot(qchisq(ppoints(d), df = dim(st)[2]), sort(d), pch = 19,
     xlab = "Chisq Theoretical Quantiles", ylab = "Squared Mahalanobis Distances",
     main="Chisq Q-Q Plot of Mahalanobis Distances")
abline(0,1)

qchi = qchisq(ppoints(d), df = dim(st)[2])
n = dim(st)[1]
out.qq = match((order(d)[c(n, n-1)]), order(d))

points(qchi[out.qq], sort(d)[out.qq], pch = 16, col = c('cyan', 'red'))
text(qchi[out.qq], sort(d)[out.qq], labels = c("Alaska", "Hawaii"),
     pos = 2, cex = 0.75, offset = 0.3)
```



The theoretical chi-squared pattern (here the theoretical quantiles are from a chi-squared distribution with p = 9 degrees of freedom) seems satisfied but for the two observations 'Alaska' and 'Hawaii'. The assumption of Multivariate Normality seems overall respected.

Multivariate Outliers

```
n = dim(st)[1]
col2 = rep('black', n)
col2[order(d)[n]] = 'cyan'
col2[order(d)[n-1]] = 'red'
plot(d,pch=19, col = col2, main = "Multivariate Outliers",
     ylab = "Squared Mahalanobis Distances", xlab = "Countries (Alphabetically)")
abline(h=qchisq((n-0.5)/n,df=dim(st)[2]),lty=2) #confidence level = 1%
text(order(d)[n], sort(d)[n], labels = names(sort(d)[n]),  pos = 4, cex = 0.75,
     offset = 0.3)
text(order(d)[n-1], sort(d)[n-1], labels = names(sort(d)[n-1]),
     pos = 4, cex = 0.75, offset = 0.3)
```

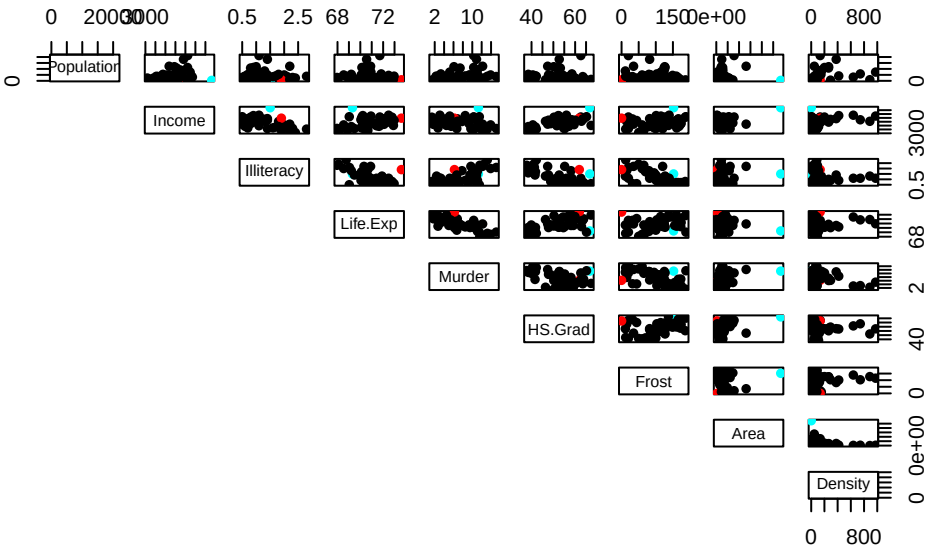


The two observations that in point 6 did not meet the multivariate normality assumption are here in point 7 highlighted as multivariate outliers (they are above the quantile (n-0.5)/n of the chi-squared distribution with p = 9 dof).

Notice that 'Hawaii' was not a univariate outlier for any variable. On the other hand, 'Alaska' was a univariate outlier for both 'Income' and 'Area'; moreover, notice that it was the maximum value for both variables and, additionally, it was quite far away from the rest of the observations. These may be the reasons why even in higher dimensions (from the univariate dimension where p = 1, to the bivariate dimension where p = 2 and finally to the multivariate dimension where p = 9) this observation persists as an outlier.

Indeed, if we look at the scatter plots of all pairs of variables, 'Alaska' often diverges from the main pattern of the data, while 'Hawaii' often stays in the cloud of points.

```
n = dim(st)[1]
col = rep('black', n)
col[2] = 'cyan'
col[11] = 'red'
pairs(st, lower.panel = NULL, pch = 16, col = col)
```



Linear Algebra for Multivariate Statistics

1 - Inverse

As in the advice we use a to denote $(1, 1, -1)^t$. Then we can observe that the matrix aa^t has an interesting property:

$$(aa^t)^2 = (aa^t)(aa^t) = a(a^t a)a^t = 3aa^t$$

So there is a similar property to the idempotent matrices and, as we do with the last ones, we can search an inverse of the form $\mathbf{I} + baa^t$ for some b to be determined. We choose this form in order to use the property of before:

$$\begin{aligned}((1+\rho)\mathbf{I} - \rho aa^t)(\mathbf{I} + baa^t) &= (1+\rho)\mathbf{I} - \rho aa^t + (1+\rho)baa^t - 3\rho baa^t \\ &= (1+\rho)\mathbf{I} + (-\rho + (1+\rho)b - 3\rho b)aa^t \\ &= (1+\rho)\mathbf{I} + (-\rho + b - 2\rho b)aa^t\end{aligned}$$

Since we want only the identity what is inside between the second brackets must be equal to zero. So $b = \frac{\rho}{1-2\rho}$. So we can deduce that:

$$\Sigma^{-1} = \frac{1}{1+\rho}\left(\mathbf{I} + \frac{\rho}{1-2\rho}aa^t\right)$$

2 - Eigenvalues

To complete this point we can do the typical computations, setting $t = 1 - \lambda$:

$$\begin{aligned}
\det(\Sigma - \lambda \mathbf{I}) &= \begin{vmatrix} t & -\rho & \rho \\ -\rho & t & \rho \\ \rho & \rho & t \end{vmatrix} \\
&= t\left(t^2 - \rho^2\right) + \rho(-\rho^2 - \rho t) - \rho\left(t\rho + \rho^2\right) \\
&= t^3 - 3\rho^2 t - 2\rho^3 \\
&= (t^3 - \rho t^2 - 2\rho^2 t) + (\rho t^2 - \rho^2 t - 2\rho^3) \\
&= (t + \rho)(t^2 - \rho t - 2\rho^2) \\
&= (t + \rho)^2 (t - 2\rho)
\end{aligned}$$

so we can deduce that $\lambda_1 = \lambda_2 = 1 + \rho$ and $\lambda_3 = 1 - 2\rho$ are the eigenvalues.

3 - Total variance

Observe that the sums of the eigenvalues is $2(1 + \rho) + 1 - 2\rho = 3$. We considers the three cases:

- When $\rho = 0$ then in this case $\lambda_1 = \lambda_2 = \lambda_3$ and so in this case the PC1 and the PC2 describes only the $66.\overline{6}\% < 80\%$ of the total variance. So we have to remove $\rho = 0$.
- When $\rho > 0$, then $\lambda_1 = \lambda_2 > \lambda_3$ so the variation of the PC1 and PC2 is $\frac{2}{3}(1 + \rho)$. So from $\frac{2}{3}(1 + \rho) > \frac{4}{5}$ we can deduce that $\rho > \frac{1}{5}$.
- When $\rho < 0$, then $\lambda_1 = \lambda_2 < \lambda_3$ so the variation of the PC1 and PC2 is $\frac{1}{3}(2 - \rho)$. So from $\frac{1}{3}(2 - \rho) > \frac{4}{5}$ we can deduce that $\rho < -\frac{2}{5}$.

So the ρ's that satisfy the request are the member of the set $\left(-1, -\frac{2}{5}\right) \cup \left(\frac{1}{5}, \frac{1}{2}\right)$.

4 - Conditional distribution

To compute the conditional distribution in a handy way we rewrite the order of the variables. We can deduce that

$$(Y_1, Y_2, X) \sim N_3(\tilde{\mu}, \tilde{\Sigma}) \quad \text{where} \quad \tilde{\mu} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} \quad \tilde{\Sigma} = \begin{pmatrix} 1 & \rho & -\rho \\ \rho & 1 & \rho \\ -\rho & \rho & 1 \end{pmatrix}$$

So we can use the classical formula for the conditional distribution of a Gaussian given a Gaussian and we obtain that:

$$Y_1, Y_2 \mid X = x \sim N_2(\mu', \Sigma') \quad \text{where} \quad \mu' = \begin{pmatrix} 0 \\ 2 \end{pmatrix} + (x-1)\begin{pmatrix} -\rho \\ \rho \end{pmatrix} = \begin{pmatrix} (1-x)\rho \\ 2 + (x-1)\rho \end{pmatrix}$$

$$\text{and} \quad \Sigma' = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} - \begin{pmatrix} -\rho \\ \rho \end{pmatrix}(-\rho \quad \rho) = \begin{pmatrix} 1-\rho^2 & \rho+\rho^2 \\ \rho+\rho^2 & 1-\rho^2 \end{pmatrix}$$

5 - The ellipse

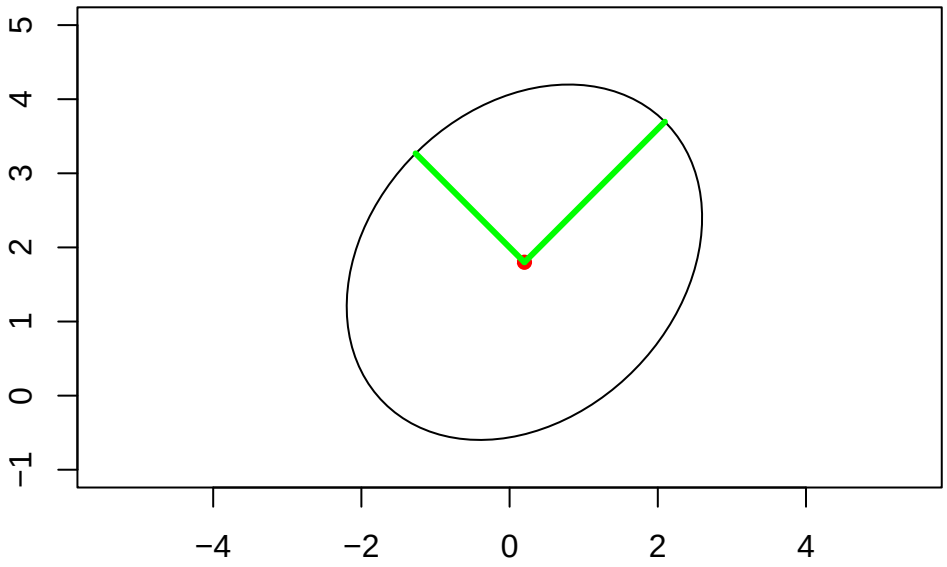
Since $Y_1, Y_2 \mid X = x \sim N_2(\mu', \Sigma')$ then there is a theorem that ensures that $(Y - \mu')^t \Sigma'^{-1}(Y - \mu') \sim \chi^2_2$. So in order to compute c^2 is sufficient to compute the 95%-quantile of the chi squared (so c is the square root):

```
remove(list = ls())
(cc <- sqrt(qchisq(0.95, 2)))
```

```
[1] 2.447747
```

So the ellipse we want is centered in μ' and their principal axes lie on the directions determined by the eigenvector of Σ' and their lengths are $c\sqrt{\lambda'_1}$, for the axis that lies on the first eigenvalue, and $c\sqrt{\lambda'_2}$, for the other one.

```
p <- 0.2
x <- 0
y.mu <- c(0,2) + (x-1)*c(-p,p)
y.si <- matrix(c(1 - p^2, p + p^2, p + p^2, 1 - p^2), 2)
egn <- eigen(y.si)
e1 <- cc*sqrt(egn$values[1])*egn$vectors[,1] + y.mu
e2 <- cc*sqrt(egn$values[2])*egn$vectors[,2] + y.mu

par(mar = c(3,3,2,2))
plot(y.mu[2] ~ y.mu[1], pch = 21, cex = 1.2, col = "white",
     bg = "red", ylim = c(-1,5), xlim = c(-3,3), xlab = "",
     ylab = "", asp = 1)
lines(ellipse(x = y.si, centre = y.mu, t = cc))
arrows(y.mu[1], y.mu[2], e1[1], e1[2], code = 0, col = "green", lwd = 3)
arrows(y.mu[1], y.mu[2], e2[1], e2[2], code = 0, col = "green", lwd = 3)
```



Principal Component Analysis

Analyze the dataset about nutritional information from 961 different food items.

```
rm(list=ls())
nut = read.table("nutritional.txt")

head(nut)
```

```
  fat food.energy carbohydrates protein cholesterol weight saturated.fat
1   2          25             2       0           2  15.00           0.2
2   6          60             2       0           4  16.00           1.0
3   1          90            22       4           0  28.35           0.1
4   0          90            22       3           0  28.35           0.1
5   0          10             1       1           0  33.00           0.0
6   1          70            21       4           0  28.35           0.1
```

PCs on adjusted data

To equalize out the different types of servings of each food, first we divide each variable by weight of the food item (which leaves us with 6 variables). Next, because of the wide variations in the different variables, we standardize each variable.

```
nut = nut/nut[,6]
nut = nut[,-6]
nut = scale(nut)
head(nut)
```

```
         fat food.energy carbohydrates    protein  cholesterol saturated.fat
1  0.1038714  -0.3030594    -0.4195816 -0.7778908 -0.181557884    -0.3604211
2  1.3529896   0.7732295    -0.4529688 -0.7778908 -0.008841597     0.3830582
3 -0.4029775   0.4759687     2.1552905  0.7909259 -0.378947926    -0.5087036
4 -0.5852973   0.4759687     2.1552905  0.3987218 -0.378947926    -0.5087036
5 -0.5852973  -1.0075394    -0.8323680 -0.4409517 -0.378947926    -0.5620426
6 -0.4029775   0.1115111     2.0139693  0.7909259 -0.378947926    -0.5087036
```

```
#in one line
#nut = scale(((nut/nut[,6])[,-6]))
```

Finally, we can perform Principal Component Analysis on the transformed data.

```
nut.pca = prcomp(nut, center = F, scale. = F)
PCs = round(nut.pca$x,3); head(PCs) #the scores of every PCs for each food item
```

```
     PC1    PC2    PC3    PC4    PC5    PC6
1  0.515  0.013 -0.771  0.171  0.397  0.031
2 -1.198  0.393 -1.124  0.153  0.712  0.055
3  0.202  1.489  1.916 -0.188 -0.190  0.246
4  0.396  1.618  1.716  0.071 -0.213  0.087
5  1.344 -0.603 -0.709 -0.178  0.022 -0.008
6  0.394  1.264  1.786 -0.201 -0.268  0.452
```

```
summary(nut.pca) #variance explained
```

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6
Standard deviation     1.6274 1.1533 1.0100 0.8247 0.51626 0.23356
Proportion of Variance 0.4414 0.2217 0.1700 0.1133 0.04442 0.00909
Cumulative Proportion  0.4414 0.6631 0.8331 0.9465 0.99091 1.00000
```

In the first table we can take a look at the scores of the six PCs for the first six observations.

In the second table, in the first line, we can read the square roots of the six eigenvalues of the covariance matrix of our data. These are the standard deviations of the PCs. The closest to each other are these values, the more the respective PCs explain approximately the same amount of overall variability and a dimension reduction is harder to achieve. Here we can notice that the second and third standard deviations are not too far away from each other.

In the third line, we can read that the proportion of variance explained by the first two PCs is only 66.31% and, according to a lot of criteria for deciding the number of PCs to retain, we cannot reach a dimensionality reduction from $p = 6$ to $p = 2$.

We can search for a possible explanation to this by looking at the correlations among variables. Indeed, if the correlations among variables are small the dimensionality of the data is close to p (the number of variables), the eigenvalues will be nearly equal and the PCs will duplicate the original variables so that we cannot obtain a useful reduction in dimension and in extreme cases PCA would be of no use.

Let's take a look to the correlation matrix of our data.

```
R = cor(nut); R
```

```
                    fat food.energy carbohydrates     protein cholesterol
fat             1.0000000   0.8159028   -0.14039306  0.15729870   0.1746512
food.energy     0.8159028   1.0000000    0.32432222  0.26061931   0.1357708
carbohydrates -0.1403931   0.3243222    1.00000000 -0.08670418  -0.1625949
protein         0.1572987   0.2606193   -0.08670418  1.00000000   0.3279447
cholesterol     0.1746512   0.1357708   -0.16259492  0.32794472   1.0000000
saturated.fat   0.7465236   0.6429217   -0.14054906  0.14204012   0.3105808
              saturated.fat
fat               0.7465236
food.energy       0.6429217
carbohydrates    -0.1405491
protein           0.1420401
cholesterol       0.3105808
saturated.fat     1.0000000
```

In particular, we can order the correlation coefficients and check the six highest values.

```
R[!lower.tri(R)]<-NA
p<-dim(nut)[2]
stack.m<-matrix(1:p^2,ncol=p)
order.cor = order(abs(R),decreasing=T,na.last = NA)
my.fun<-function(x) which(stack.m==x,arr.ind=T)
my.fun<-Vectorize(my.fun)
Order.cor<-t(my.fun(order.cor))
colnames(Order.cor)<-c("row","col")
out<-matrix(colnames(nut)[Order.cor],ncol=2)
out = cbind(out, round(R[Order.cor],3))
colnames(out)<-c("FirstVar","SecondVar", "correlation")
head(out)
```

```
     FirstVar        SecondVar     correlation
[1,] "food.energy"   "fat"         "0.816"
[2,] "saturated.fat" "fat"         "0.747"
[3,] "saturated.fat" "food.energy" "0.643"
[4,] "cholesterol"   "protein"     "0.328"
```

```
[5,] "carbohydrates" "food.energy" "0.324"
[6,] "saturated.fat" "cholesterol" "0.311"
```

Apart from the first three pairs of variables, correlation coefficients are not too high and this could be a possible explanation to our PCA output.

How many PCs to retain?

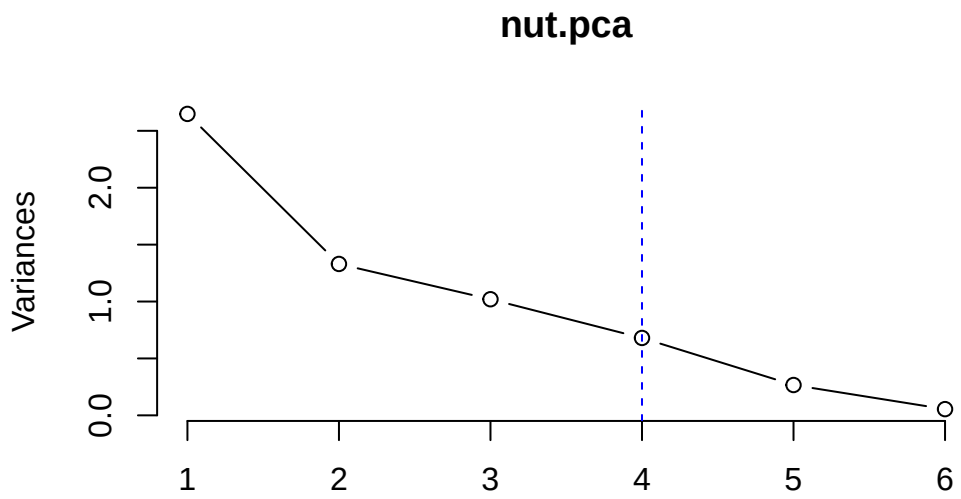
Reduction of dimensionality is attained by retaining, in statistical analysis, only the first PCs, i.e. those with larger variances .The last PCs convey just a small amount of information since their variances are small and therefore are dropped. Several criteria exist to decide on the number of components to retain.

1. Numerical Criterion: choose the first k components that explain at least between 70% and 80% of the variability of the data.

   According to this criterion, we saw from the Cumulative Proportion in the second table that we should retain at least the first three components.

2. Graphical Criterion: screeplot. Choose the first k components when observing a bend in the curve at k+1.

   ```
   screeplot(nut.pca, type=c("lines"))
   abline(v=4,lty=2,col="blue")
   ```

   Here the interpretation is far from clear-cut: the highest bend is at k+1 = 2, but retaining only the first PC would be accounting for only 44% of the overall variability which is too little. Another, smaller, bend is at k+1 = 5 so we could decide to retain 4 PCs by looking at the screeplot.

3. Kaiser criterion: retain components that account for more variance than the average variance. Notice that when the data is scaled and the covariance matrix equals the correlation matrix, average variance is simply 1.

```
nut.pca$sdev^2>mean(nut.pca$sdev^2)
```

```
[1]  TRUE  TRUE  TRUE FALSE FALSE FALSE
```

According to this criteria we should retain three components.

4. Joliffe criterion: the Kaiser rule is to drop all components with eigenvalues under 1.0 (for standardized data). Jolliffe suggested a more liberal criterion: retain eigenvalues greater than 0.7.

```
nut.pca$sdev^2>0.7
```

```
[1]  TRUE  TRUE  TRUE FALSE FALSE FALSE
```

Again, retain three components.

Comparing all the criteria, three is a good number of components to retain so that we can reduce the number of variables, avoiding loss of relevant information.

Interpretation of the first two PCs

```
eigenvec = nut.pca$rotation; eigenvec #the 6 eigenvectors
```

```
                    PC1         PC2        PC3         PC4         PC5
fat           -0.55723936  0.09870077 -0.2750890 -0.13040139  0.4546980
food.energy   -0.53615066  0.35676646  0.1370762 -0.07454684  0.2729547
carbohydrates  0.02455362  0.67163163  0.5684779  0.28616806 -0.1568663
protein       -0.23522713 -0.37384298  0.6388770 -0.59910351 -0.1538186
cholesterol   -0.25250455 -0.52130441  0.3256120  0.71709615  0.2102965
saturated.fat -0.53135067 -0.01923360 -0.2611169  0.14964683 -0.7913619
                    PC6
fat            0.616695791
food.energy   -0.697430105
carbohydrates  0.344444078
protein        0.118998503
cholesterol   -0.002904374
saturated.fat  0.021604346
```

These are the eigenvectors of our covariance (correlation) matrix, also called PCs' coefficients or loadings. They can be interpreted as the contribution of the original variables to the PCs. Indeed, the higher the loading, the higher the correlation between the component and the original variable.

```
R = cor(nut); R
```

```
                    fat food.energy carbohydrates     protein cholesterol
fat            1.0000000   0.8159028   -0.14039306  0.15729870   0.1746512
food.energy    0.8159028   1.0000000    0.32432222  0.26061931   0.1357708
carbohydrates -0.1403931   0.3243222    1.00000000 -0.08670418  -0.1625949
protein        0.1572987   0.2606193   -0.08670418  1.00000000   0.3279447
cholesterol    0.1746512   0.1357708   -0.16259492  0.32794472   1.0000000
saturated.fat  0.7465236   0.6429217   -0.14054906  0.14204012   0.3105808
               saturated.fat
fat                0.7465236
food.energy        0.6429217
carbohydrates     -0.1405491
protein            0.1420401
cholesterol        0.3105808
saturated.fat      1.0000000
```

```
D.sq = diag(sqrt(diag(R)))
Lambda.sq = diag(nut.pca$sdev)

Corr_Coeff = solve(D.sq)%*%eigenvec%*%Lambda.sq; Corr_Coeff
```

```
            [,1]       [,2]       [,3]        [,4]        [,5]          [,6]
[1,] -0.90687912  0.1138330 -0.2778434 -0.10753724  0.23474252  0.1440384126
[2,] -0.87255831  0.4114640  0.1384487 -0.06147604  0.14091569 -0.1628951045
[3,]  0.03995979  0.7746026  0.5741699  0.23599229 -0.08098385  0.0804500030
[4,] -0.38282035 -0.4311586  0.6452739 -0.49405867 -0.07941043  0.0277938585
[5,] -0.41093848 -0.6012280  0.3288723  0.59136287  0.10856774 -0.0006783594
[6,] -0.86474657 -0.0221824 -0.2637314  0.12340825 -0.40854869  0.0050460143
```

By taking a look to the matrix containing the correlations between original variables and PCs we get the same ranking we got by looking at the matrix containing the loadings. The most important variables to the first component are 'fat', 'food.energy', and 'saturated.fat'. The most important variables to the second component are 'carbohydrates' and 'cholesterol'.

Notice that the first component is mostly determined by the three variables which belong to the three pairs of most highly correlated variables.

Moreover, by looking at the sign of the loadings, the first component is mostly a weighted sum of fat, saturated fat and calories. Hence, we can interpret this component as a measure of the level of fats and calories provided by the food: a low score indicates high values for these three nutrients.

Many high-fat foods have a lot of saturated fat too (e.g. pizza). An increase of fat and saturated fat in the diet can lead to higher levels of calories. All fats contain 9 calories per

gram of fat. This is more than twice the amount found in carbohydrates and protein. So this might be a reason why the three variables are together. Hence, PC1 can be seen as a measure of **food energy**.

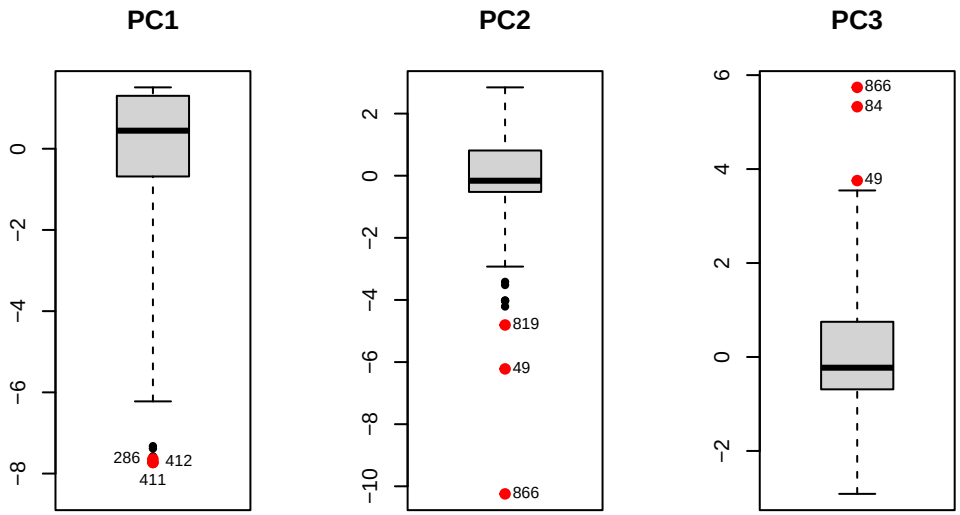
The second component, according to the signs, is mostly a weighted difference of carbohydrates and cholesterol. Food such as complex carbs have high levels of carbohydrates and have low levels of (help reduce) cholesterol. Hence, PC2 can be seen as a measure of **cholesterol-friendly food**: positive values of PC2 indicate healthy food.

Finally, notice also that carbohydrates has very low loading in PC1 and very high loading in PC2.

Univariate Outliers with respect to the first three PCs

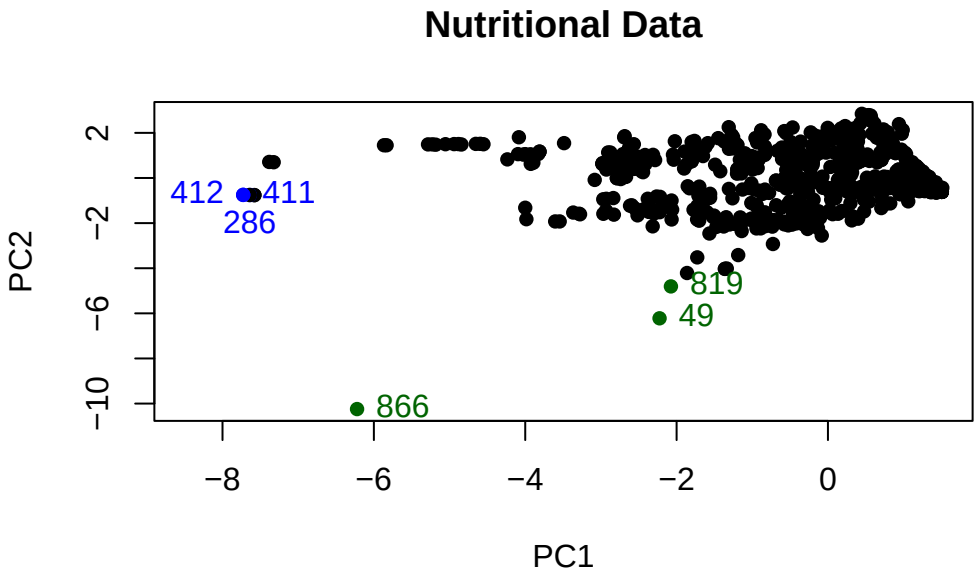
By looking at the boxplot for each PC we can find (at least) three univariate outliers per component.

```
par(mfrow = c(1,3))
bp = boxplot(PCs[,1], main = "PC1", outpch = 19, range = 3,
             ylim = c(-8.5,max(PCs[,1])))
out = bp[["out"]]
real_out = names(head(sort(out), 3))
text(1, head(sort(out), 3), labels = real_out, pos = c(1,4,2), cex = 0.75)
points(rep(1, length(real_out)), head(sort(out), 3), pch = 19, col = 'red')

retained = 3
for (j in 2:retained) {
  values = PCs[,j]
  main.lab = paste("PC", as.character(j), sep = '')
  bp = boxplot(values, main = main.lab, outpch = 19, range = 2)
  out = bp[["out"]]
  real_out = names(head(sort(out), 3))
  text(1, head(sort(out), 3), labels = real_out, pos = 4, cex = 0.75, offset = 0.3)
  points(rep(1, length(real_out)), head(sort(out), 3), pch = 19, col = 'red')
}
```



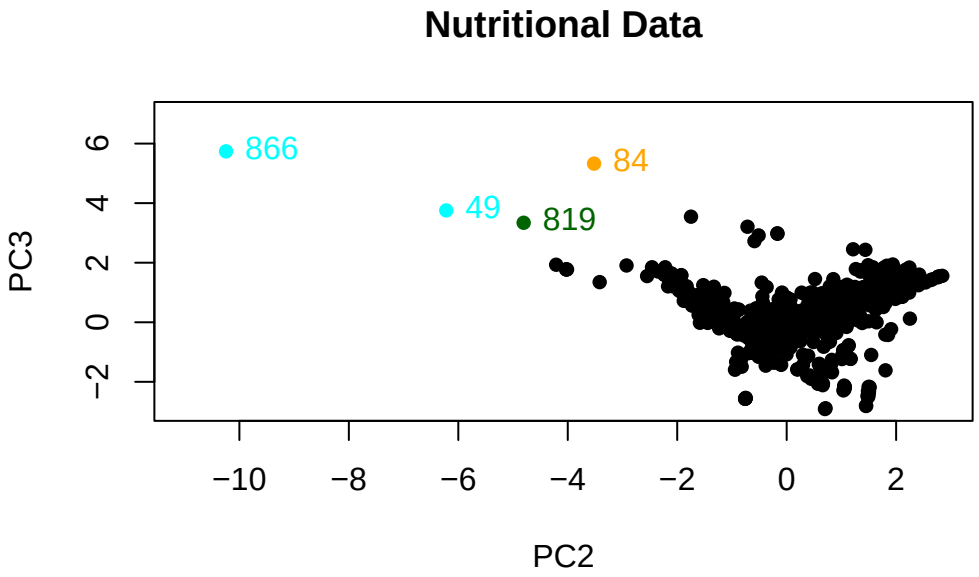
We can also take a look at the scatter plots of pairs of PCs and notice if the univariate outliers found via the boxplots differ from the main pattern of the rest of the observations.

```
col.ind = rep('black', dim(nut)[1])
col.ind[c(286, 411, 412)] = 'blue'
col.ind[c(49, 819, 866)] = 'darkgreen'

plot(PCs[,1],PCs[,2],pch=16, xlab="PC1",ylab="PC2", main="Nutritional Data",
     col = col.ind, xlim = c(-8.5,max(PCs[,1])))
text(c(PCs[286,1],PCs[411,1],PCs[412,1]),c(PCs[286,2],PCs[411,2],PCs[412,2]),
     labels = as.character(c(286, 411, 412)), pos = c(1,4,2), col = 'blue')
text(c(PCs[49,1],PCs[819,1],PCs[866,1]),c(PCs[49,2],PCs[819,2],PCs[866,2]),
     labels = as.character(c(49, 819, 866)), pos = c(4,4,4), col="darkgreen")
```



In blue (main) outliers of PC1 and in green (main) outliers of PC2.

```
col.ind = rep('black', dim(nut)[1])
col.ind[819] = 'darkgreen'
col.ind[84] = 'orange'
col.ind[c(49, 866)] = 'cyan'

plot(PCs[,2],PCs[,3],pch=16, xlab="PC2",ylab="PC3", main="Nutritional Data",
     col = col.ind, xlim = c(-11, max(PCs[,2])),
     ylim = c(min(PCs[,3]), 7))
text(PCs[819,2],PCs[819,3], labels = as.character(819), pos = 4, col = 'darkgreen')
text(PCs[84,2],PCs[84,3], labels = as.character(84), pos = 4, col="orange")
text(c(PCs[49,2],PCs[866,2]),c(PCs[49,3],PCs[866,3]),
     labels = as.character(c(49, 866)), pos = c(4,4), col="cyan")
```



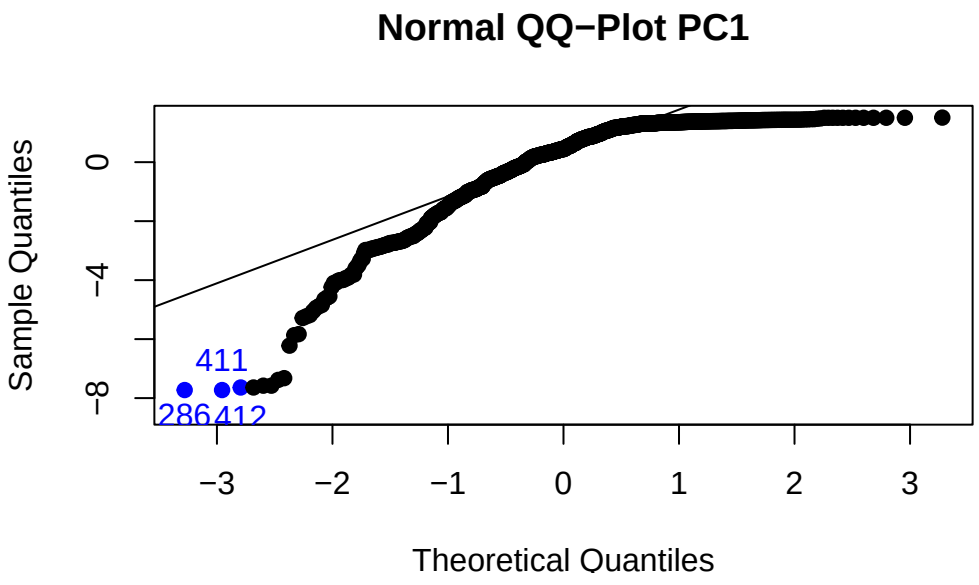
In cyan (main) outliers common to both PC2 and PC3 and in orange (main) outliers of PC3.

Clearly, all the observations found via the boxplots differ from the main pattern of the data.

Both outliers of PC1 and PC2 present very low scores of the components. The outliers of PC1 are very high energy foods (high in fat and calories). The outliers of PC2 can been seen as not cholesterol-friendly foods (not healthy foods; have low carbs and high cholesterol). Finally, outliers of PC3 are richer in proteins.

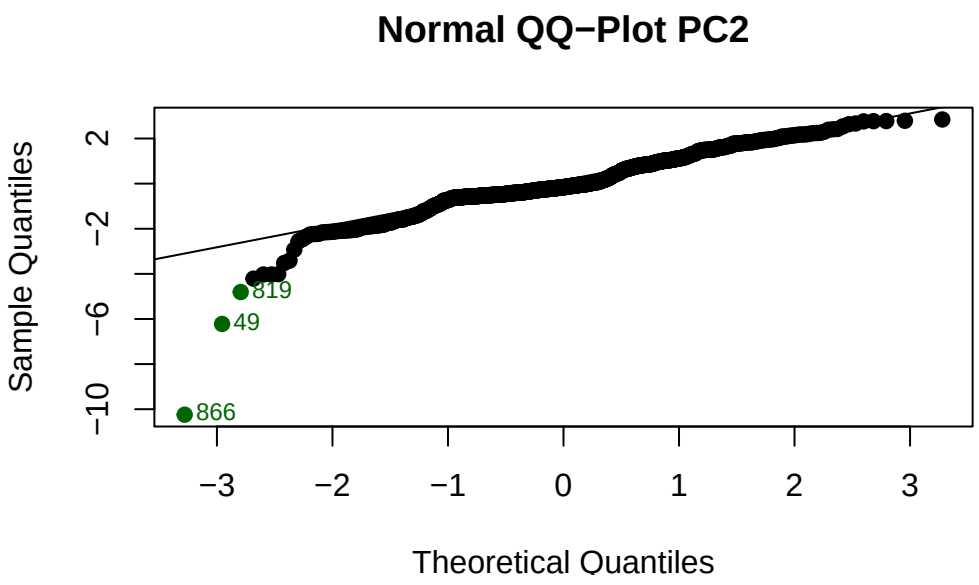
```
col.ind = rep('black', dim(nut)[1])
col.ind[c(286, 411, 412)] = 'blue'
out.PC1 = c(286, 411, 412)

qqnorm(PCs[,1],main = "Normal QQ-Plot PC1", pch =19, col = col.ind,
```

```
       ylim = c(-8.5, max(PCs[,1])))
qqline(PCs[,1])

text(head(qnorm(ppoints(dim(nut)[1])),3), PCs[c(286, 411, 412),1],
     labels = as.character(c(286, 411, 412)), pos = c(1,3,1), col = 'blue')
```

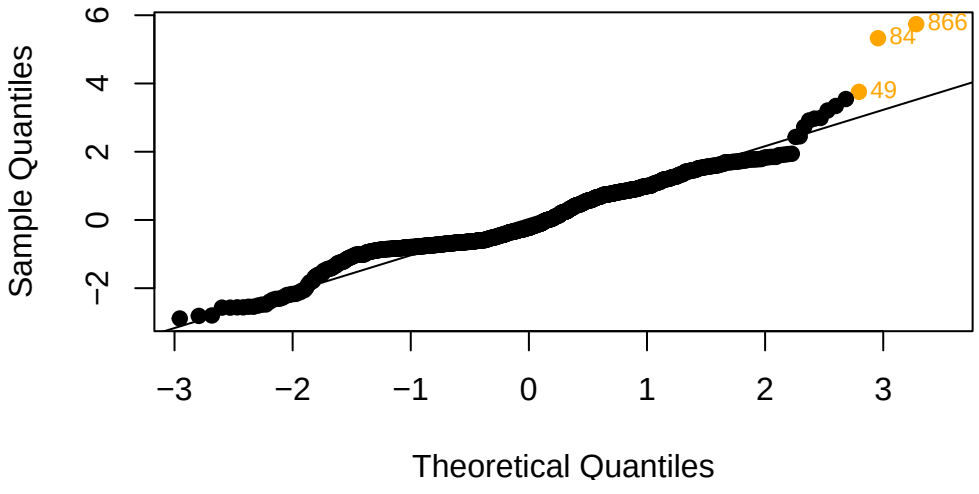


Finally, we can take a quick look to the normal QQ-Plots and see that these observations do not respect the normality assumption. Moreover, the entire first component does not seem to meet the normality assumption.

```
col.ind = rep('black', dim(nut)[1])
col.ind[c(49, 819, 866)] = 'darkgreen'
out.PC2 = c(49, 819, 866)

qqnorm(PCs[,2],main = "Normal QQ-Plot PC2", pch =19, col = col.ind)
qqline(PCs[,2])

text(head(qnorm(ppoints(dim(nut)[1])),3), PCs[c(866, 49, 819),2],
     labels = as.character(c(866, 49, 819)), pos = 4, col = 'darkgreen',
     offset = 0.3, cex = 0.75)
```

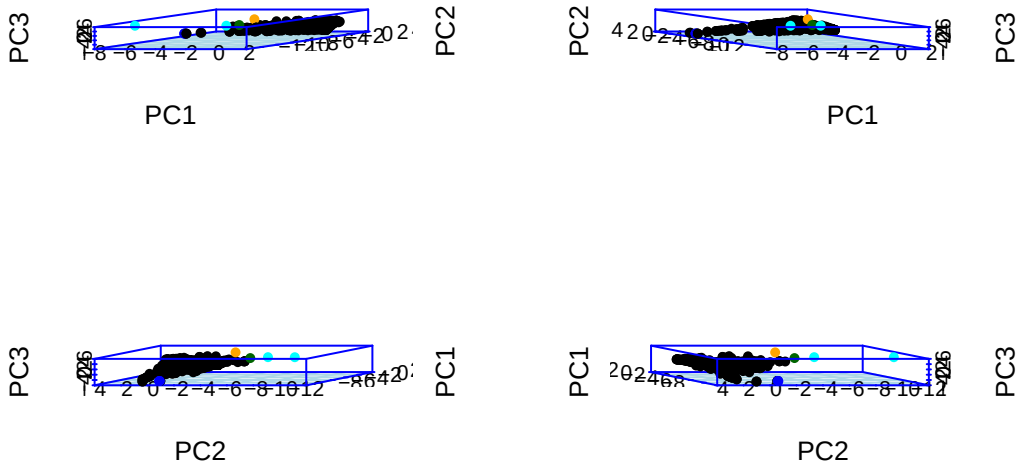



```
col.ind = rep('black', dim(nut)[1])
col.ind[c(49, 84, 866)] = 'orange'
out.PC3 = c(49, 84, 866)

qqnorm(PCs[,3],main = "Normal QQ-Plot PC3", pch =19, col = col.ind,
       xlim = c(min(PCs[,3]), 3.5))
qqline(PCs[,3])

text(tail(qnorm(ppoints(dim(nut)[1])),3), PCs[c(49, 84, 866),3],
     labels = as.character(c(49, 84, 866)), pos = 4, col = 'orange',
     offset = 0.3, cex = 0.75)
```



3D Scatter Plot

```
col.ind = rep('black', dim(nut)[1])
col.ind[c(286, 411, 412)] = 'blue'
col.ind[819] = 'darkgreen'
col.ind[84] = 'orange'
col.ind[c(49, 866)] = 'cyan'

library(scatterplot3d)
par(mfrow=c(2,2))
for (j in c(45, 135, 225, 315)){
  scatterplot3d(PCs[,1:3], color = col.ind, pch=16, angle = j,
                col.axis="blue", col.grid="lightblue",type="p",)
}
```



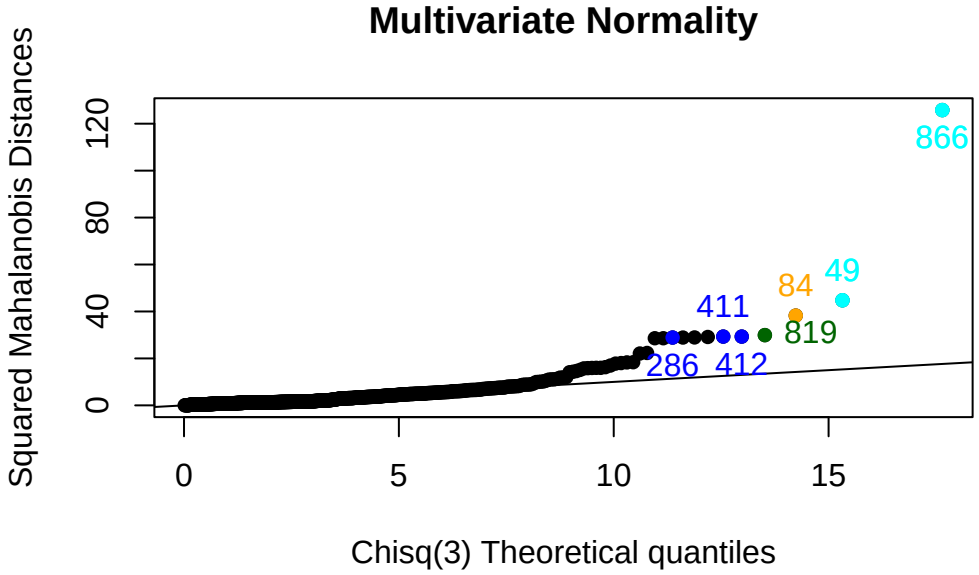

```
#for a "moving" 3D plot
library(rgl)
plot3d(PCs[,1], PCs[,2], PCs[,3], type="s", main="3D Plot", xlab="PC1",
       ylab="PC2",zlab="PC3",size=1, col = col.ind)
```

Even in three dimensions, these observations differ from the main pattern and look like "three-variate" outliers.

Multivariate Normality

```
X = PCs[,1:3]
d = mahalanobis(X, center = colMeans(X), cov = cov(X))
```

```r
qchi = qchisq(ppoints(d), df = retained)

qqplot(qchi, d, pch = 16, main = "Multivariate Normality",
       xlab = 'Chisq(3) Theoretical quantiles', ylab = 'Squared Mahalanobis Distances')
abline(0,1)

out.qq = match(c(out.PC1, out.PC2, out.PC3), order(d))

points(qchi[out.qq], sort(d)[out.qq], pch = 16,
       col = c(rep('blue', 3), 'cyan', 'darkgreen', rep('cyan',2), 'orange', 'cyan'))

pos = c(1, 3, 1, 3, 4, 1, 3, 3, 1)

text(qchi[out.qq], sort(d)[out.qq],
     col = c(rep('blue', 3), 'cyan', 'darkgreen', rep('cyan',2), 'orange', 'cyan'),
     labels = c(out.PC1, out.PC2, out.PC3), pos = pos)
```

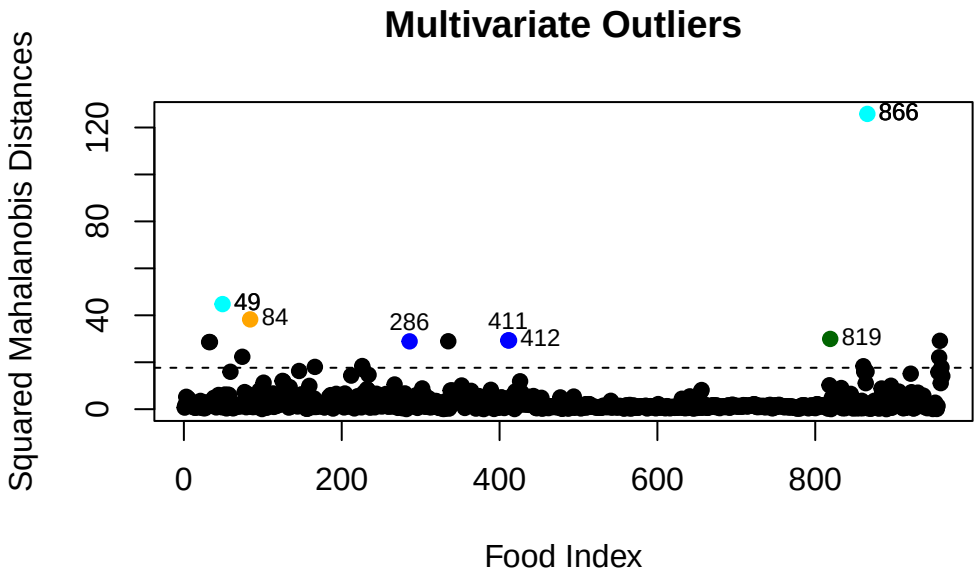


The entire bulk of the distribution and the left tail follow the quantiles of a chi squared distribution with 3 dof. The right tail is however too fat and seems to not meet the multivariate normality assumption. We can see from the plot that all the outliers found previously are exactly in this tail. They were not meeting the univariate normality assumption, as shown in point 4, and neither they meet the assumption of multivariate normality here.

Multivariate Outliers

```r
X = PCs[,1:3]
d = mahalanobis(X, center = colMeans(X), cov = cov(X))
```

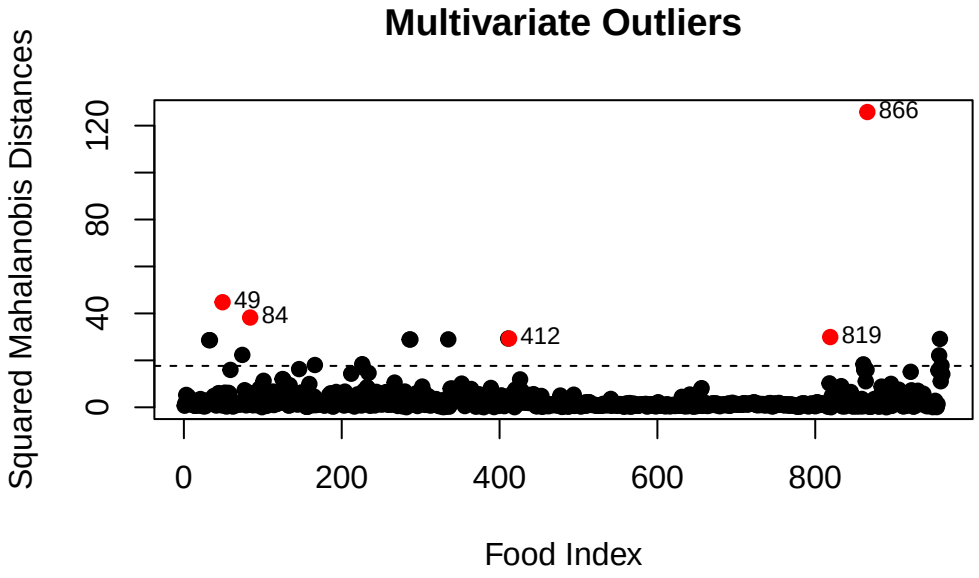
```
n = dim(nut)[1]

mycol = rgb(0,0,0,alpha = 0)
col.ind = rep('black', dim(nut)[1])
col.ind[c(286, 411, 412)] = 'blue'
col.ind[819] = 'darkgreen'
col.ind[84] = 'orange'
col.ind[c(49, 866)] = 'cyan'
col.ind[287] = mycol

plot(d,pch=19, col = col.ind, main = "Multivariate Outliers",
     ylab = "Squared Mahalanobis Distances", xlab = "Food Index")
abline(h=qchisq((n-0.5)/n,df= retained),lty=2)

text(order(d)[n], sort(d)[n], labels = names(sort(d)[n]),  pos = 4,
     cex = 0.75, offset = 0.3)

pos = c(3, 3, 4, 4, 4, 4, 4, 4, 4)

text(c(out.PC1, out.PC2, out.PC3), d[c(out.PC1, out.PC2, out.PC3)],
     labels = c(out.PC1, out.PC2, out.PC3),  pos = pos, cex = 0.75, offset = 0.3)
```



We can see that the squared Mahalanobis distances above the (n-0.5)/n quantile of the chi squared distribution with p = 3 dof are several and among those we can find the univariate (and three-variate) outliers found in the previous points.

We can pick the 5 most extreme Mahalanobis distances (red coloured in the graph below)

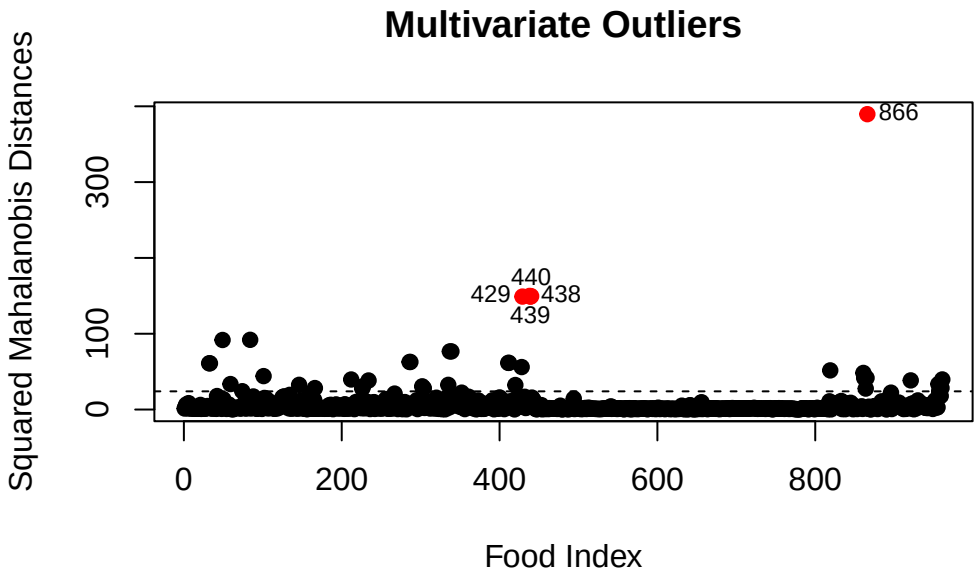
```
n = dim(nut)[1]

out.mult = tail(sort(d),5)
index = names(out.mult)
col.ind = rep('black', n)
col.ind[as.integer(index)] = 'red'

plot(d,pch=19, col = col.ind, main = "Multivariate Outliers",
     ylab = "Squared Mahalanobis Distances", xlab = "Food Index")
abline(h=qchisq((n-0.5)/n,df= retained),lty=2)

text(as.integer(index), out.mult, labels = index,  pos = 4, cex = 0.75, offset = 0.3)
```



We can then take the squared Mahalobis distances of the six original variables and check for the same quantile (n-0.5)/n but for a chi squared distribution with 6 dof. In red, we can see the 5 most extreme values.

```
d = mahalanobis(nut, center = colMeans(nut), cov = cov(nut))

out.mult = tail(sort(d),5)
index = names(out.mult)
col.ind = rep('black', n)
col.ind[as.integer(index)] = 'red'

plot(d,pch=19, col = col.ind, main = "Multivariate Outliers",
     ylab = "Squared Mahalanobis Distances", xlab = "Food Index")
abline(h=qchisq((n-0.5)/n, df = dim(nut)[2]),lty=2)

pos = c(1,2,3,4,4)
```

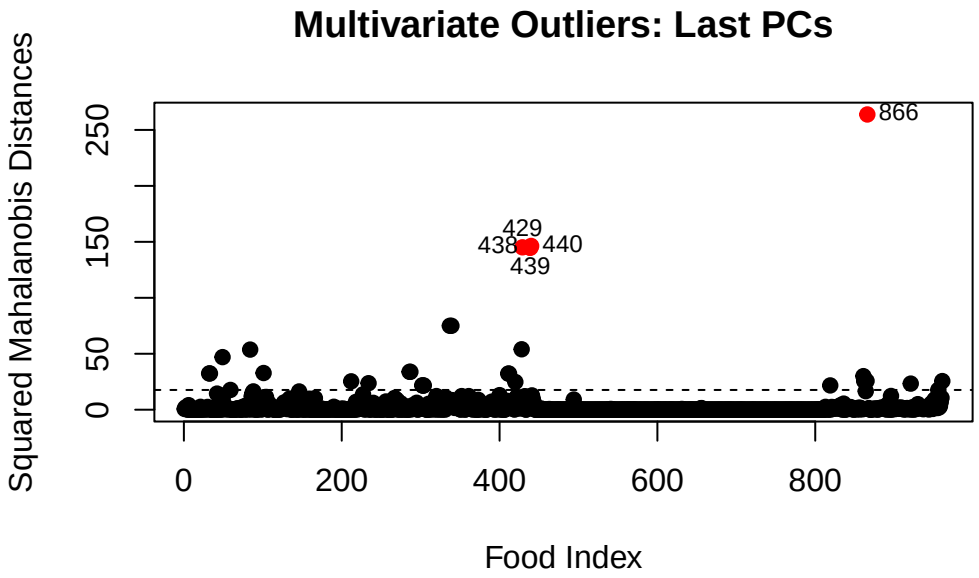
```
text(as.integer(index), out.mult, labels = index,  pos = pos, cex = 0.75, offset = 0.3)
```



Except for obs number 866, the multivariate outliers found with the first three PCs do not coincide with the multivariate outliers found with the six original variables.

Lastly, we plot the squared Mahalanobis distances of the last PCs.

```
X = PCs[,4:6]
d = mahalanobis(X, center = colMeans(X), cov = cov(X))

out.mult = tail(sort(d),5)
index = names(out.mult)
col.ind = rep('black', n)
col.ind[as.integer(index)] = 'red'

plot(d,pch=19, col = col.ind, main = "Multivariate Outliers: Last PCs",
     ylab = "Squared Mahalanobis Distances", xlab = "Food Index")
abline(h=qchisq((n-0.5)/n,df= retained),lty=2)

pos = c(1,2,3,4,4)

text(as.integer(index), out.mult, labels = index,  pos = pos, cex = 0.75, offset = 0.3)
```



The multivariate outliers found via the last PCs are the same as the multivariate outliers found via the six original variables and they differ from the multivariate outliers found via the first PCs. This might indicate that the outliers from the original variables are not adequately represented by the first PCs. Hence, it might be the case that the last PCs are artificially created to account for the variation of the outliers (artificial dimensions).