Outlier

Haineycf

December 12, 2016

:

```
library(outliers)
```

```
## Warning: package 'outliers' was built under R version 3.3.2
```

```
set.seed(1234)
x = rnorm(100)
d=data.frame(x=x,group=rep(1:10,10))
y=runif(5)
z = rnorm(10)
head(x)
```

```
## [1] -1.2070657  0.2774292  1.0844412 -2.3456977  0.4291247  0.5060559
```

```
head(y)
```

```
## [1] 0.6607546 0.5283594 0.3174938 0.7678555 0.5263085
```

```
head(d)
```

```
##            x group
## 1 -1.2070657     1
## 2  0.2774292     2
## 3  1.0844412     3
## 4 -2.3456977     4
## 5  0.4291247     5
## 6  0.5060559     6
```

check to see if there is 1 outlier, do another test for more than 1 this test is from 1950 and is a quick dirty method for extreme outlier with no known variance

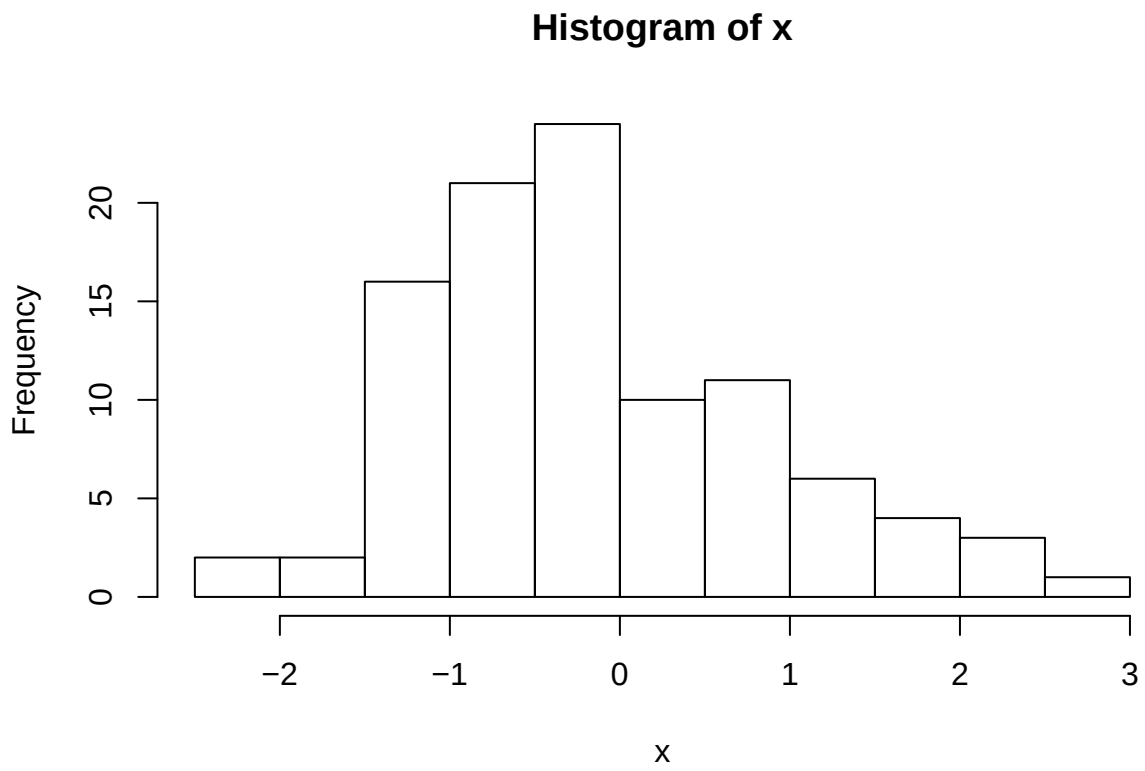
```
chisq.out.test(x)
```

```
##
##  chi-squared test for outlier
##
## data:  x
## X-squared = 7.257, p-value = 0.007062
## alternative hypothesis: highest value 2.54899107071786 is an outlier
```

```r
chisq.out.test(x,opposite=TRUE)
```

```
##
##  chi-squared test for outlier
##
## data:  x
## X-squared = 4.7495, p-value = 0.02931
## alternative hypothesis: lowest value -2.34569770262935 is an outlier
```

```r
hist(x)
```



to test for outlie or inlie variance, useful for structured data to determine if they should be separated

```r
cochran.test(x~group,d)
```

```
##
##  Cochran test for outlying variance
##
## data:  x ~ group
## C = 0.1619, df = 10, k = 10, p-value = 0.9349
## alternative hypothesis: Group 10 has outlying variance
## sample estimates:
##         1         2         3         4         5         6         7
## 0.9702567 1.0762180 0.9489474 0.9098832 1.1665495 0.9753678 1.2960769
```

```
##         8         9        10
## 0.8036918 0.7193931 1.7127172
```

```
cochran.test(x~group,d,inlying=TRUE)
```

```
##
##  Cochran test for inlying variance
##
## data:  x ~ group
## C = 0.068001, df = 10, k = 10, p-value < 2.2e-16
## alternative hypothesis: Group 9 has inlying variance
## sample estimates:
##         1         2         3         4         5         6         7
## 0.9702567 1.0762180 0.9489474 0.9098832 1.1665495 0.9753678 1.2960769
##         8         9        10
## 0.8036918 0.7193931 1.7127172
```

```
y=runif(5)
head(y)
```

```
## [1] 0.4995464 0.8021356 0.3371532 0.5089206 0.4944386
```

```
cochran.test(y,rep(5,5))
```

```
##
##  Cochran test for outlying variance
##
## data:  y
## C = 0.30359, df = 5, k = 5, p-value = 0.9485
## alternative hypothesis: Group 2 has outlying variance
## sample estimates:
##         1         2         3         4         5
## 0.4995464 0.8021356 0.3371532 0.5089206 0.4944386
```

```
cochran.test(y,rep(100,5))
```

```
##
##  Cochran test for outlying variance
##
## data:  y
## C = 0.30359, df = 100, k = 5, p-value = 0.0005095
## alternative hypothesis: Group 2 has outlying variance
## sample estimates:
##         1         2         3         4         5
## 0.4995464 0.8021356 0.3371532 0.5089206 0.4944386
```

```
dixon.test(z)
```

```
##
##  Dixon test for outliers
```

```
##
## data:  z
## Q = 0.38146, p-value = 0.2558
## alternative hypothesis: lowest value -1.86071635142819 is an outlier
```

```
dixon.test(z,opposite=TRUE)
```

```
##
##  Dixon test for outliers
##
## data:  z
## Q = 0.25099, p-value = 0.654
## alternative hypothesis: highest value 2.18711916123311 is an outlier
```

```
dixon.test(z,type=10)
```

```
##
##  Dixon test for outliers
##
## data:  z
## Q = 0.31597, p-value = 0.2757
## alternative hypothesis: lowest value -1.86071635142819 is an outlier
```

```
set.seed(1234)
x = rnorm(10)
grubbs.test(x)
```

```
##
##  Grubbs test for one outlier
##
## data:  x
## G = 1.97080, U = 0.52047, p-value = 0.1323
## alternative hypothesis: lowest value -2.34569770262935 is an outlier
```

```
grubbs.test(z,type=20)
```

```
##
##  Grubbs test for two outliers
##
## data:  z
## U = 0.4249, p-value = 0.3318
## alternative hypothesis: lowest values -1.86071635142819 , -0.581727450246202 are outliers
```

```
grubbs.test(z,type=11)
```

```
##
##  Grubbs test for two opposite outliers
##
## data:  z
## G = 3.49270, U = 0.31544, p-value = 0.1544
## alternative hypothesis: -1.86071635142819 and 2.18711916123311 are outliers
```

```r
x=rnorm(100)
outlier(x)
```

```
## [1] 2.548991
```

```r
outlier(x,opposite=TRUE)
```

```
## [1] -2.18004
```

```r
dim(x) <- c(20,5)
outlier(x)
```

```
## [1] 2.415835 1.449496 2.548991 2.070271 2.121117
```

```r
outlier(x,opposite=TRUE)
```

```
## [1] -1.448205 -2.180040 -1.806031 -1.390701 -1.134608
```

```r
qcochran(0.05,5,5)
```

```
## [1] 0.3034135
```

```r
pcochran(0.293,5,5)
```

```
## [1] 0
```

```r
outlier(x)
```

```
## [1] 2.415835 1.449496 2.548991 2.070271 2.121117
```

```r
outlier(x,opposite=TRUE)
```

```
## [1] -1.448205 -2.180040 -1.806031 -1.390701 -1.134608
```

```r
rm.outlier(x)
```

```
##              [,1]       [,2]         [,3]         [,4]        [,5]
##  [1,] -0.47719270  1.1022975 -1.806031257  0.006892838 -0.05315882
##  [2,] -0.99838644 -0.4755931 -0.582075925 -0.455468738  0.25519600
##  [3,] -0.77625389 -0.7094400 -1.108889624 -0.366523933  1.70596401
##  [4,]  0.06445882 -0.5012581 -1.014962009  0.648286568  1.00151325
##  [5,]  0.95949406 -1.6290935 -0.162309524 -0.153398412 -0.49558344
##  [6,] -0.11028549 -1.1676193  0.563055819 -1.390700947  0.35555030
##  [7,] -0.51100951 -2.1800396  1.647817473 -0.723581777 -1.13460804
##  [8,] -0.91119542 -1.3409932 -0.773353424  0.258261762  0.87820363
##  [9,] -0.83717168 -0.2942939  1.605909629 -0.317059115  0.97291675
## [10,]  0.13408822 -0.4658975 -1.157808548 -0.177789958  0.41452353
```

```
## [11,] -0.49068590 -1.0686427  0.656588464 -0.169994077 -0.47471847
## [12,] -0.44054787 -0.8553646 -0.034760390 -1.372301886  0.06599349
## [13,]  0.45958944 -0.2806230 -0.669633580 -0.173787170 -0.50247778
## [14,] -0.69372025 -0.9943401 -0.007604756  0.850232257 -0.82599859
## [15,] -1.44820491 -0.9685143  1.777084448  0.697608712  0.16698928
## [16,]  0.57475572 -1.1073182 -1.138607737  0.549997351 -0.89626463
## [17,] -1.02365572 -1.2519859  1.367827179 -0.402731975  0.16818539
## [18,] -0.01513830 -0.5238281  1.329564791 -0.191593770  0.35496826
## [19,] -0.93594860 -0.4968500  0.336472797 -1.194527880 -0.05210512
```

```
rm.outlier(x,opposite=TRUE)
```

```
##              [,1]       [,2]         [,3]         [,4]        [,5]
##  [1,] -0.47719270  1.1022975 -0.582075925  0.006892838 -0.05315882
##  [2,] -0.99838644 -0.4755931 -1.108889624 -0.455468738  0.25519600
##  [3,] -0.77625389 -0.7094400 -1.014962009 -0.366523933  1.70596401
##  [4,]  0.06445882 -0.5012581 -0.162309524  0.648286568  1.00151325
##  [5,]  0.95949406 -1.6290935  0.563055819  2.070270861 -0.49558344
##  [6,] -0.11028549 -1.1676193  1.647817473 -0.153398412  0.35555030
##  [7,] -0.51100951 -1.3409932 -0.773353424 -0.723581777  0.87820363
##  [8,] -0.91119542 -0.2942939  1.605909629  0.258261762  0.97291675
##  [9,] -0.83717168 -0.4658975 -1.157808548 -0.317059115  2.12111711
## [10,]  2.41583518  1.4494963  0.656588464 -0.177789958  0.41452353
## [11,]  0.13408822 -1.0686427  2.548991071 -0.169994077 -0.47471847
## [12,] -0.49068590 -0.8553646 -0.034760390 -1.372301886  0.06599349
## [13,] -0.44054787 -0.2806230 -0.669633580 -0.173787170 -0.50247778
## [14,]  0.45958944 -0.9943401 -0.007604756  0.850232257 -0.82599859
## [15,] -0.69372025 -0.9685143  1.777084448  0.697608712  0.16698928
## [16,]  0.57475572 -1.1073182 -1.138607737  0.549997351 -0.89626463
## [17,] -1.02365572 -1.2519859  1.367827179 -0.402731975  0.16818539
## [18,] -0.01513830 -0.5238281  1.329564791 -0.191593770  0.35496826
## [19,] -0.93594860 -0.4968500  0.336472797 -1.194527880 -0.05210512
```

```
dim(x) <- c(20,5)
outlier(x)
```

```
## [1] 2.415835 1.449496 2.548991 2.070271 2.121117
```

```
outlier(x,logical=TRUE)
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]
##  [1,] FALSE FALSE FALSE FALSE FALSE
##  [2,] FALSE FALSE FALSE FALSE FALSE
##  [3,] FALSE FALSE FALSE FALSE FALSE
##  [4,] FALSE FALSE FALSE FALSE FALSE
##  [5,] FALSE FALSE FALSE  TRUE FALSE
##  [6,] FALSE FALSE FALSE FALSE FALSE
##  [7,] FALSE FALSE FALSE FALSE FALSE
##  [8,] FALSE FALSE FALSE FALSE FALSE
##  [9,] FALSE FALSE FALSE FALSE FALSE
## [10,]  TRUE FALSE FALSE FALSE  TRUE
## [11,] FALSE  TRUE FALSE FALSE FALSE
```

```
## [12,] FALSE FALSE  TRUE FALSE FALSE
## [13,] FALSE FALSE FALSE FALSE FALSE
## [14,] FALSE FALSE FALSE FALSE FALSE
## [15,] FALSE FALSE FALSE FALSE FALSE
## [16,] FALSE FALSE FALSE FALSE FALSE
## [17,] FALSE FALSE FALSE FALSE FALSE
## [18,] FALSE FALSE FALSE FALSE FALSE
## [19,] FALSE FALSE FALSE FALSE FALSE
## [20,] FALSE FALSE FALSE FALSE FALSE
```

```
outlier(x,logical=TRUE,opposite=TRUE)
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]
##  [1,] FALSE FALSE  TRUE FALSE FALSE
##  [2,] FALSE FALSE FALSE FALSE FALSE
##  [3,] FALSE FALSE FALSE FALSE FALSE
##  [4,] FALSE FALSE FALSE FALSE FALSE
##  [5,] FALSE FALSE FALSE FALSE FALSE
##  [6,] FALSE FALSE FALSE FALSE FALSE
##  [7,] FALSE  TRUE FALSE  TRUE  TRUE
##  [8,] FALSE FALSE FALSE FALSE FALSE
##  [9,] FALSE FALSE FALSE FALSE FALSE
## [10,] FALSE FALSE FALSE FALSE FALSE
## [11,] FALSE FALSE FALSE FALSE FALSE
## [12,] FALSE FALSE FALSE FALSE FALSE
## [13,] FALSE FALSE FALSE FALSE FALSE
## [14,] FALSE FALSE FALSE FALSE FALSE
## [15,] FALSE FALSE FALSE FALSE FALSE
## [16,]  TRUE FALSE FALSE FALSE FALSE
## [17,] FALSE FALSE FALSE FALSE FALSE
## [18,] FALSE FALSE FALSE FALSE FALSE
## [19,] FALSE FALSE FALSE FALSE FALSE
## [20,] FALSE FALSE FALSE FALSE FALSE
```

```
rm.outlier(x)
```

```
##              [,1]       [,2]         [,3]         [,4]        [,5]
##  [1,] -0.47719270  1.1022975 -1.806031257  0.006892838 -0.05315882
##  [2,] -0.99838644 -0.4755931 -0.582075925 -0.455468738  0.25519600
##  [3,] -0.77625389 -0.7094400 -1.108889624 -0.366523933  1.70596401
##  [4,]  0.06445882 -0.5012581 -1.014962009  0.648286568  1.00151325
##  [5,]  0.95949406 -1.6290935 -0.162309524 -0.153398412 -0.49558344
##  [6,] -0.11028549 -1.1676193  0.563055819 -1.390700947  0.35555030
##  [7,] -0.51100951 -2.1800396  1.647817473 -0.723581777 -1.13460804
##  [8,] -0.91119542 -1.3409932 -0.773353424  0.258261762  0.87820363
##  [9,] -0.83717168 -0.2942939  1.605909629 -0.317059115  0.97291675
## [10,]  0.13408822 -0.4658975 -1.157808548 -0.177789958  0.41452353
## [11,] -0.49068590 -1.0686427  0.656588464 -0.169994077 -0.47471847
## [12,] -0.44054787 -0.8553646 -0.034760390 -1.372301886  0.06599349
## [13,]  0.45958944 -0.2806230 -0.669633580 -0.173787170 -0.50247778
## [14,] -0.69372025 -0.9943401 -0.007604756  0.850232257 -0.82599859
## [15,] -1.44820491 -0.9685143  1.777084448  0.697608712  0.16698928
## [16,]  0.57475572 -1.1073182 -1.138607737  0.549997351 -0.89626463
```

```
## [17,] -1.02365572 -1.2519859  1.367827179 -0.402731975  0.16818539
## [18,] -0.01513830 -0.5238281  1.329564791 -0.191593770  0.35496826
## [19,] -0.93594860 -0.4968500  0.336472797 -1.194527880 -0.05210512
```

```
rm.outlier(x,opposite=TRUE)
```

```
##              [,1]       [,2]         [,3]         [,4]        [,5]
##  [1,] -0.47719270  1.1022975 -0.582075925  0.006892838 -0.05315882
##  [2,] -0.99838644 -0.4755931 -1.108889624 -0.455468738  0.25519600
##  [3,] -0.77625389 -0.7094400 -1.014962009 -0.366523933  1.70596401
##  [4,]  0.06445882 -0.5012581 -0.162309524  0.648286568  1.00151325
##  [5,]  0.95949406 -1.6290935  0.563055819  2.070270861 -0.49558344
##  [6,] -0.11028549 -1.1676193  1.647817473 -0.153398412  0.35555030
##  [7,] -0.51100951 -1.3409932 -0.773353424 -0.723581777  0.87820363
##  [8,] -0.91119542 -0.2942939  1.605909629  0.258261762  0.97291675
##  [9,] -0.83717168 -0.4658975 -1.157808548 -0.317059115  2.12111711
## [10,]  2.41583518  1.4494963  0.656588464 -0.177789958  0.41452353
## [11,]  0.13408822 -1.0686427  2.548991071 -0.169994077 -0.47471847
## [12,] -0.49068590 -0.8553646 -0.034760390 -1.372301886  0.06599349
## [13,] -0.44054787 -0.2806230 -0.669633580 -0.173787170 -0.50247778
## [14,]  0.45958944 -0.9943401 -0.007604756  0.850232257 -0.82599859
## [15,] -0.69372025 -0.9685143  1.777084448  0.697608712  0.16698928
## [16,]  0.57475572 -1.1073182 -1.138607737  0.549997351 -0.89626463
## [17,] -1.02365572 -1.2519859  1.367827179 -0.402731975  0.16818539
## [18,] -0.01513830 -0.5238281  1.329564791 -0.191593770  0.35496826
## [19,] -0.93594860 -0.4968500  0.336472797 -1.194527880 -0.05210512
```

```
set.seed(1234)
z = rnorm(10)
scores(x)
```

```
##              [,1]        [,2]       [,3]        [,4]        [,5]
##  [1,] -0.2556096  2.17307642 -1.6138895  0.13171296 -0.30333388
##  [2,] -0.8499958  0.25781581 -0.6136699 -0.43590728  0.06423316
##  [3,] -0.5966686 -0.02603014 -1.0441835 -0.32671380  1.79358671
##  [4,]  0.3621073  0.22666338 -0.9674256  0.91912283  0.95386292
##  [5,]  1.3828344 -1.14231537 -0.2706357  2.66482817 -0.83071564
##  [6,]  0.1628232 -0.58217301  0.3221348 -0.06506930  0.18385812
##  [7,] -0.2941754 -1.81105977  1.2086050 -1.58404927 -1.59244974
##  [8,] -0.7505603 -0.79261615 -0.7699824 -0.76505745  0.80687461
##  [9,] -0.6661413  0.47787876  1.1743578  0.44030714  0.91977514
## [10,]  3.0436929  0.26958436 -1.0841602 -0.26598809  2.28846011
## [11,]  0.4415149  2.59450996  0.3985700 -0.09501369  0.25415577
## [12,] -0.2709977 -0.46203423  1.9450465 -0.08544305 -0.80584405
## [13,] -0.2138186 -0.20315499 -0.1664022 -1.56146158 -0.16130119
## [14,]  0.8127270  0.49447259 -0.6852223 -0.09009965 -0.83893388
## [15,] -0.5025446 -0.37184487 -0.1442105  1.16704235 -1.22457917
## [16,] -1.3629833 -0.34049729  1.3142423  0.97967338 -0.04091156
## [17,]  0.9440664 -0.50897893 -1.0684692  0.79845763 -1.30833813
## [18,] -0.8788137 -0.68457813  0.9797961 -0.37116476 -0.03948577
## [19,]  0.2713322  0.19926760  0.9485280 -0.11196001  0.18316432
## [20,] -0.7787896  0.23201398  0.1369706 -1.34321653 -0.30207784
```

```r
scores(x,prob=1)
```

```
##             [,1]      [,2]      [,3]       [,4]       [,5]
##  [1,] 0.39912616 0.9851127 0.0532757 0.55239433 0.38081772
##  [2,] 0.19766372 0.6017255 0.2697167 0.33145200 0.52560771
##  [3,] 0.27536432 0.4896166 0.1482003 0.37194218 0.96356042
##  [4,] 0.64136408 0.5896572 0.1666657 0.82098433 0.82992348
##  [5,] 0.91664218 0.1266615 0.3933356 0.99614862 0.20306714
##  [6,] 0.56467120 0.2802251 0.6263247 0.47405941 0.57293762
##  [7,] 0.38431195 0.0350658 0.8865927 0.05659125 0.05564184
##  [8,] 0.22645866 0.2140007 0.2206552 0.22211867 0.79013064
##  [9,] 0.25266041 0.6836317 0.8798741 0.67014266 0.82115486
## [10,] 0.99883153 0.6062600 0.1391469 0.39512419 0.98894463
## [11,] 0.67057987 0.9952637 0.6548949 0.46215198 0.60031239
## [12,] 0.39319641 0.3220284 0.9741153 0.46595459 0.21016639
## [13,] 0.41534424 0.4195069 0.4339202 0.05920744 0.43592809
## [14,] 0.79181271 0.6895138 0.2466018 0.46410401 0.20075321
## [15,] 0.30764224 0.3550042 0.4426671 0.87840337 0.11036691
## [16,] 0.08644389 0.3667410 0.9056177 0.83637631 0.48368320
## [17,] 0.82743214 0.3053835 0.1426545 0.78769752 0.09537932
## [18,] 0.18975115 0.2468051 0.8364066 0.35525741 0.48425155
## [19,] 0.60693222 0.5789733 0.8285696 0.45542756 0.57266545
## [20,] 0.21805182 0.5917364 0.5544730 0.08960094 0.38129636
```

```r
scores(x,prob=0.5)
```

```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,] TRUE TRUE TRUE TRUE TRUE
##  [2,] TRUE TRUE TRUE TRUE TRUE
##  [3,] TRUE TRUE TRUE TRUE TRUE
##  [4,] TRUE TRUE TRUE TRUE TRUE
##  [5,] TRUE TRUE TRUE TRUE TRUE
##  [6,] TRUE TRUE TRUE TRUE TRUE
##  [7,] TRUE TRUE TRUE TRUE TRUE
##  [8,] TRUE TRUE TRUE TRUE TRUE
##  [9,] TRUE TRUE TRUE TRUE TRUE
## [10,] TRUE TRUE TRUE TRUE TRUE
## [11,] TRUE TRUE TRUE TRUE TRUE
## [12,] TRUE TRUE TRUE TRUE TRUE
## [13,] TRUE TRUE TRUE TRUE TRUE
## [14,] TRUE TRUE TRUE TRUE TRUE
## [15,] TRUE TRUE TRUE TRUE TRUE
## [16,] TRUE TRUE TRUE TRUE TRUE
## [17,] TRUE TRUE TRUE TRUE TRUE
## [18,] TRUE TRUE TRUE TRUE TRUE
## [19,] TRUE TRUE TRUE TRUE TRUE
## [20,] TRUE TRUE TRUE TRUE TRUE
```

```r
scores(x,prob=0.1)
```

```
##       [,1] [,2] [,3] [,4] [,5]
```

```
##  [1,] TRUE TRUE TRUE TRUE TRUE
##  [2,] TRUE TRUE TRUE TRUE TRUE
##  [3,] TRUE TRUE TRUE TRUE TRUE
##  [4,] TRUE TRUE TRUE TRUE TRUE
##  [5,] TRUE TRUE TRUE TRUE TRUE
##  [6,] TRUE TRUE TRUE TRUE TRUE
##  [7,] TRUE TRUE TRUE TRUE TRUE
##  [8,] TRUE TRUE TRUE TRUE TRUE
##  [9,] TRUE TRUE TRUE TRUE TRUE
## [10,] TRUE TRUE TRUE TRUE TRUE
## [11,] TRUE TRUE TRUE TRUE TRUE
## [12,] TRUE TRUE TRUE TRUE TRUE
## [13,] TRUE TRUE TRUE TRUE TRUE
## [14,] TRUE TRUE TRUE TRUE TRUE
## [15,] TRUE TRUE TRUE TRUE TRUE
## [16,] TRUE TRUE TRUE TRUE TRUE
## [17,] TRUE TRUE TRUE TRUE TRUE
## [18,] TRUE TRUE TRUE TRUE TRUE
## [19,] TRUE TRUE TRUE TRUE TRUE
## [20,] TRUE TRUE TRUE TRUE TRUE
```

```
scores(x,prob=0.93)
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]
##  [1,] FALSE  TRUE  TRUE FALSE FALSE
##  [2,] FALSE FALSE FALSE FALSE FALSE
##  [3,] FALSE FALSE FALSE FALSE  TRUE
##  [4,] FALSE FALSE FALSE FALSE FALSE
##  [5,] FALSE FALSE FALSE  TRUE FALSE
##  [6,] FALSE FALSE FALSE FALSE FALSE
##  [7,] FALSE  TRUE FALSE  TRUE  TRUE
##  [8,] FALSE FALSE FALSE FALSE FALSE
##  [9,] FALSE FALSE FALSE FALSE FALSE
## [10,]  TRUE FALSE FALSE FALSE  TRUE
## [11,] FALSE  TRUE FALSE FALSE FALSE
## [12,] FALSE FALSE  TRUE FALSE FALSE
## [13,] FALSE FALSE FALSE  TRUE FALSE
## [14,] FALSE FALSE FALSE FALSE FALSE
## [15,] FALSE FALSE FALSE FALSE FALSE
## [16,] FALSE FALSE FALSE FALSE FALSE
## [17,] FALSE FALSE FALSE FALSE FALSE
## [18,] FALSE FALSE FALSE FALSE FALSE
## [19,] FALSE FALSE FALSE FALSE FALSE
## [20,] FALSE FALSE FALSE FALSE FALSE
```

```
scores(x,type="iqr")
```

```
##             [,1]        [,2]        [,3]        [,4]        [,5]
##  [1,]  0.00000000  2.42669109 -0.44745822  0.00000000  0.00000000
##  [2,] -0.15221564  0.00000000  0.00000000 -0.05294063  0.00000000
##  [3,]  0.00000000  0.00000000 -0.12662149  0.00000000  1.16344589
##  [4,]  0.00000000  0.00000000 -0.08339436  0.42442228  0.46623102
##  [5,]  0.93609270 -0.78046992  0.00000000  2.32773079 -0.01548799
```

```
##  [6,]  0.00000000 -0.06966129  0.00000000  0.00000000  0.00000000
##  [7,]  0.00000000 -1.62909220  0.14206318 -1.30473745 -0.64794874
##  [8,] -0.05921622 -0.33670912  0.00000000 -0.41180664  0.34418799
##  [9,]  0.00000000  0.27552164  0.12277646  0.00000000  0.43792826
## [10,]  2.48945068  0.01120053 -0.14913483  0.00000000  1.57433467
## [11,]  0.05570092  2.96148120  0.00000000  0.00000000  0.00000000
## [12,]  0.00000000  0.00000000  0.55679898  0.00000000  0.00000000
## [13,]  0.00000000  0.00000000  0.00000000 -1.28011054  0.00000000
## [14,]  0.40288601  0.29657886  0.00000000  0.00000000 -0.02231151
## [15,]  0.00000000  0.00000000  0.00000000  0.69472411 -0.34250922
## [16,] -0.63199960  0.00000000  0.20155409  0.49043937  0.00000000
## [17,]  0.52572430  0.00000000 -0.14029828  0.29286338 -0.41205351
## [18,] -0.17916828 -0.19961119  0.01320676  0.00000000  0.00000000
## [19,]  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000
## [20,] -0.08561839  0.00000000  0.00000000 -1.04216221  0.00000000
```

```
scores(x,type="mad")
```

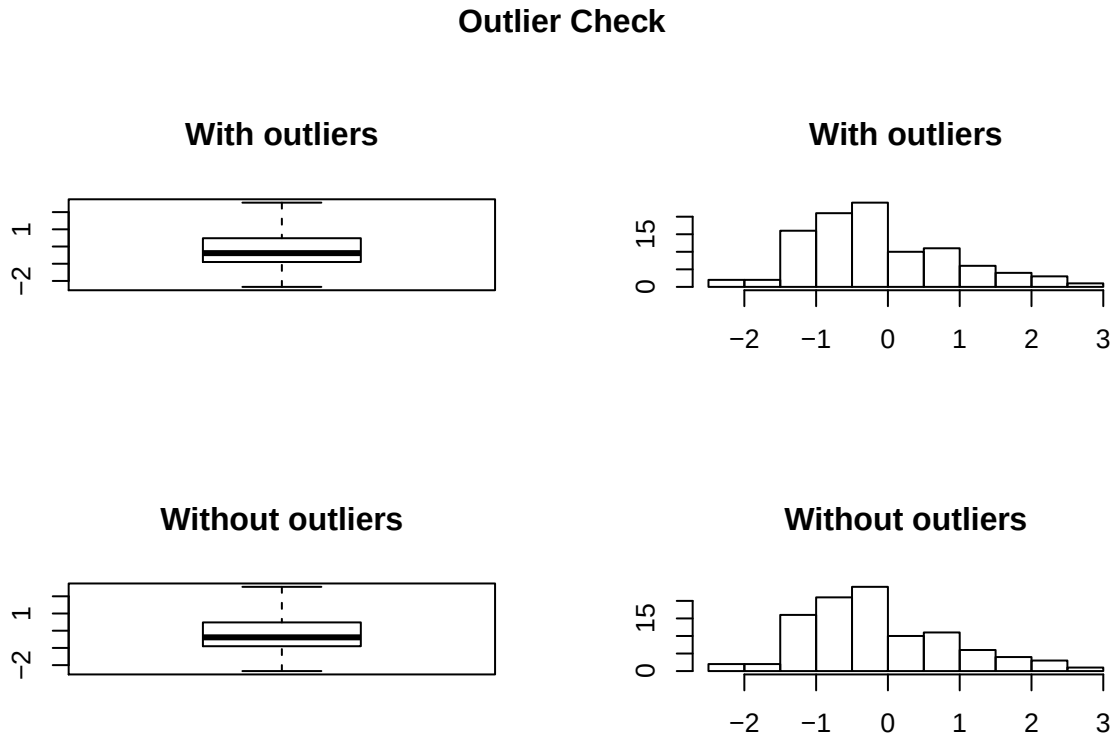
```
##               [,1]       [,2]         [,3]         [,4]          [,5]
##  [1,]  0.009883726  3.9637410 -1.156734815  0.345275753 -0.2281025058
##  [2,] -0.753661957  0.6452552 -0.363506931 -0.528607627  0.0905282213
##  [3,] -0.428239115  0.1534481 -0.704927329 -0.360498098  1.5896429912
##  [4,]  0.803399921  0.5912788 -0.644054190  1.557537645  0.8617164538
##  [5,]  2.114621134 -1.7806890 -0.091462351  4.245149470 -0.6852709062
##  [6,]  0.547400562 -0.8101556  0.378636517  0.042318355  0.1942268187
##  [7,] -0.039657690 -2.9393923  1.081654975 -2.296236956 -1.3455909662
##  [8,] -0.625927628 -1.1747809 -0.487471133 -1.035352868  0.7342975419
##  [9,] -0.517483302  1.0265484  1.054495103  0.820373892  0.8321669781
## [10,]  4.248152062  0.6656460 -0.736630981 -0.267007452  2.0186310434
## [11,]  0.905406575  4.6939399  0.439253682 -0.003782721  0.2551653359
## [12,] -0.009883726 -0.6019965  1.665692650  0.010951831 -0.6637106128
## [13,]  0.063568176 -0.1534481 -0.008799588 -2.261461935 -0.1049794491
## [14,]  1.382263894  1.0552997 -0.420251791  0.003782721 -0.6923949991
## [15,] -0.307327821 -0.4457295  0.008799588  1.939223734 -1.0266971167
## [16,] -1.412643287 -0.3914149  1.165431047  1.650758631 -0.0006179841
## [17,]  1.550981797 -0.6833355 -0.724187211  1.371766774 -1.0993047677
## [18,] -0.790681296 -0.9875883  0.900197285 -0.428932864  0.0006179841
## [19,]  0.686790625  0.5438114  0.875399980 -0.029872524  0.1936253864
## [20,] -0.662190894  0.6005495  0.231791312 -1.925461373 -0.2270136893
```

```
scores(x,prob=0)
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]
##  [1,] FALSE FALSE FALSE FALSE FALSE
##  [2,] FALSE FALSE FALSE FALSE FALSE
##  [3,] FALSE FALSE FALSE FALSE FALSE
##  [4,] FALSE FALSE FALSE FALSE FALSE
##  [5,] FALSE FALSE FALSE FALSE FALSE
##  [6,] FALSE FALSE FALSE FALSE FALSE
##  [7,] FALSE FALSE FALSE FALSE FALSE
##  [8,] FALSE FALSE FALSE FALSE FALSE
##  [9,] FALSE FALSE FALSE FALSE FALSE
## [10,] FALSE FALSE FALSE FALSE FALSE
```

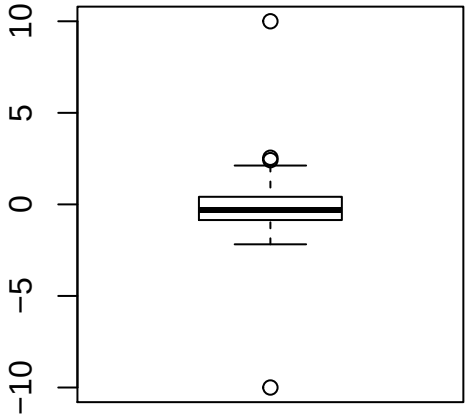
```
## [11,] FALSE FALSE FALSE FALSE FALSE
## [12,] FALSE FALSE FALSE FALSE FALSE
## [13,] FALSE FALSE FALSE FALSE FALSE
## [14,] FALSE FALSE FALSE FALSE FALSE
## [15,] FALSE FALSE FALSE FALSE FALSE
## [16,] FALSE FALSE FALSE FALSE FALSE
## [17,] FALSE FALSE FALSE FALSE FALSE
## [18,] FALSE FALSE FALSE FALSE FALSE
## [19,] FALSE FALSE FALSE FALSE FALSE
## [20,] FALSE FALSE FALSE FALSE FALSE
```

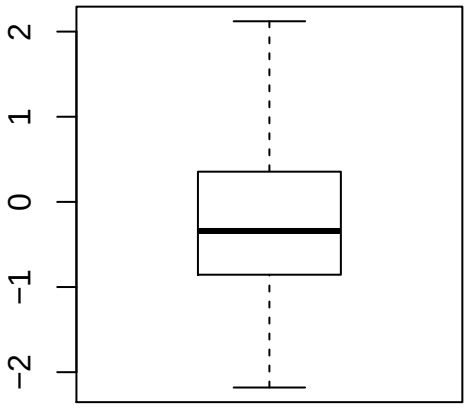
```
###############################
##https://www.r-bloggers.com/identify-describe-plot-and-remove-the-outliers-from-the-dataset/
## By Klodian Dhana

outlierKD <- function(dt, var) {
  var_name <- eval(substitute(var),eval(dt))
  na1 <- sum(is.na(var_name))
  m1 <- mean(var_name, na.rm = T)
  par(mfrow=c(2, 2), oma=c(0,0,3,0))
  boxplot(var_name, main="With outliers")
  hist(var_name, main="With outliers", xlab=NA, ylab=NA)
  outlier <- boxplot.stats(var_name)$out
  mo <- mean(outlier)
  var_name <- ifelse(var_name %in% outlier, NA, var_name)
  boxplot(var_name, main="Without outliers")
  hist(var_name, main="Without outliers", xlab=NA, ylab=NA)
  title("Outlier Check", outer=TRUE)
  na2 <- sum(is.na(var_name))
  cat("Outliers identified:", na2 - na1, "n")
  cat("Propotion (%) of outliers:", round((na2 - na1) / sum(!is.na(var_name))*100, 1), "n")
  cat("Mean of the outliers:", round(mo, 2), "n")
  m2 <- mean(var_name, na.rm = T)
  cat("Mean without removing outliers:", round(m1, 2), "n")
  cat("Mean if we remove outliers:", round(m2, 2), "n")
  response <- readline(prompt="Do you want to remove outliers and to replace with NA? [yes/no]: ")
  if(response == "y" | response == "yes"){
    dt[as.character(substitute(var))] <- invisible(var_name)
    assign(as.character(as.list(match.call())$dt), dt, envir = .GlobalEnv)
    cat("Outliers successfully removed", "n")
    return(invisible(dt))
  } else{
    cat("Nothing changed", "n")
    return(invisible(var_name))
  }
}
outlierKD(d, x)  ### need wide screen to show plots
```




```
## Outliers identified: 0 nPropotion (%) of outliers: 0 nMean of the outliers: NaN nMean without removi
## Nothing changed n
```

```
############################
remove_outliers <- function(x, na.rm = TRUE, ...) {
  qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm, ...)
  H <- 1.5 * IQR(x, na.rm = na.rm)
  x_rm <- x
  x_rm[x < (qnt[1] - H)] <- NA
  x_rm[x > (qnt[2] + H)] <- NA
  x_rm
}

x <- c(-10, x, 10)
x_rm <- remove_outliers(x)
## png()
par(mfrow = c(1, 2))
boxplot(x)
boxplot(x_rm)
```






```
## dev.off()
```