QUANTILE CALCULATIONS IN R

Objective:
Showing how quantiles (esp. quartiles) are calculated in R.
R offers different functions to calculate quartiles, which can produce different output.

Examples:

```
> data <- c(1,12,14,3,96,111)

> summary(data)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    5.25   13.00   39.50   75.50  111.00

> quantile(data, c(0.25, 0.5, 0.75), type = 1)
25% 50% 75%
  3  12  96
```

Sources:
- http://stat.ethz.ch/R-manual/R-patched/library/stats/html/quantile.html
– http://en.wikipedia.org/wiki/Quantile

1.Defining test sets

Q1 **MEDIAN** **Q3**

3			**3**			**3**			**3**		
2	3	5	7	11	13	17	19	23	29	31	37

```
> data12 <- c(2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37)

> length(data12) / 4 ## Length of each quartile
[1] 3
```

Q1 **MEDIAN** **Q3**

2.75		**2.75**			**2.75**			**2.75**		
2	3	5	7	11	13	17	19	23	29	31

```
> data11 <- c(2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)

> length(data11) / 4 ## Length of each quartile
[1] 2.75
```

Q1 **MEDIAN** **Q3**

2.5		**2.5**			**2.5**		**2.5**		
2	3	5	7	11	13	17	19	23	29

```
> data10 <- c(2, 3, 5, 7, 11, 13, 17, 19, 23, 29)

> length(data10) / 4 ## Length of each quartile
[1] 2.5
```

Q1 **MEDIAN** **Q3**

2.25		**2.25**		**2.25**		**2.25**		
2	3	5	7	11	13	17	19	23

```
> data9 <- c(2, 3, 5, 7, 11, 13, 17, 19, 23)

> length(data9) / 4 ## Length of each quartile
[1] 2.25
```

Comparison of result sets for different functions

DATA	FUNCTION	Q1	MEDIAN	Q3
data12	quantile(data12, c(0.25, 0.5, 0.75), type = 1)	5	13	23
	quantile(data12, c(0.25, 0.5, 0.75), type = 2)	6	15	26
	quantile(data12, c(0.25, 0.5, 0.75), type = 3)	5	13	23
	quantile(data12, c(0.25, 0.5, 0.75), type = 4)	5	13	23
	quantile(data12, c(0.25, 0.5, 0.75), type = 5)	6	15	26
	quantile(data12, c(0.25, 0.5, 0.75), type = 6)	5.5	15	27.5
	quantile(data12, c(0.25, 0.5, 0.75), type = 7)	**6.5**	**15**	**24.5**
	quantile(data12, c(0.25, 0.5, 0.75), type = 8)	5.833333	15	26.5
	quantile(data12, c(0.25, 0.5, 0.75), type = 9)	5.875	15	26.375
	summary(data12)	**6.5**	**15**	**24.5**
	boxplot(data12)	6	15	26
data11	quantile(data11, c(0.25, 0.5, 0.75), type = 1)	5	13	23
	quantile(data11, c(0.25, 0.5, 0.75), type = 2)	5	13	23
	quantile(data11, c(0.25, 0.5, 0.75), type = 3)	5	13	19
	quantile(data11, c(0.25, 0.5, 0.75), type = 4)	4.5	12	20
	quantile(data11, c(0.25, 0.5, 0.75), type = 5)	5.5	13	22
	quantile(data11, c(0.25, 0.5, 0.75), type = 6)	5	13	23
	quantile(data11, c(0.25, 0.5, 0.75), type = 7)	**6**	**13**	**21**
	quantile(data11, c(0.25, 0.5, 0.75), type = 8)	5.333333	13	22.333333
	quantile(data11, c(0.25, 0.5, 0.75), type = 9)	5.375	13	22.25
	summary(data11)	**6**	**13**	**21**
	boxplot(data11)	6	13	21
data10	quantile(data10, c(0.25, 0.5, 0.75), type = 1)	5	11	19
	quantile(data10, c(0.25, 0.5, 0.75), type = 2)	5	12	19
	quantile(data10, c(0.25, 0.5, 0.75), type = 3)	3	11	19
	quantile(data10, c(0.25, 0.5, 0.75), type = 4)	4	11	18
	quantile(data10, c(0.25, 0.5, 0.75), type = 5)	5	12	19
	quantile(data10, c(0.25, 0.5, 0.75), type = 6)	4.5	12	20
	quantile(data10, c(0.25, 0.5, 0.75), type = 7)	**5.5**	**12**	**18.5**
	quantile(data10, c(0.25, 0.5, 0.75), type = 8)	4.833333	12	19.333333
	quantile(data10, c(0.25, 0.5, 0.75), type = 9)	4.875	12	19.25
	summary(data10)	**5.5**	**12**	**18.5**
	boxplot(data10)	5	12	19
data9	quantile(data9, c(0.25, 0.5, 0.75), type = 1)	5	11	17
	quantile(data9, c(0.25, 0.5, 0.75), type = 2)	5	11	17
	quantile(data9, c(0.25, 0.5, 0.75), type = 3)	3	7	17
	quantile(data9, c(0.25, 0.5, 0.75), type = 4)	3.5	9	16
	quantile(data9, c(0.25, 0.5, 0.75), type = 5)	4.5	11	17.5
	quantile(data9, c(0.25, 0.5, 0.75), type = 6)	4	11	18
	quantile(data9, c(0.25, 0.5, 0.75), type = 7)	**5**	**11**	**17**
	quantile(data9, c(0.25, 0.5, 0.75), type = 8)	4.333333	11	17.666667
	quantile(data9, c(0.25, 0.5, 0.75), type = 9)	4.375	11	17.625
	summary(data9)	**5**	**11**	**17**
	boxplot(data9)	5	11	17

Custom quantile functions per type

```
QuantileType1 <- function (v, p) {
                                   v = sort(v)
                                   m = 0
                                   n = length(v)
                                   j = floor((n * p) + m)
                                   g = (n * p) + m - j
                                   y = ifelse (g == 0, 0, 1)
                                   ((1 - y) * v[j]) + (y * v[j+1])
                                 }


QuantileType2 <- function (v, p) {
                                   v = sort(v)
                                   m = 0
                                   n = length(v)
                                   j = floor((n * p) + m)
                                   g = (n * p) + m - j
                                   y = ifelse (g == 0, 0.5, 1)
                                   ((1 - y) * v[j]) + (y * v[j+1])
                                 }


QuantileType3 <- function (v, p) {
                                   v = sort(v)
                                   m = -0.5
                                   n = length(v)
                                   j = floor((n * p) + m)
                                   g = (n * p) + m - j
                                   y = ifelse(trunc(j/2)*2==j, ifelse(g==0, 0, 1), 1)
                                   ((1 - y) * v[j]) + (y * v[j+1])
                                 }


QuantileType7 <- function (v, p) {
                                   v = sort(v)
                                   h = ((length(v)-1)*p)+1
                                   v[floor(h)]+((h-floor(h))*(v[floor(h)+1]- v[floor(h)]))
                                 }
```

Example:

```
> data12 <- c(2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37)
> QuantileType1(data12, 0.25)
[1] 5
```

Conclusions

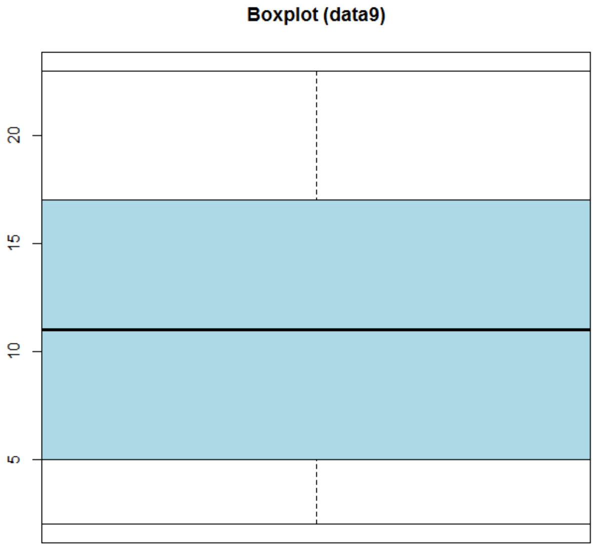
The function summary() seems to use the same algorithm for calculating Q1, median and Q3 as does the function quantiles() with type set to 7.

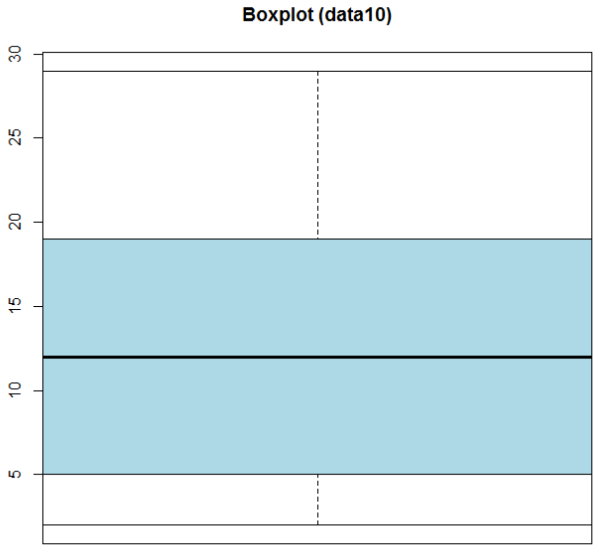
Sometimes, the function quantiles() generates the same results with different types set.

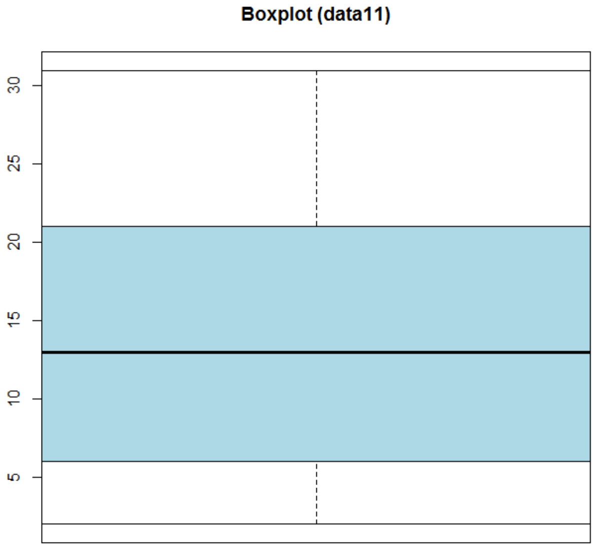
Boxplot does not seem to use one of the 9 types that quantiles() uses to calculate Q1, median and Q3.

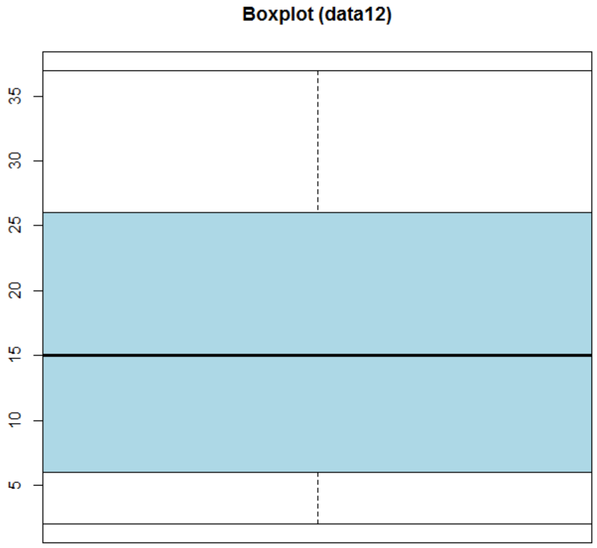
Boxplots

```
boxplot(data9 , pch=15, main="Boxplot (data9)" , col = "lightblue", pars = list(boxwex = 5))
boxplot(data10, pch=15, main="Boxplot (data10)", col = "lightblue", pars = list(boxwex = 5))
boxplot(data11, pch=15, main="Boxplot (data11)", col = "lightblue", pars = list(boxwex = 5))
boxplot(data12, pch=15, main="Boxplot (data12)", col = "lightblue", pars = list(boxwex = 5))
```


Boxplot (data9)


Boxplot (data10)


Boxplot (data11)


Boxplot (data12)