Data screening

KEY

Fri Sep 23 16:47:38 2022

## Screening data for use in multivariate analysis

First, import the shells data.

library(readxl)  
read\_excel("multivariate\_data\_f22.xlsx","shells") -> shells

Check if the file has complete data (no missing cells):

table(complete.cases(shells))

##   
## TRUE   
## 120

Make a vector that has the names of the numeric variables in it - this will make it easier to refer to them as a group below:

shell.vars <- colnames(shells[,-c(1:2)])

## Initial graphical screening

We will need to make boxplots and normal probability plots for each variable to check for outliers and distributional issues.

We will want to use residuals for normality plots to get rid of differences between group means. We will get those by fitting a MANOVA first, and then extracting residuals from the fitted model.

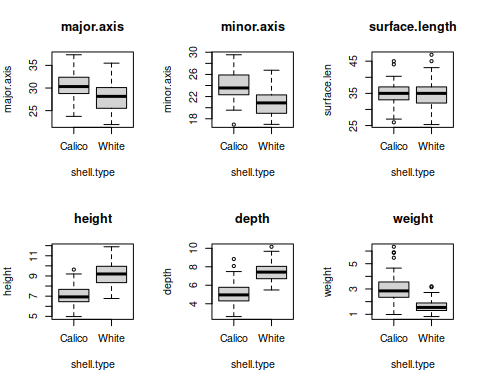
manova(as.matrix(shells[shell.vars]) ~ shell.type, data = shells) -> shells.manova  
data.frame(residuals(shells.manova)) -> shells.resid

We will make boxplots and normal probability plots in a single multiple-panel graph, so that we can scan the graphs more easily. Set up the graphical parameters (par()) for 2 rows of graphs with 5 columns, and then plot the boxplots, followed by the normal probability plots.

library(car)

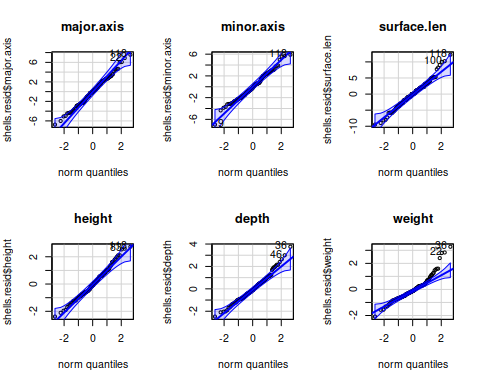
## Loading required package: carData

par(mfrow=c(2,3))  
boxplot(major.axis ~ shell.type, data = shells, main = "major.axis")  
boxplot(minor.axis ~ shell.type, data = shells, main = "minor.axis")  
boxplot(surface.len ~ shell.type, data = shells, main = "surface.length")  
boxplot(height ~ shell.type, data = shells, main = "height")  
boxplot(depth ~ shell.type, data = shells, main = "depth")  
boxplot(weight ~ shell.type, data = shells, main = "weight")



Normal probability plots:

library(car)  
par(mfrow=c(2,3))  
qqPlot(shells.resid$major.axis, distribution = "norm", main = "major.axis")  
qqPlot(shells.resid$minor.axis, distribution = "norm", main = "minor.axis")  
qqPlot(shells.resid$surface.len, distribution = "norm", main = "surface.len")  
qqPlot(shells.resid$height, distribution = "norm", main = "height")  
qqPlot(shells.resid$depth, distribution = "norm", main = "depth")  
qqPlot(shells.resid$weight, distribution = "norm", main = "weight")



**Question: how can you detect a skewed deistribution in a boxplot?**

If the median is not in the middle of the box, or the whiskers are different lengths the data are skewed.

**Question: how do you know from a normal probability plot if the variable is normally distributed?**

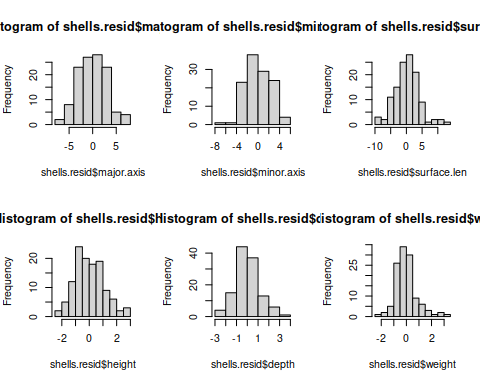
The points follow the diagonal line when the data are normally distributed.

**Question: which variable is the most right-skewed, according to your boxplots and normal probability plots?**

Weight is right skewed.

We can check the distribution of the residuals using histograms as well:

par(mfrow = c(2,3))  
hist(shells.resid$major.axis)  
hist(shells.resid$minor.axis)  
hist(shells.resid$surface.len)  
hist(shells.resid$height)  
hist(shells.resid$depth)  
hist(shells.resid$weight)



**Question: is the same variable you thought was right-skewed based on boxplot and normal probability plot also right skewed as a histogram? How can you tell skew from a histogram?**

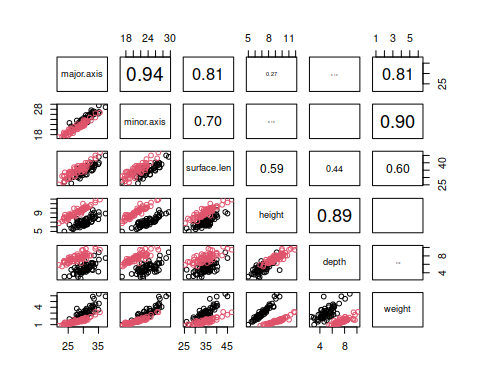
Yes, weight is still right skewed.

Next we can make a scatterplot matrix that puts scatterplots in the lower triangle, and correlations in the upper triangle. Run this code chunk to add the panel.cor() function into your environment:

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)  
{  
 par(usr = c(0, 1, 0, 1))  
 r <- abs(cor(x, y))  
 txt <- format(c(r, 0.123456789), digits = digits)[1]  
 txt <- paste0(prefix, txt)  
 if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)  
 text(0.5, 0.5, txt, cex = cex.cor \* r)  
}

Next we can make the scatterplot matrix using the pairs() function:

pairs(shells[shell.vars], upper.panel = panel.cor, col = factor(shells$shell.type))



**Question: which variable is involved in the plots that show non-linear relationships?**

Weight has non-linear relationships with other variables.

Weight seems to be right-skewed and non-linearly related to other variables, so let’s try log-transforming it:

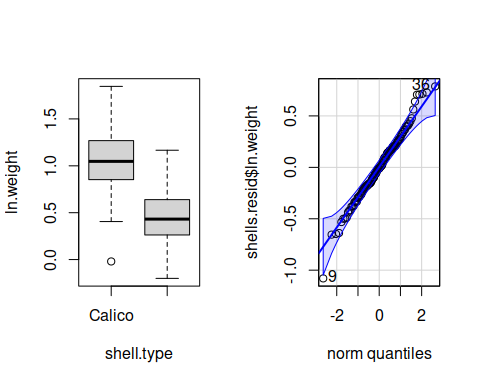
shells$ln.weight <- log(shells$weight)

Add the residuals for ln.weight to shells.resid:

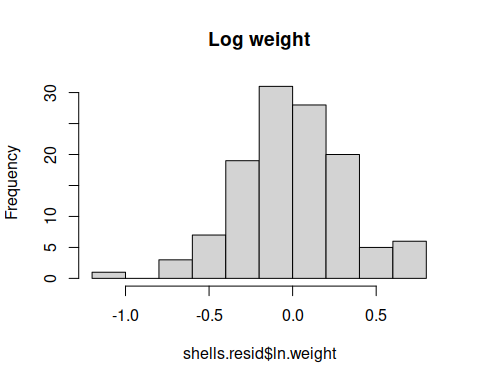
shells.resid$ln.weight <- residuals(aov(ln.weight ~ shell.type, data = shells))

Now make a boxplot and qqPlot for ln.weight.

par(mfrow = c(1,2))  
boxplot(ln.weight ~ shell.type, data = shells)  
qqPlot(shells.resid$ln.weight, dist = "norm")

 Make a new histogram for ln.weight:

hist(shells.resid$ln.weight, main = "Log weight")



**Question: did log-transforming weight help with the distribution?**

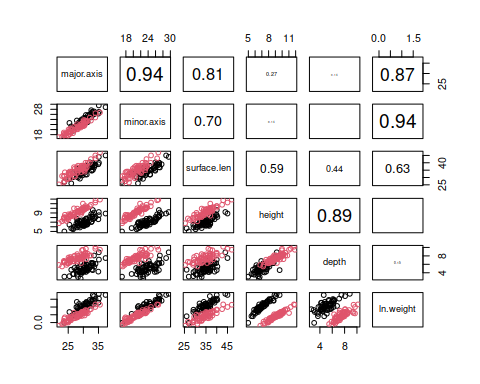
Yes, weight now looks bell-shaped.

Make a list of the variables you will use in the pairs() plot:

shell.vars[6] <- "ln.weight"

Plot the scatterplot matrix using log.weight instead of weight.

pairs(shells[shell.vars], col = factor(shells$shell.type), upper.panel = panel.cor)



**Question: did the transformation straighten out the curved relationships with weight? Did it introduce any new issues that weren’t there before?**

Yes, the relationships now look linear.

## Multivariate distribution check

Now we will get the residuals from a MANOVA of the shells data, and check if their Mahalanobis distances follow the Chi-square distribution, like we expect them to.

Calculate the means of the residuals (these should all be 0 or very close to it):

colMeans(shells.resid[shell.vars]) -> shell.means

The covariance matrix of the residuals next:

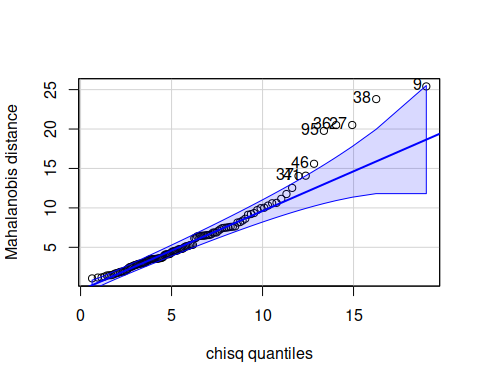
var(shells.resid[shell.vars]) -> shell.cov

Mahalanobis distances are calcuated from the residuals, the vector of means, and the covariance matrix:

mahalanobis(shells.resid[shell.vars], shell.means, shell.cov) -> shells$mahal

Compare the Mahalanobis distances to the Chi-square distribution:

qqPlot(shells$mahal, distribution = "chisq", df = 6, id = list(n = 8), ylab = "Mahalanobis distance")



**Question: based on the graph, are there multivariate outliers?**

Yes, there are still multivariate outliers, because there are points outside of the confidence bands.

## Teeth

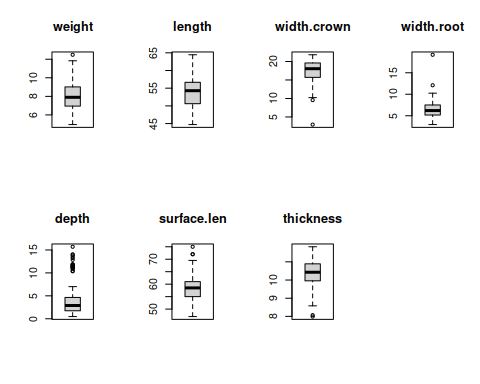
Repeat the steps on the buffalo teeth.

Import the data:

data.frame(read\_excel("multivariate\_data\_f22.xlsx","teeth")) -> teeth

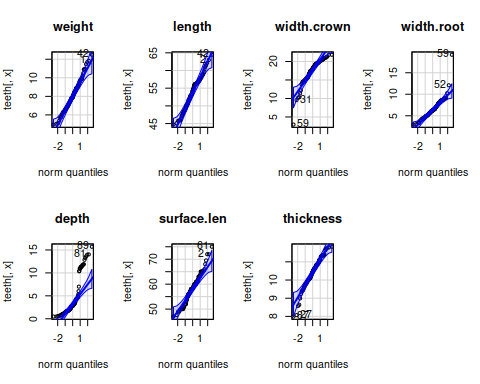
Plot boxplots and normal probability plots - note that there is no grouping variable for the teeth, so you don’t need to split the variables by groups for the boxplots this time. Put the variable names into an object called teeth.names, and then use lapply() to make the plots:

c("weight", "length", "width.crown", "width.root", "depth", "surface.len", "thickness") -> teeth.names  
  
par(mfrow = c(2,4))  
lapply(teeth.names, FUN = function(x) boxplot(teeth[,x], main = x))



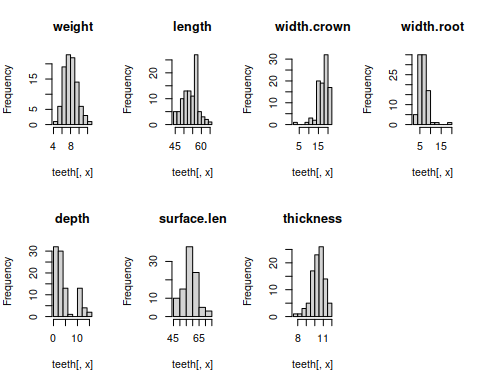
Now use lapply() to get normal probability plots:

par(mfrow = c(2,4))  
lapply(teeth.names, FUN = function(x) qqPlot(teeth[,x], distribution = "norm", main = x))



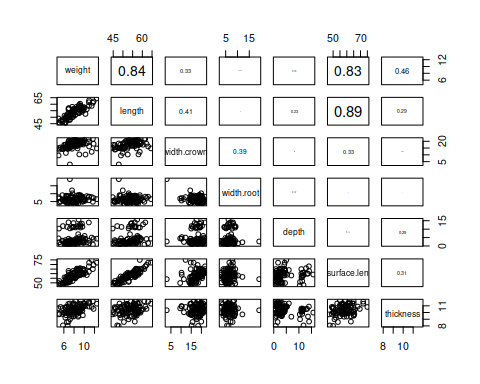
Now use an lapply() to get histograms:

par(mfrow = c(2,4))  
lapply(teeth.names, FUN = function(x) hist(teeth[,x], main = x))



Scatterplot matrix (no need to color by groups this time):

pairs(teeth[teeth.names], upper.panel = panel.cor)



### Log transformation - only if needed, skip if not needed

If there are any right-skewed variables log-transform them:

Re-do the boxplots and normal probability plots using the variables you log-transformed in place of their un-transformed versions:

Re-do the scatterplot matrix using the log-transformed variables:

### Mahalanobis distances for teeth data

Calculate Mahalanobis distances - since there are no groups in the teeth data you just need to use colMeans on the variables, and var() to get the covariance matrix for the variables (no need to do a MANOVA and extract residuals):

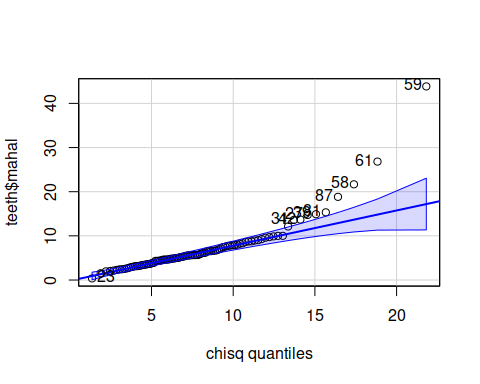
colMeans(teeth[teeth.names]) -> teeth.means  
var(teeth[teeth.names]) -> teeth.covar

Calculate the Mahalanobis distances and add them as a variable in your teeth data:

mahalanobis(teeth[teeth.names], teeth.means, teeth.covar) -> teeth$mahal

Do a qqPlot on the Mahalanobis distances (remember, df is equal to the number of variables):

qqPlot(teeth$mahal, df = 8, dist = "chisq", id = list(n = 10))



## [1] 59 61 58 87 81 39 27 42 31 23

**Question: are there multivariate outliers? Could you see them in the other plots, or was the Mahalanobis distance needed to identify them?**

Yes, several.