Статистическая обработка данных в R

© А.Б. Шипунов*, Е.М. Балдин†

3 мая 2008 г.

1 Приёмы элементарного анализа данных

Зарплаты сотрудников виртуальной компании:

```
> salary <- c(21, 19, 27, 11, 102, 25, 21)

[1]  21  19  27  11 102  25  21

> names(salary) <- c("Коля", "Женя", "Петя", "Саша", "Катя", "Вася",
+     "Жора")

[1] "Коля" "Женя" "Петя" "Саша" "Катя" "Вася" "Жора"

> salary

Коля Женя Петя Саша Катя Вася Жора
  21   19   27   11  102   25   21
```

Посмотрим чему равен центр:

```
> mean(salary)

[1] 32.28571

> median(salary)

[1] 21
```

Получение среднего на примере встроенных данных trees:

```
> attach(trees)
```

*e-mail: dactylorhiza@gmail.com
†e-mail: E.M.Baldin@inp.nsk.su

```
<environment: 0x8725348>
attr(,"name")
[1] "trees"

> mean(Girth)

[1] 13.24839

> mean(Height)

[1] 76

> mean(Volume/Height)

[1] 0.3890012

> detach(trees)

NULL

> with(trees, mean(Volume/Height))

[1] 0.3890012

> lapply(trees, mean)

$Girth
[1] 13.24839

$Height
[1] 76

$Volume
[1] 30.17097
```

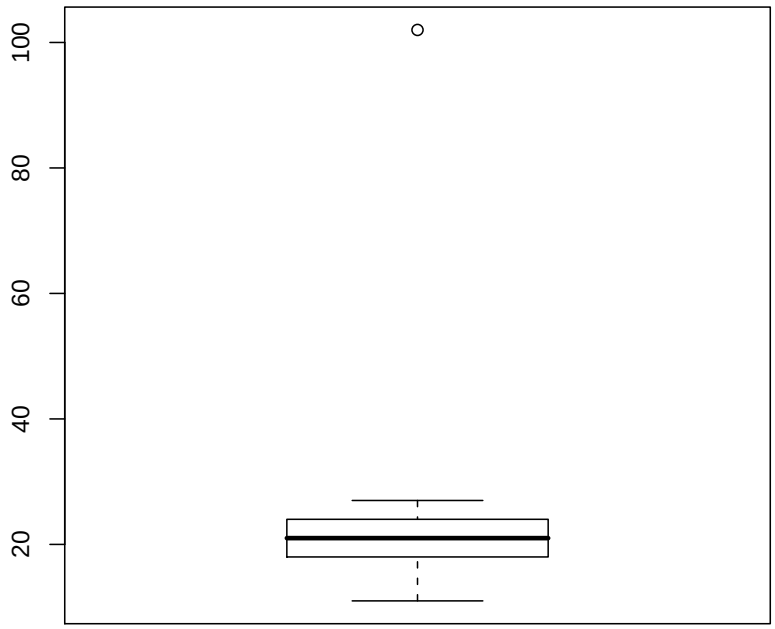
Стандартное отклонение, варанса (его квадрат) и межквартильный размах:

```
> sd(salary)

[1] 31.15934

> var(salary)

[1] 970.9048

> IQR(salary)
```

```
[1] 6

> attach(trees)

<environment: 0x85b8b6c>
attr(,"name")
[1] "trees"

> mean(Height)

[1] 76

> median(Height)

[1] 76

> sd(Height)

[1] 6.371813

> IQR(Height)

[1] 8

> detach(trees)

NULL
```

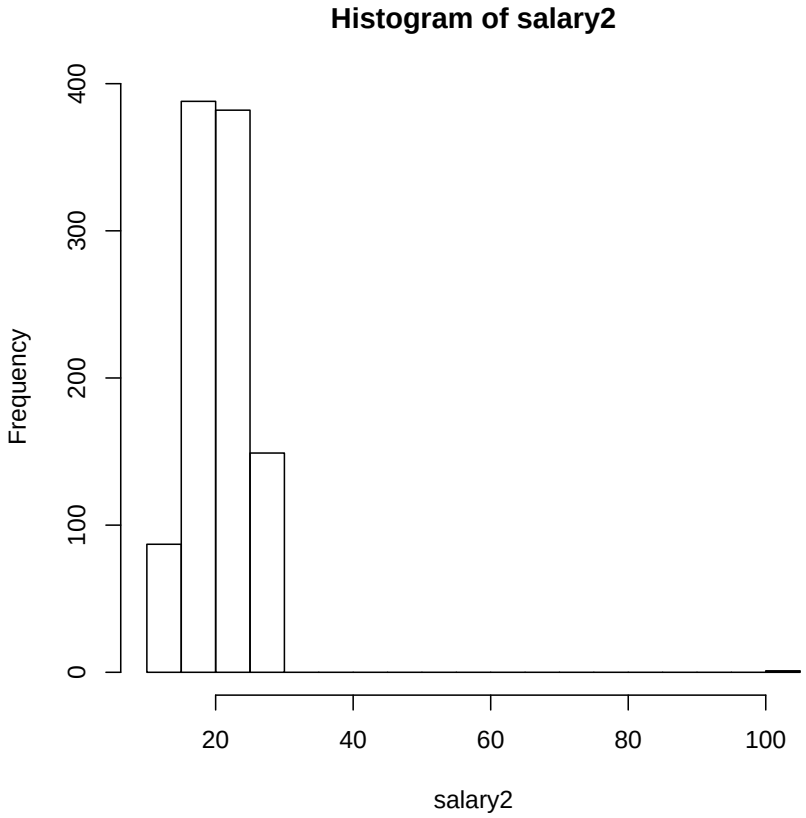
«Ящик-с-усами», или боксплот:

```
> new.1000 <- sample((median(salary) - IQR(salary)):(median(salary) +
+     IQR(salary)), 1000, replace = TRUE)
> salary2 <- c(salary, new.1000)
> boxplot(salary2)
> boxplot(trees)
```



Гистограммы:

```
> hist(salary2, breaks = 20)
```

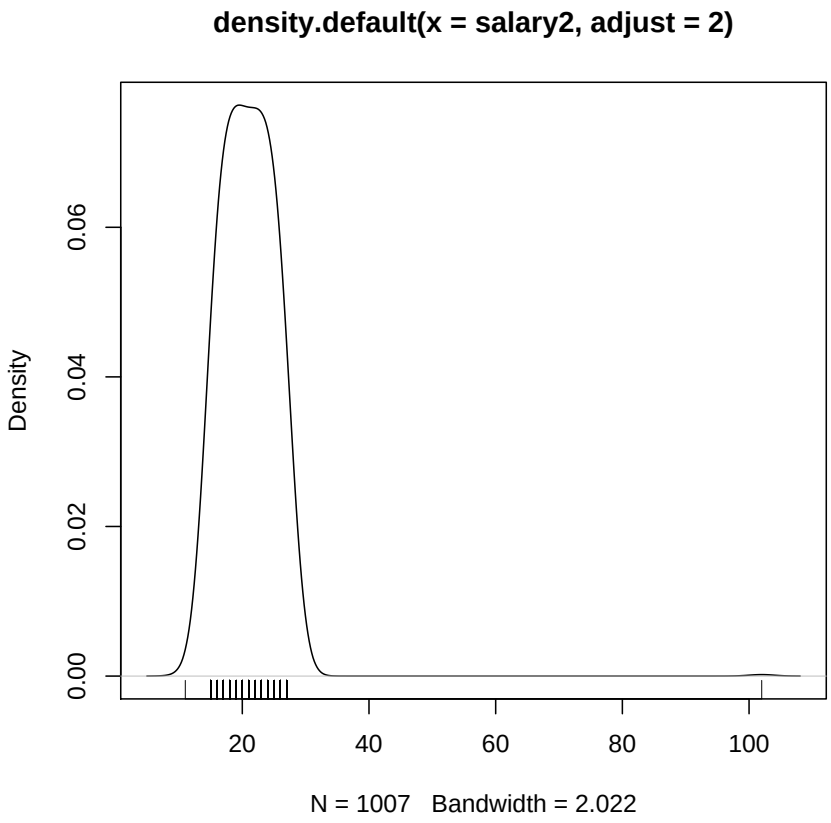


Текстовое представление гистограммы:

```
> table(cut(salary2, 20))

(10.9,15.5]   (15.5,20]   (20,24.6] (24.6,29.1] (29.1,33.7] (33.7,38.3]
         87         388         311         220           0           0
(38.3,42.8] (42.8,47.4] (47.4,51.9] (51.9,56.5] (56.5,61.1] (61.1,65.6]
          0           0           0           0           0           0
(65.6,70.2] (70.2,74.7] (74.7,79.3] (79.3,83.9] (83.9,88.4]   (88.4,93]
          0           0           0           0           0           0
  (93,97.5]  (97.5,102]
          0           1

> stem(salary, scale = 2)

  The decimal point is 1 digit(s) to the right of the |

   1 | 19
   2 | 1157
   3 |
   4 |
   5 |
   6 |
   7 |
   8 |
   9 |
  10 | 2

NULL
```

Сглаженная гистограмма:

```
> plot(density(salary2, adjust = 2))
> rug(salary2)
```



Самая главная функция для описания базовой статистики:

```
> summary(trees)
```

```
     Girth           Height       Volume
 Min.   : 8.30   Min.   :63   Min.   :10.20
 1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
 Median :12.90   Median :76   Median :24.20
 Mean   :13.25   Mean   :76   Mean   :30.17
 3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
 Max.   :20.60   Max.   :87   Max.   :77.00
```

```
> lapply(list(salary, salary2), summary)
```

```
[[1]]
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
  11.00   20.00   21.00   32.29   26.00  102.00

[[2]]
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  11.00   18.00   21.00   20.97   24.00  102.00
```

```
> summary(attenu)
```

```
     event            mag           station         dist
 Min.   : 1.00   Min.   :5.000   117     :  5   Min.   :  0.50
 1st Qu.: 9.00   1st Qu.:5.300   1028    :  4   1st Qu.: 11.32
 Median :18.00   Median :6.100   113     :  4   Median : 23.40
 Mean   :14.74   Mean   :6.084   112     :  3   Mean   : 45.60
 3rd Qu.:20.00   3rd Qu.:6.600   135     :  3   3rd Qu.: 47.55
 Max.   :23.00   Max.   :7.700   (Other):147   Max.   :370.00
                                 NA's    : 16
     accel
 Min.   :0.00300
 1st Qu.:0.04425
 Median :0.11300
 Mean   :0.15422
 3rd Qu.:0.21925
 Max.   :0.81000
```

```
> methods(summary)
```

```
 [1] summary.aov            summary.aovlist        summary.connection
 [4] summary.data.frame     summary.Date           summary.default
 [7] summary.ecdf*          summary.factor         summary.glm
[10] summary.infl           summary.lm             summary.loess*
[13] summary.manova         summary.matrix         summary.mlm
[16] summary.nls*           summary.packageStatus* summary.POSIXct
[19] summary.POSIXlt        summary.ppr*           summary.prcomp*
[22] summary.princomp*      summary.stepfun        summary.stl*
[25] summary.table          summary.tukeysmooth*

   Non-visible functions are asterisked
```

2 Одномерные статистические тесты

Тест Стьюдента:

```
> t.test(salary, mu = 32)

        One Sample t-test

data:  salary
t = 0.0243, df = 6, p-value = 0.9814
alternative hypothesis: true mean is not equal to 32
95 percent confidence interval:
  3.468127 61.103302
sample estimates:
mean of x
 32.28571
```

Ранговый тест Уилкоксона (Wilcoxon signed-rank test):

```
> wilcox.test(salary2, mu = median(salary2), conf.int = TRUE)

        Wilcoxon signed rank test with continuity correction

data:  salary2
V = 211035, p-value = 0.3752
alternative hypothesis: true location is not equal to 21
95 percent confidence interval:
 20.50008 21.00001
sample estimates:
(pseudo)median
      20.99997
```

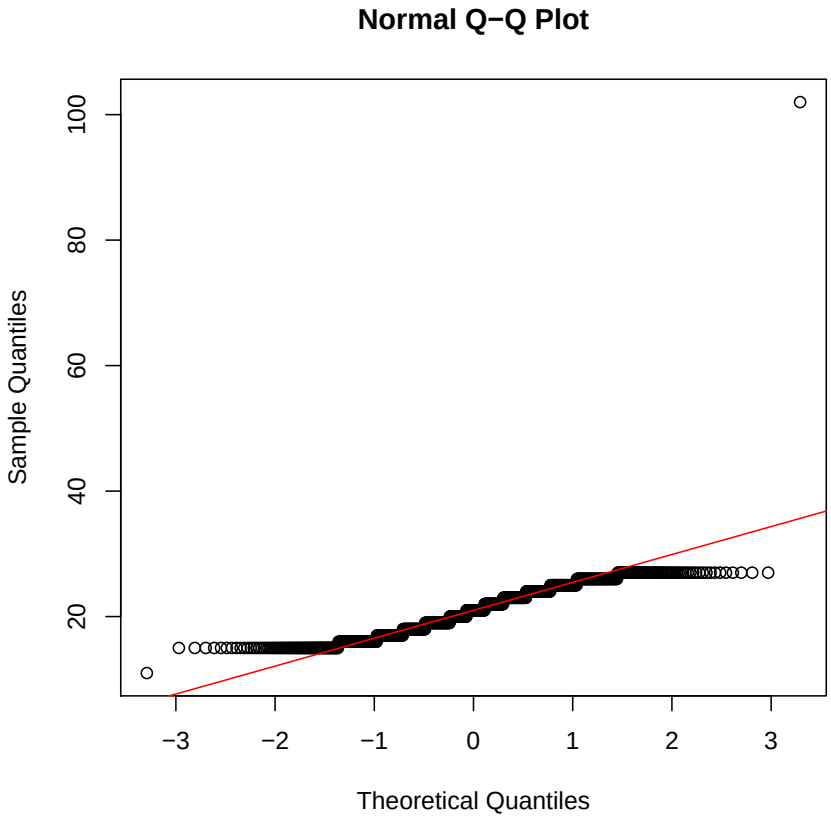
Тест Шапиро-Уилкса (Shapiro-Wilk test):

```
> shapiro.test(salary)

        Shapiro-Wilk normality test

data:  salary
W = 0.6116, p-value = 0.0003726

> shapiro.test(salary2)

        Shapiro-Wilk normality test

data:  salary2
W = 0.7422, p-value < 2.2e-16
```

```
> set.seed(1638)

NULL

> shapiro.test(rnorm(100))

        Shapiro-Wilk normality test

data:  rnorm(100)
W = 0.9934, p-value = 0.9094
```

Графическая проверка выборки на нормальность:

```
> qqnorm(salary2)
> qqline(salary2, col = 2)
```



3 Как создавать свои функции

Пример пользовательской функции, векторизующей тест Шапиро-Уилкса:

```
> normality <- function(data.f) {
+     result <- data.frame(var = names(data.f), p.value = rep(0,
+         ncol(data.f)), normality = is.numeric(names(data.f)))
+     for (i in 1:ncol(data.f)) {
+         data.sh <- shapiro.test(data.f[, i])$p.value
+         result[i, 2] <- round(data.sh, 5)
+         result[i, 3] <- (data.sh > 0.05)
+     }
+     return(result)
+ }

function(data.f)
{
 result <- data.frame(var=names(data.f),
           p.value=rep(0, ncol(data.f)),
           normality=is.numeric(names(data.f)))
 for (i in 1:ncol(data.f))
  {
   data.sh <- shapiro.test(data.f[, i])$p.value
   result[i, 2] <- round(data.sh, 5)
   result[i, 3] <- (data.sh > .05)
  }
 return(result)
}

> normality(trees)

     var p.value normality
1  Girth 0.08893      TRUE
2 Height 0.40341      TRUE
3 Volume 0.00358     FALSE
```

Ещё один пример:

```
> normality2 <- function(data.f, p = 0.05) {
+     nn <- ncol(data.f)
+     result <- data.frame(var = names(data.f), p.value = numeric(nn),
+         normality = logical(nn))
+     for (i in 1:nn) {
+         data.sh <- shapiro.test(data.f[, i])$p.value
+         result[i, 2:3] <- list(round(data.sh, 5), data.sh > p)
```

```
+     }
+     return(result)
+ }

function(data.f, p=.05)
{
 nn <- ncol(data.f)
 result <- data.frame(var=names(data.f),p.value=numeric(nn),
 normality=logical(nn))
 for (i in 1:nn)
  {
   data.sh <- shapiro.test(data.f[, i])$p.value
   result[i, 2:3] <- list(round(data.sh, 5), data.sh > p)
  }
 return(result)
}

> normality2(trees)

     var p.value normality
1  Girth 0.08893      TRUE
2 Height 0.40341      TRUE
3 Volume 0.00358     FALSE

> normality2(trees, 0.1)

     var p.value normality
1  Girth 0.08893     FALSE
2 Height 0.40341      TRUE
3 Volume 0.00358     FALSE
```

Способ избежать циклы:

```
> lapply(trees, shapiro.test)

$Girth

        Shapiro-Wilk normality test

data:  X[[1]]
W = 0.9412, p-value = 0.08893


$Height
```

```
        Shapiro-Wilk normality test

data:  X[[2]]
W = 0.9655, p-value = 0.4034


$Volume

        Shapiro-Wilk normality test

data:  X[[3]]
W = 0.8876, p-value = 0.003579

> lapply(trees, function(.x) ifelse(shapiro.test(.x)$p.value >
+     0.05, "NORMAL", "NOT NORMAL"))

$Girth
[1] "NORMAL"

$Height
[1] "NORMAL"

$Volume
[1] "NOT NORMAL"

> normality3 <- function(df, p = 0.05) {
+     lapply(df, function(.x) ifelse(shapiro.test(.x)$p.value >
+         p, "NORMAL", "NOT NORMAL"))
+ }

function(df, p=.05)
{
 lapply(df, function(.x)
       ifelse(shapiro.test(.x)$p.value > p, "NORMAL","NOT NORMAL"))
}

> normality3(list(salary, salary2))

[[1]]
[1] "NOT NORMAL"

[[2]]
[1] "NOT NORMAL"
```

Ответ на вопрос

```
> str(shapiro.test(rnorm(100)))

List of 4
 $ statistic: Named num 0.992
  ..- attr(*, "names")= chr "W"
 $ p.value  : num 0.854
 $ method   : chr "Shapiro-Wilk normality test"
 $ data.name: chr "rnorm(100)"
 - attr(*, "class")= chr "htest"
NULL
```