MVNTest

Mark Randall

2024-06-05

MVN TEST

The following is a referential of test using the the following code from the url

https://cran.r-project.org/web/packages/MVN/vignettes/MVN.html#11_The_mvn_function | It demonstrates that mvn can be utilised.

```
data(iris)
setosa <- iris[1:50, 1:4]
mvnTest = "mardia"
str(iris)
```

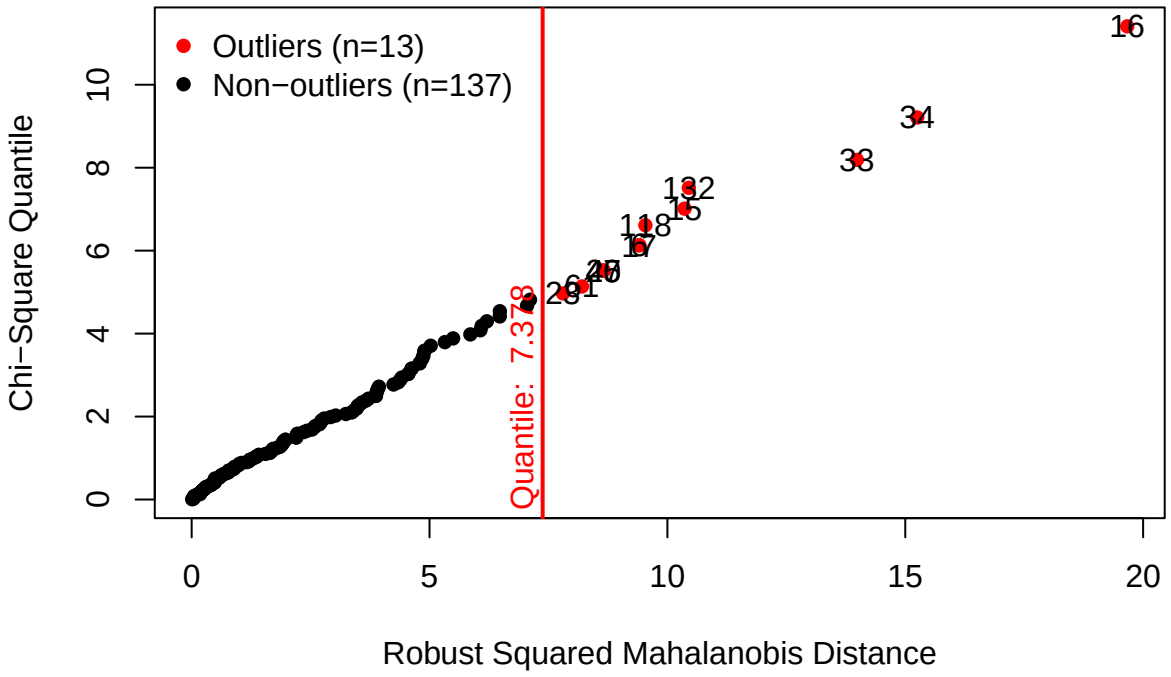
```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
result <- mvn(data = setosa, mvnTest = "mardia")
result$multivariateNormality
```

```
             Test        Statistic           p value Result
1 Mardia Skewness 25.6643445196298 0.177185884467652    YES
2 Mardia Kurtosis 1.29499223711605 0.195322907441935    YES
3             MVN             <NA>              <NA>    YES
```

Generating the plot as per course code. Note data frame has only 150 varaibles.

```
testIris <- iris[, 1:2]
mvnQQplot1 <- mvn(data = testIris, multivariateOutlierMethod = "quan", showOutliers = TRUE)
```

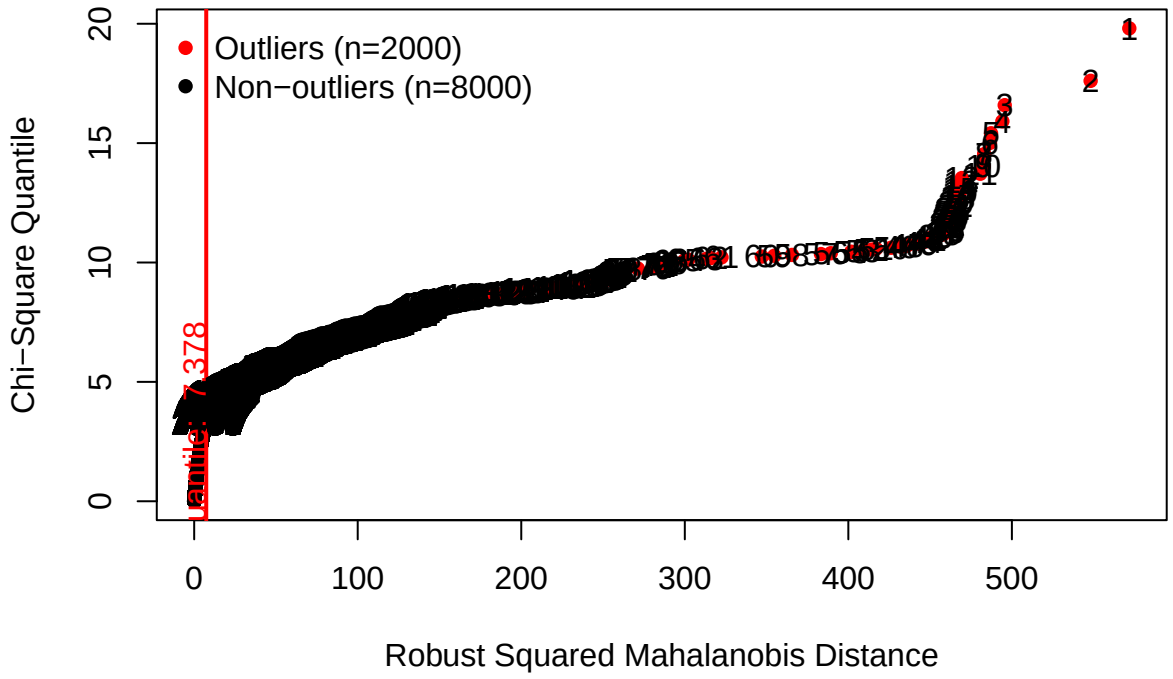


Larger Data Set Derived from Assessment.

Thew next test will be using data generated by the assignment.
Read in data and test at 10,000 incidents.

```
testDF <- readRDS("testDF")


testDF %<>% filter(!(is.na(Dist_GPO))) %>%
           select(Time_Float,Dist_GPO)
ss10000 <- testDF[1:10000,]
timeStartTest1 <- Sys.time()
mvnQQplot2 <- mvn(data = ss10000, multivariateOutlierMethod = "quan", showOutliers = TRUE)
```

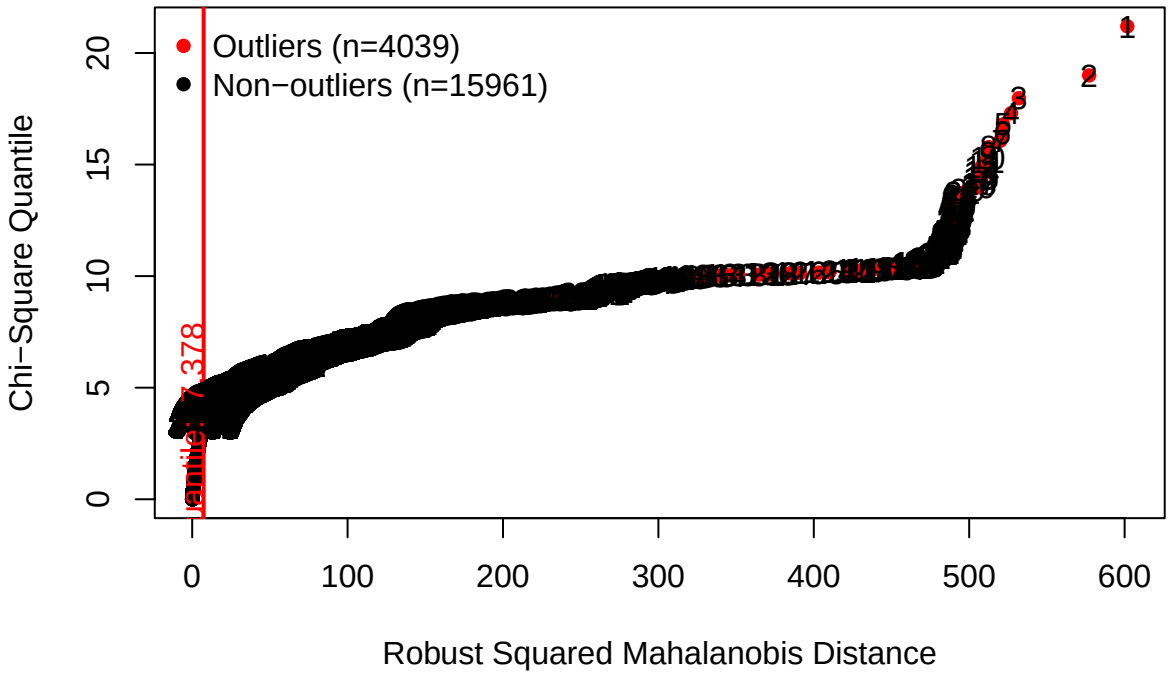



```
timeEndTest1 <- Sys.time()
timetaken1 <- difftime(timeEndTest1,timeStartTest1, units = "secs")
cat(glue('10,000 test secs: {timetaken1}'))
```

```
10,000 test secs: 10.9643378257751
```

Test for 20000 observations

```
ss20000 <- testDF[1:20000,]
timeStartTest2 <- Sys.time()
mvnQQplot3 <- mvn(data = ss20000, multivariateOutlierMethod = "quan", showOutliers = TRUE)
```




```
timeEndTest2 <- Sys.time()
timetaken2 <- difftime(timeEndTest2,timeStartTest2, units = "secs")
cat(glue('20,000 test secs: {timetaken2}'))
```

```
20,000 test secs: 41.2003490924835
```

Half data set at 83,000.

```
# ss83000 <- testDF[1:83000,]
#
# timeStartTest3 <- Sys.time()
# mvnQQplot4 <- tryCatch({
#   mvn(data = ss83000, multivariateOutlierMethod = "quan", showOutliers = TRUE)
# }, error = function(E) {
#    cat(glue("83,000 error: {conditionMessage(E)}"))
# }, warning = function(W){
#   cat(glue("83,000 warning: {conditionMessage(W)}"))
#
# })
# timeEndTest3 <- Sys.time()
# timetaken3 <- difftime(timeEndTest3,timeStartTest3, units = "secs")
# cat(glue('83,000 test secs: {timetaken3}'))
```

83,000 error: cannot allocate vector of size 51.3 Gb
83,000 test secs: 7187.8790512085

```
timeStartTest4 <- Sys.time()
mvnQQplot5 <- tryCatch({
  mvn(data = testDF, multivariateOutlierMethod = "quan", showOutliers = TRUE)
}, error = function(E) {
   cat(glue("165,156 error: {conditionMessage(E)}"))
}, warning = function(W){
  cat(glue("165,156 warning: {conditionMessage(W)}"))

})
```

```
165,156 error: cannot allocate vector of size 203.2 Gb
```

```
timeEndTest4 <- Sys.time()
timetaken4 <- difftime(timeEndTest4,timeStartTest4, units = "secs")
cat(glue('165,156 test secs: {timetaken4}'))
```

```
165,156 test secs: 0.156327962875366
```

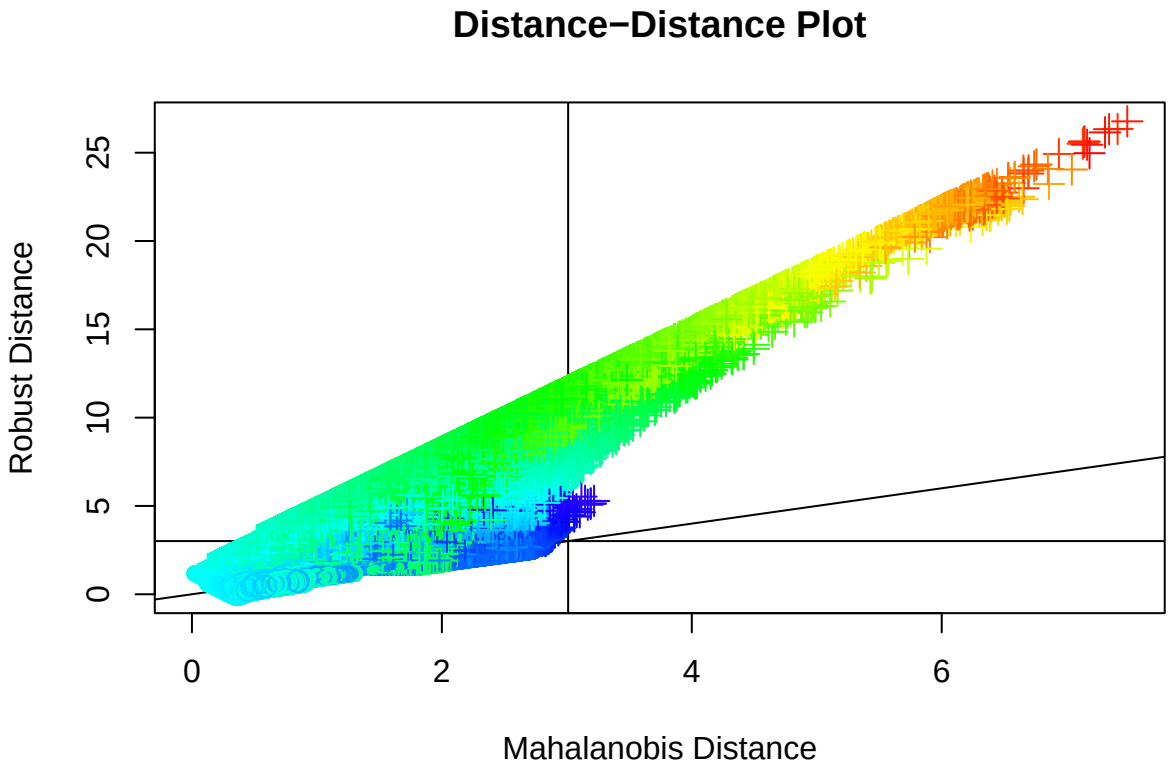
Test input against the mvoutlier package

These plots were taken from the mvoutlier package.

https://rdrr.io/cran/mvoutlier/

DDplot The function dd.plot plots the classical mahalanobis distance of the data against the robust mahalanobis distance based on the mcd estimator. Different symbols (see function symbol.plot) and colours (see function color.plot) are used depending on the mahalanobis and euclidean distance of the observations (see Filzmoser et al., 2005).

```
timeStartTest5 <- Sys.time()
testDD <- dd.plot(testDF,quan= 1/2, alpha = 0.025)
```

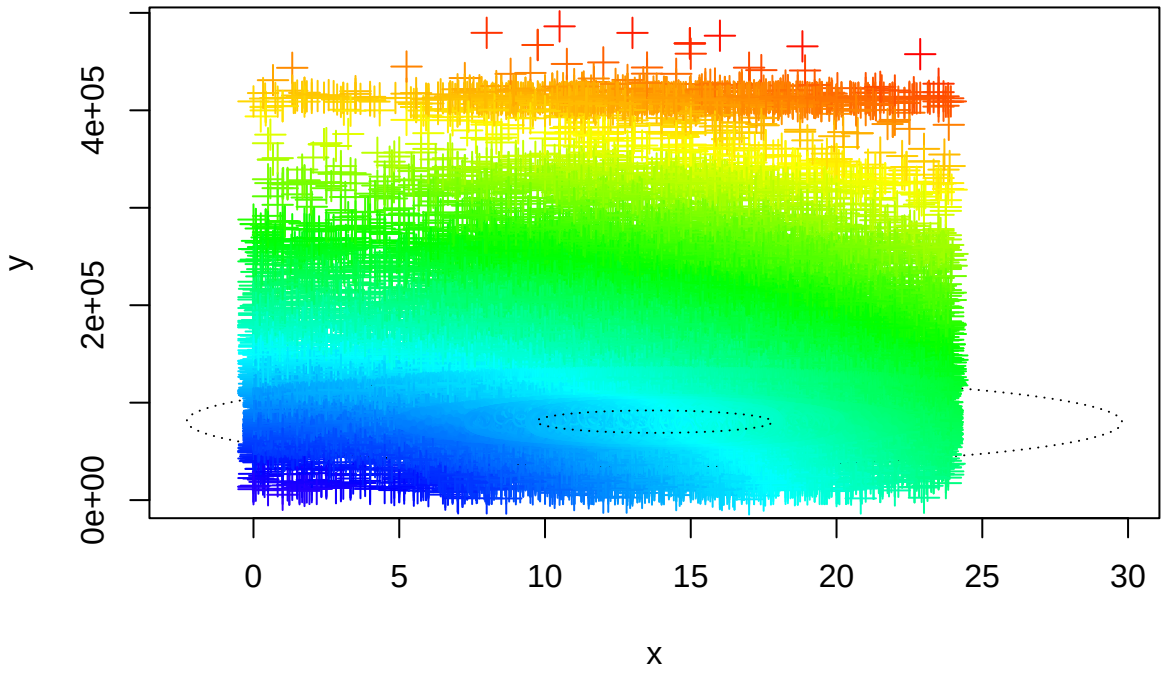



```
timeEndTest5 <- Sys.time()
timetaken5 <- difftime(timeEndTest5,timeStartTest5, units = "secs")
cat(glue('DDplot test secs: {timetaken5}'))
```

```
DDplot test secs: 0.754509925842285
```

Colorplot The function color.plot plots the (two-dimensional) data using different symbols according to the robust mahalanobis distance based on the mcd estimator with adjustment and using different colors according to the euclidean distances of the observations.

```
timeStartTest6 <- Sys.time()
testColorPlot <-color.plot(as.data.frame (testDF),quan= 1/2, alpha = 0.025)
```

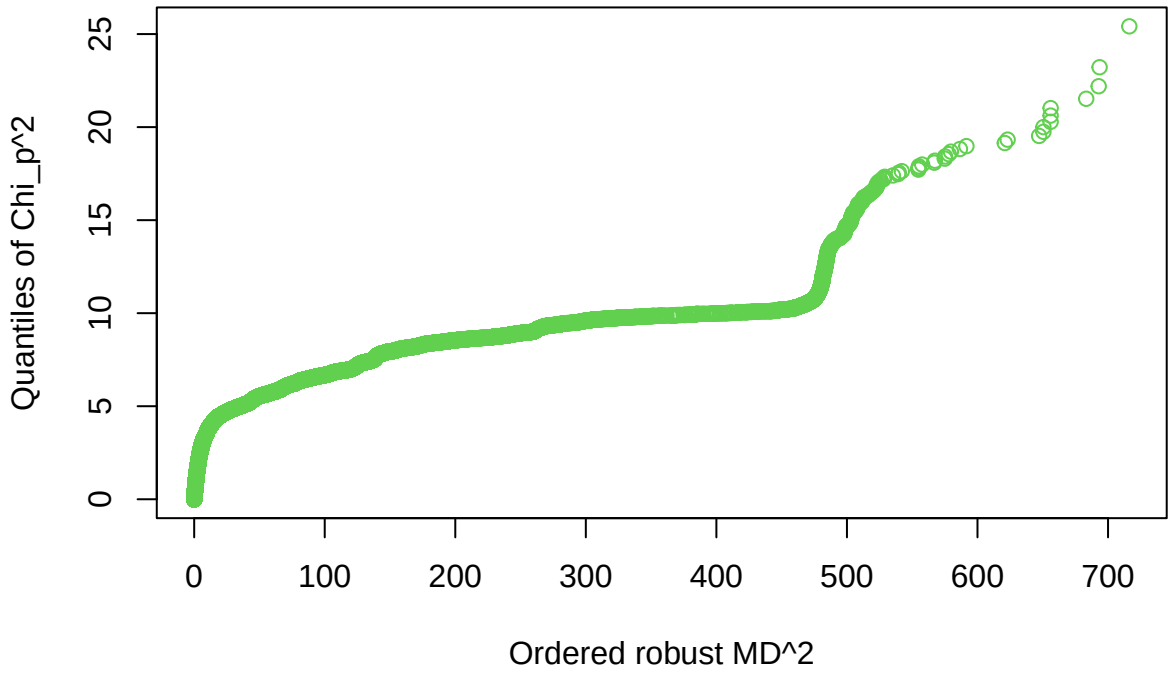



```
timeEndTest6 <- Sys.time()
timetaken6 <- difftime(timeEndTest6,timeStartTest6, units = "secs")
cat(glue('colorplot test secs: {timetaken6}'))
```

```
colorplot test secs: 0.644259214401245
```

chisq.plot The function chisq.plot plots the ordered robust mahalanobis distances of the data against the quantiles of the Chi-squared distribution. By user interaction this plotting is iterated each time leaving out the observation with the greatest distance.

```
timeStartTest7 <- Sys.time()
testchisq <- chisq.plot(testDF, quan = 0.5, ask = FALSE)
```




```
timeEndTest7 <- Sys.time()
timetaken7 <- difftime(timeEndTest7,timeStartTest7, units = "secs")
cat(glue('chisqplot test secs: {timetaken7}'))
```

```
chisqplot test secs: 0.348552942276001
```

```
sink("mvnSessionInfo.txt")
sessionInfo()
```

```
R version 4.4.0 (2024-04-24 ucrt)
Platform: x86_64-w64-mingw32/x64
Running under: Windows 11 x64 (build 22631)

Matrix products: default


locale:
[1] LC_COLLATE=English_Australia.utf8  LC_CTYPE=English_Australia.utf8
[3] LC_MONETARY=English_Australia.utf8 LC_NUMERIC=C
[5] LC_TIME=English_Australia.utf8

time zone: Australia/Sydney
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
```

```
 [1] glue_1.7.0       forcats_1.0.0    stringr_1.5.1    purrr_1.0.2
 [5] readr_2.1.5      tibble_3.2.1     ggplot2_3.5.1    tidyverse_2.0.0
 [9] kableExtra_1.4.0 knitr_1.47       dplyr_1.1.4      tidyr_1.3.1
[13] lubridate_1.9.3  mvoutlier_2.1.1  sgeostat_1.0-27  MVN_5.9

loaded via a namespace (and not attached):
 [1] gtable_0.3.5      xfun_0.44         psych_2.4.3       lattice_0.22-6
 [5] tzdb_0.4.0        vctrs_0.6.5       tools_4.4.0       generics_0.1.3
 [9] parallel_4.4.0    fansi_1.0.6       DEoptimR_1.1-3    highr_0.11
[13] pkgconfig_2.0.3   lifecycle_1.0.4   compiler_4.4.0    munsell_0.5.1
[17] mnormt_2.1.1      tinytex_0.51      carData_3.0-5     htmltools_0.5.8.1
[21] yaml_2.3.8        pillar_1.9.0      car_3.1-2         MASS_7.3-60.2
[25] boot_1.3-30       abind_1.4-5       nlme_3.1-164      robustbase_0.99-2
[29] tidyselect_1.2.1  digest_0.6.35     nortest_1.0-4     stringi_1.8.4
[33] gsl_2.1-8         fastmap_1.2.0     grid_4.4.0        colorspace_2.1-0
[37] cli_3.6.2         magrittr_2.0.3    utf8_1.2.4        withr_3.0.0
[41] scales_1.3.0      energy_1.7-11     timechange_0.3.0  rmarkdown_2.27
[45] moments_0.14.1    hms_1.1.3         evaluate_0.23     viridisLite_0.4.2
[49] rlang_1.1.3       Rcpp_1.0.12       xml2_1.3.6        svglite_2.1.3
[53] rstudioapi_0.16.0 R6_2.5.1          plyr_1.8.9        systemfonts_1.1.0
```

```
sink()
```