hw\_2\_q2.R

Nadim

Wed Jan 23 17:46:37 2019

library(HSAUR3)

## Warning: package 'HSAUR3' was built under R version 3.5.2

## Loading required package: tools

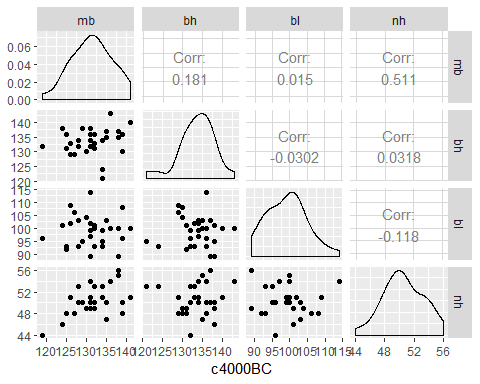
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.2

library(GGally)

## Warning: package 'GGally' was built under R version 3.5.2

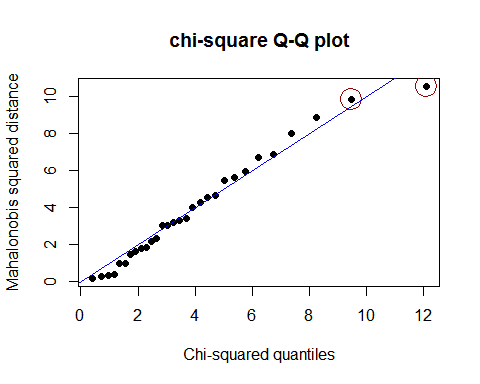
epochs<-skulls$epoch  
skull\_cov<- skulls[epochs=="c4000BC",2:5]  
ggpairs(skull\_cov,xlab="c4000BC")



chisquare.plot <- function(x,mark){  
 p <- ncol(x)  
 n<-nrow(x)  
 xbar<- colMeans(x)  
 s <- cov(x)  
   
 ###Mahalonobis Dist  
   
 x.cent <- scale(x,center =T, scale = F)  
 d2<- diag(x.cent%\*%solve(s)%\*%t(x.cent))  
   
 qchi <- qchisq((1:n-0.5)/n, df =p)  
 sortd <- sort(d2)  
   
 plot(qchi,sortd,pch=19,xlab="Chi-squared quantiles",ylab="Mahalonobis squared distance",main="chi-square Q-Q plot")  
   
 points(qchi[(n-mark+1):n],sortd[(n-mark+1):n],cex=3,col="#990000")  
   
 return((sortd[(n-mark+1):n]))  
}  
  
chisquare.plot(x=skull\_cov,mark = 2)

## 12 29   
## 9.861683 10.573099

abline(0,1,col="blue")



# epochs<-skulls$epoch  
# skull\_cov<- skulls[epochs=="c4000BC",2:5]  
  
z\_col<- list()  
#z\_col<-matrix(list(),nrow=length(skull\_cov$mb),ncol=4)  
for (i in 1:4){  
 z\_col[i]<- list((scale(skull\_cov[i])))  
 skull\_cov[i+4]<- z\_col[i]  
 #skull\_cov<-cbind(z\_col[-i],skull\_cov)  
  
}  
  
colnames(skull\_cov)<-c("mb","bh","bl","nh","Z\_mb","z\_bh","z\_bl","z\_nh")  
skull\_cov

## mb bh bl nh mb bh bl nh  
## 1 131 138 89 49 -0.07148545 0.98454914 -1.72772526 -0.5548574  
## 2 125 131 92 48 -1.24124734 -0.58177904 -1.21790469 -0.9167209  
## 3 131 132 99 50 -0.07148545 -0.35801787 -0.02832336 -0.1929939  
## 4 119 132 96 44 -2.41100922 -0.35801787 -0.53814393 -2.3641751  
## 5 136 143 100 54 0.90331612 2.10335499 0.14161682 1.2544602  
## 6 138 137 89 56 1.29323675 0.76078797 -1.72772526 1.9781873  
## 7 139 130 108 48 1.48819707 -0.80554021 1.50113834 -0.9167209  
## 8 125 136 93 48 -1.24124734 0.53702681 -1.04796450 -0.9167209  
## 9 131 134 102 51 -0.07148545 0.08950447 0.48149720 0.1688696  
## 10 134 134 99 51 0.51339549 0.08950447 -0.02832336 0.1688696  
## 11 129 138 95 50 -0.46140608 0.98454914 -0.70808412 -0.1929939  
## 12 134 121 95 53 0.51339549 -2.81939073 -0.70808412 0.8925967  
## 13 126 129 109 51 -1.04628702 -1.02930138 1.67107853 0.1688696  
## 14 132 136 100 50 0.12347487 0.53702681 0.14161682 -0.1929939  
## 15 141 140 100 51 1.87811770 1.43207148 0.14161682 0.1688696  
## 16 131 134 97 54 -0.07148545 0.08950447 -0.36820374 1.2544602  
## 17 135 137 103 50 0.70835581 0.76078797 0.65143739 -0.1929939  
## 18 132 133 93 53 0.12347487 -0.13425670 -1.04796450 0.8925967  
## 19 139 136 96 50 1.48819707 0.53702681 -0.53814393 -0.1929939  
## 20 132 131 101 49 0.12347487 -0.58177904 0.31155701 -0.5548574  
## 21 126 133 102 51 -1.04628702 -0.13425670 0.48149720 0.1688696  
## 22 135 135 103 47 0.70835581 0.31326564 0.65143739 -1.2785845  
## 23 134 124 93 53 0.51339549 -2.14810722 -1.04796450 0.8925967  
## 24 128 134 103 50 -0.65636639 0.08950447 0.65143739 -0.1929939  
## 25 130 130 104 49 -0.26644576 -0.80554021 0.82137758 -0.5548574  
## 26 138 135 100 55 1.29323675 0.31326564 0.14161682 1.6163238  
## 27 128 132 93 53 -0.65636639 -0.35801787 -1.04796450 0.8925967  
## 28 127 129 106 48 -0.85132671 -1.02930138 1.16125796 -0.9167209  
## 29 131 136 114 54 -0.07148545 0.53702681 2.52077948 1.2544602  
## 30 124 138 101 46 -1.43620765 0.98454914 0.31155701 -1.6404480

sk<-skull\_cov[,1:4]  
s<-cov(sk)  
x.cent <- scale(sk,center =T, scale = F)  
d2<- diag(x.cent%\*%solve(s)%\*%t(x.cent))  
skull\_cov<-cbind(d2=d2,skull\_cov)  
  
apply(skull\_cov[,2:5], 2, shapiro.test)

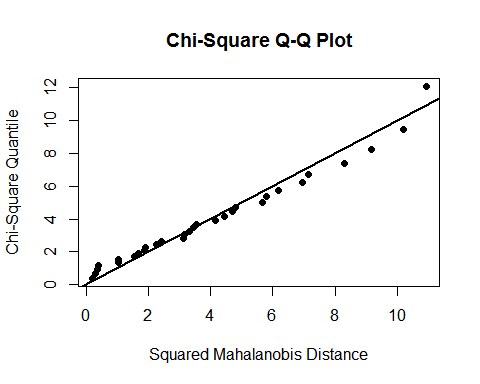
## $mb  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.98136, p-value = 0.8603  
##   
##   
## $bh  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.95664, p-value = 0.2536  
##   
##   
## $bl  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.97314, p-value = 0.6282  
##   
##   
## $nh  
##   
## Shapiro-Wilk normality test  
##   
## data: newX[, i]  
## W = 0.97481, p-value = 0.6772

library(MVN)

## Warning: package 'MVN' was built under R version 3.5.2

## sROC 0.1-2 loaded

mvn(skull\_cov[, 2:5], mvnTest = "royston", multivariatePlot = "qq")



## $multivariateNormality  
## Test H p value MVN  
## 1 Royston 2.752767 0.603866 YES  
##   
## $univariateNormality  
## Test Variable Statistic p value Normality  
## 1 Shapiro-Wilk mb 0.9814 0.8603 YES   
## 2 Shapiro-Wilk bh 0.9566 0.2536 YES   
## 3 Shapiro-Wilk bl 0.9731 0.6282 YES   
## 4 Shapiro-Wilk nh 0.9748 0.6772 YES   
##   
## $Descriptives  
## n Mean Std.Dev Median Min Max 25th 75th Skew  
## mb 30 131.36667 5.129249 131 119 141 128.00 134.75 -0.16642216  
## bh 30 133.60000 4.469051 134 121 143 131.25 136.00 -0.64720446  
## bl 30 99.16667 5.884423 100 89 114 95.00 102.75 0.31717217  
## nh 30 50.53333 2.763473 50 44 56 49.00 53.00 -0.08670975  
## Kurtosis  
## mb -0.4879548  
## bh 0.8488047  
## bl -0.2756768  
## nh -0.4538837

skull\_cov

## d2 mb bh bl nh mb bh bl  
## 1 4.5320852 131 138 89 49 -0.07148545 0.98454914 -1.72772526  
## 2 3.4358752 125 131 92 48 -1.24124734 -0.58177904 -1.21790469  
## 3 0.1764118 131 132 99 50 -0.07148545 -0.35801787 -0.02832336  
## 4 8.0418366 119 132 96 44 -2.41100922 -0.35801787 -0.53814393  
## 5 5.9719964 136 143 100 54 0.90331612 2.10335499 0.14161682  
## 6 6.7271613 138 137 89 56 1.29323675 0.76078797 -1.72772526  
## 7 8.8814765 139 130 108 48 1.48819707 -0.80554021 1.50113834  
## 8 3.3320641 125 136 93 48 -1.24124734 0.53702681 -1.04796450  
## 9 0.3570568 131 134 102 51 -0.07148545 0.08950447 0.48149720  
## 10 0.2783793 134 134 99 51 0.51339549 0.08950447 -0.02832336  
## 11 1.8359202 129 138 95 50 -0.46140608 0.98454914 -0.70808412  
## 12 9.8616831 134 121 95 53 0.51339549 -2.81939073 -0.70808412  
## 13 5.6144356 126 129 109 51 -1.04628702 -1.02930138 1.67107853  
## 14 0.3715983 132 136 100 50 0.12347487 0.53702681 0.14161682  
## 15 5.4712627 141 140 100 51 1.87811770 1.43207148 0.14161682  
## 16 2.3261707 131 134 97 54 -0.07148545 0.08950447 -0.36820374  
## 17 1.6126762 135 137 103 50 0.70835581 0.76078797 0.65143739  
## 18 1.8077558 132 133 93 53 0.12347487 -0.13425670 -1.04796450  
## 19 4.0059824 139 136 96 50 1.48819707 0.53702681 -0.53814393  
## 20 1.0094819 132 131 101 49 0.12347487 -0.58177904 0.31155701  
## 21 2.1747998 126 133 102 51 -1.04628702 -0.13425670 0.48149720  
## 22 4.2840657 135 135 103 47 0.70835581 0.31326564 0.65143739  
## 23 6.9044891 134 124 93 53 0.51339549 -2.14810722 -1.04796450  
## 24 1.0040121 128 134 103 50 -0.65636639 0.08950447 0.65143739  
## 25 1.4884986 130 130 104 49 -0.26644576 -0.80554021 0.82137758  
## 26 3.0207550 138 135 100 55 1.29323675 0.31326564 0.14161682  
## 27 3.2070250 128 132 93 53 -0.65636639 -0.35801787 -1.04796450  
## 28 3.0395173 127 129 106 48 -0.85132671 -1.02930138 1.16125796  
## 29 10.5730986 131 136 114 54 -0.07148545 0.53702681 2.52077948  
## 30 4.6524286 124 138 101 46 -1.43620765 0.98454914 0.31155701  
## nh  
## 1 -0.5548574  
## 2 -0.9167209  
## 3 -0.1929939  
## 4 -2.3641751  
## 5 1.2544602  
## 6 1.9781873  
## 7 -0.9167209  
## 8 -0.9167209  
## 9 0.1688696  
## 10 0.1688696  
## 11 -0.1929939  
## 12 0.8925967  
## 13 0.1688696  
## 14 -0.1929939  
## 15 0.1688696  
## 16 1.2544602  
## 17 -0.1929939  
## 18 0.8925967  
## 19 -0.1929939  
## 20 -0.5548574  
## 21 0.1688696  
## 22 -1.2785845  
## 23 0.8925967  
## 24 -0.1929939  
## 25 -0.5548574  
## 26 1.6163238  
## 27 0.8925967  
## 28 -0.9167209  
## 29 1.2544602  
## 30 -1.6404480