

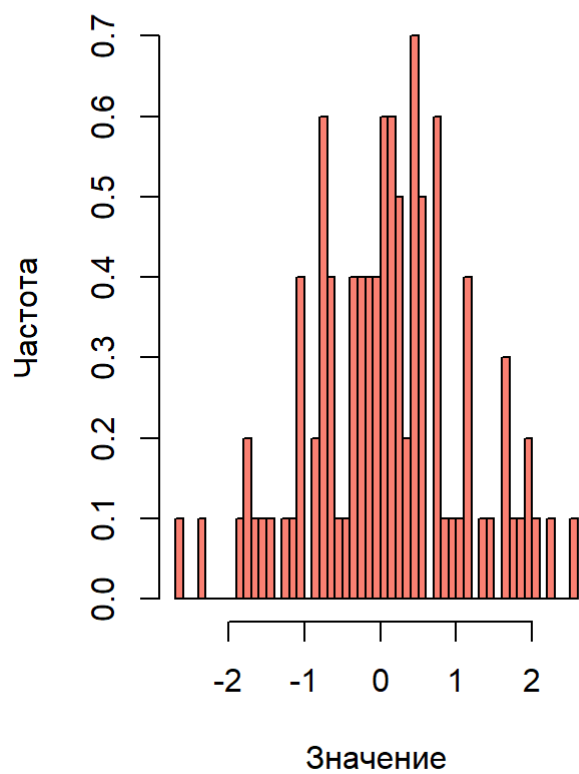
# Харинаев Артём 316

## 1. Задания из комментариев

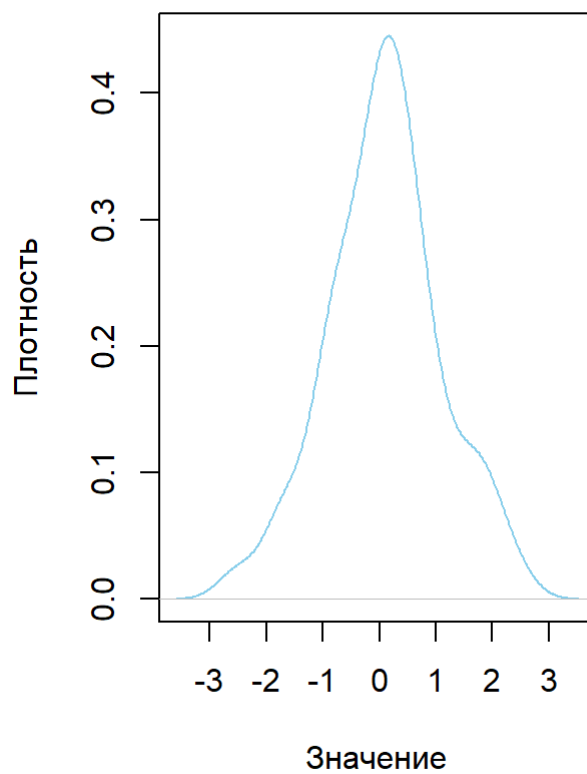
### 1.1 Вариация размера нормально распределенной выборки

```
n_range <- c(seq(100, 1000, 300), seq(2000, 10000, 4000))
for (n in n_range){
  par(mfrow=c(1,2))
  x <- rnorm(n = n, mean = 0, sd = 1)
  hist(x, xlab='Значение', ylab='Частота', main=n, freq=FALSE, breaks=40, col='salmon')
  plot(density(x), xlab='Значение', ylab='Плотность', main=n, col='skyblue')
}
```

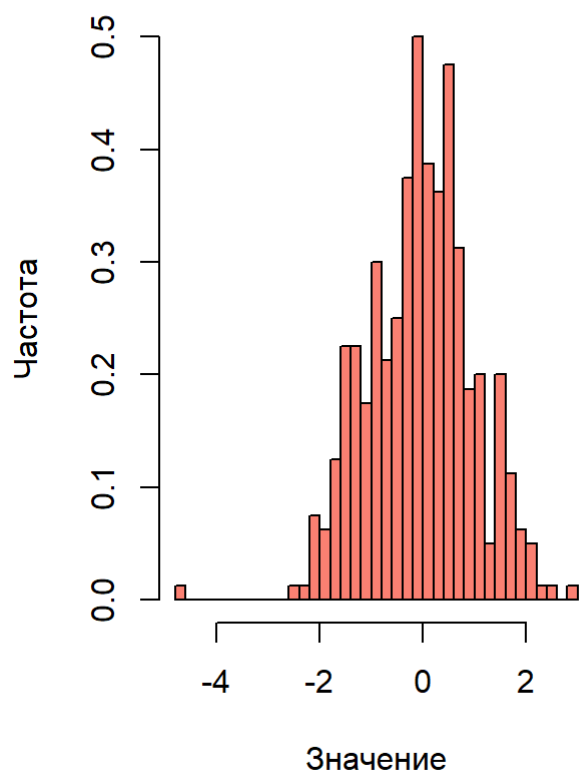
100



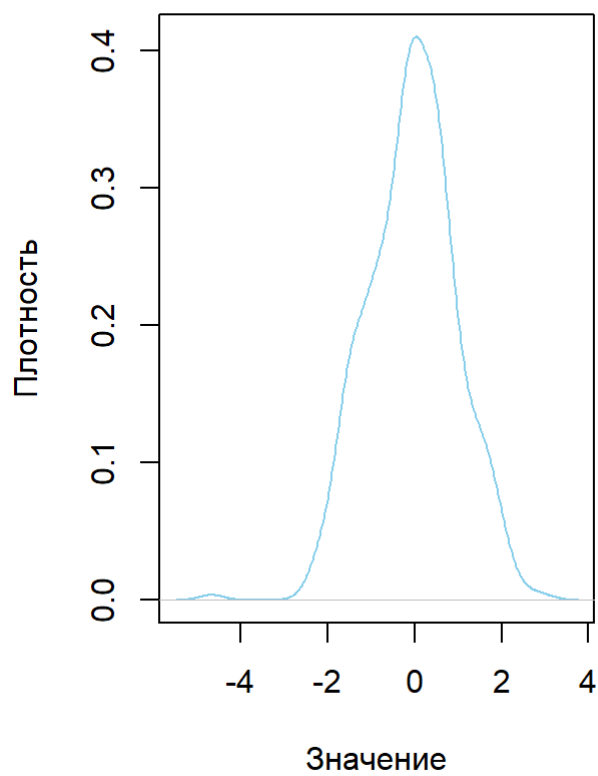
100



400

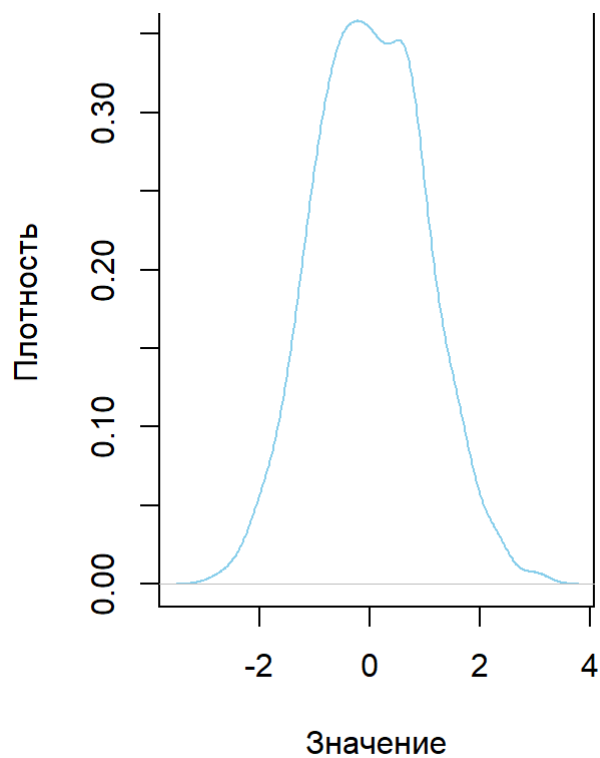
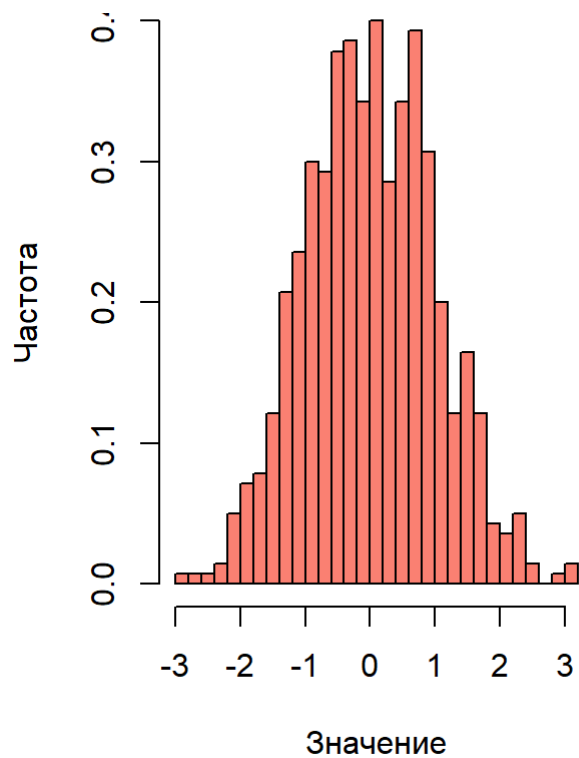


400

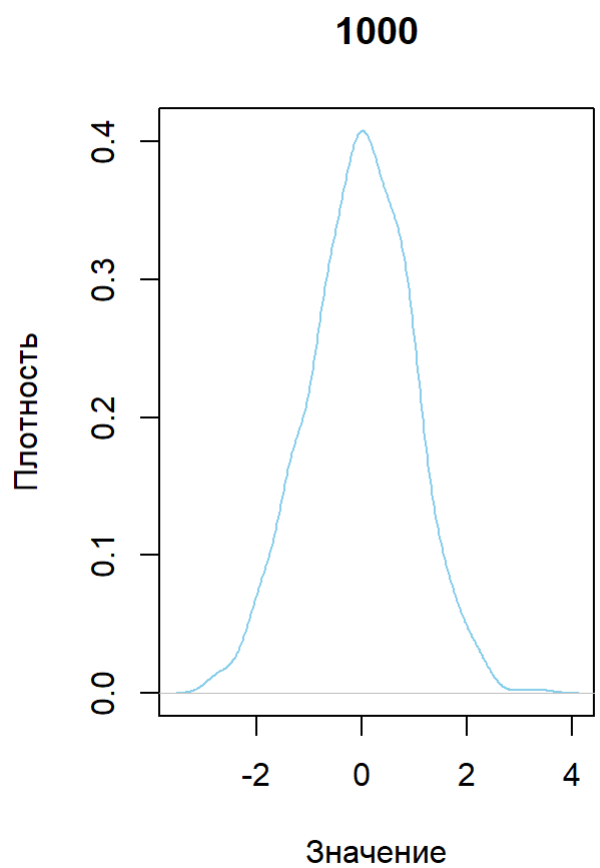
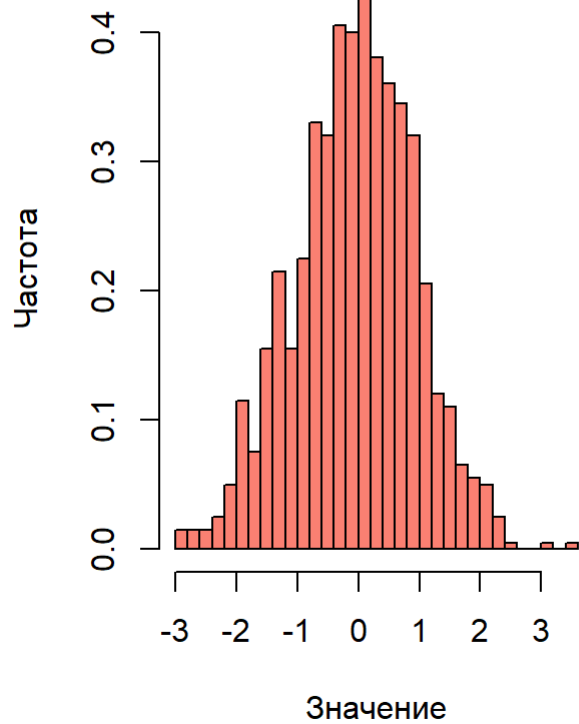


700

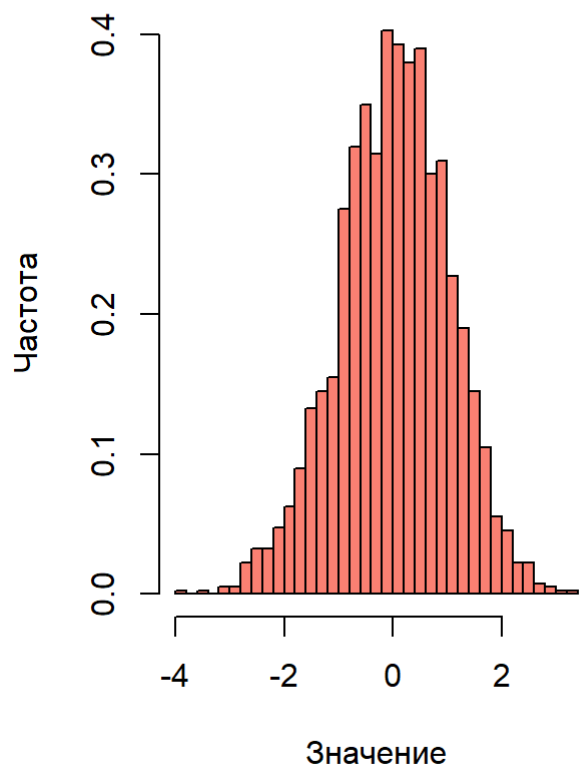
700



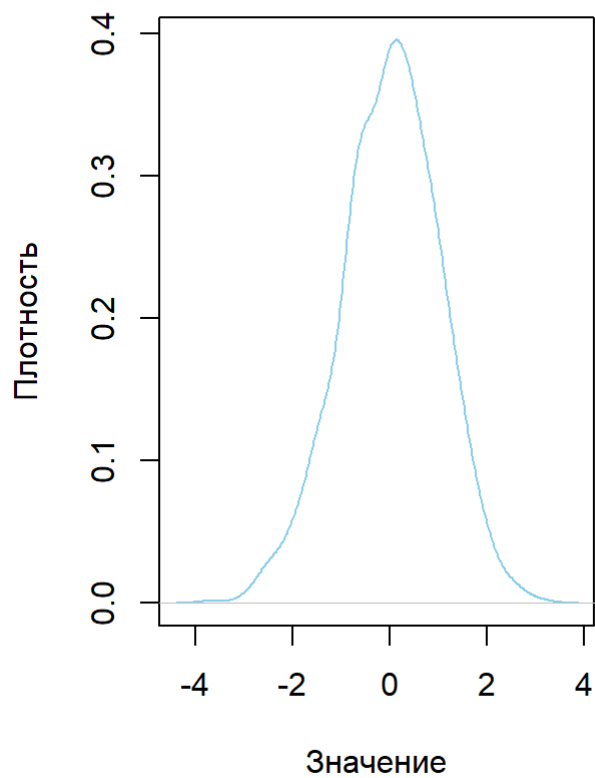
1000



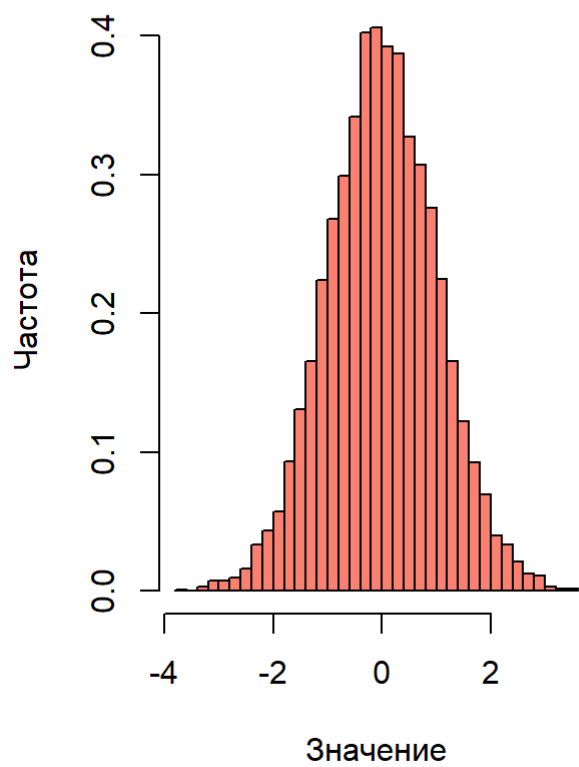
**2000**



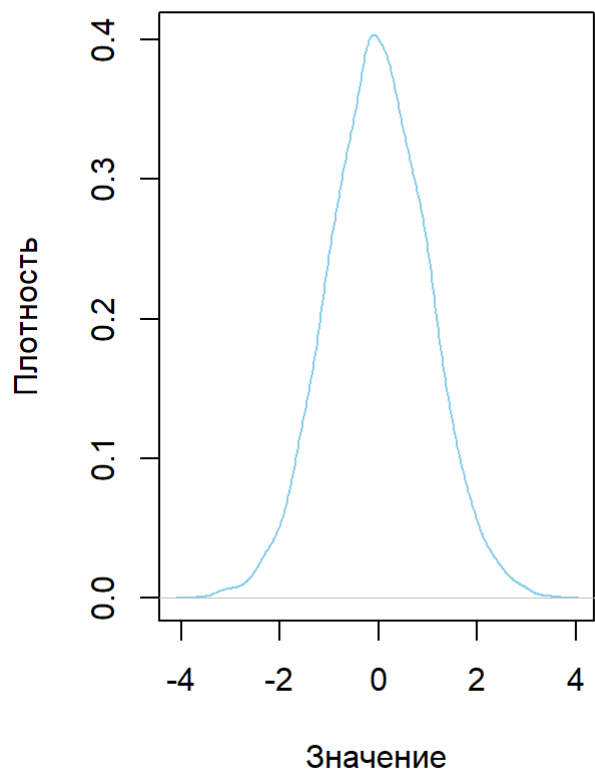
**2000**

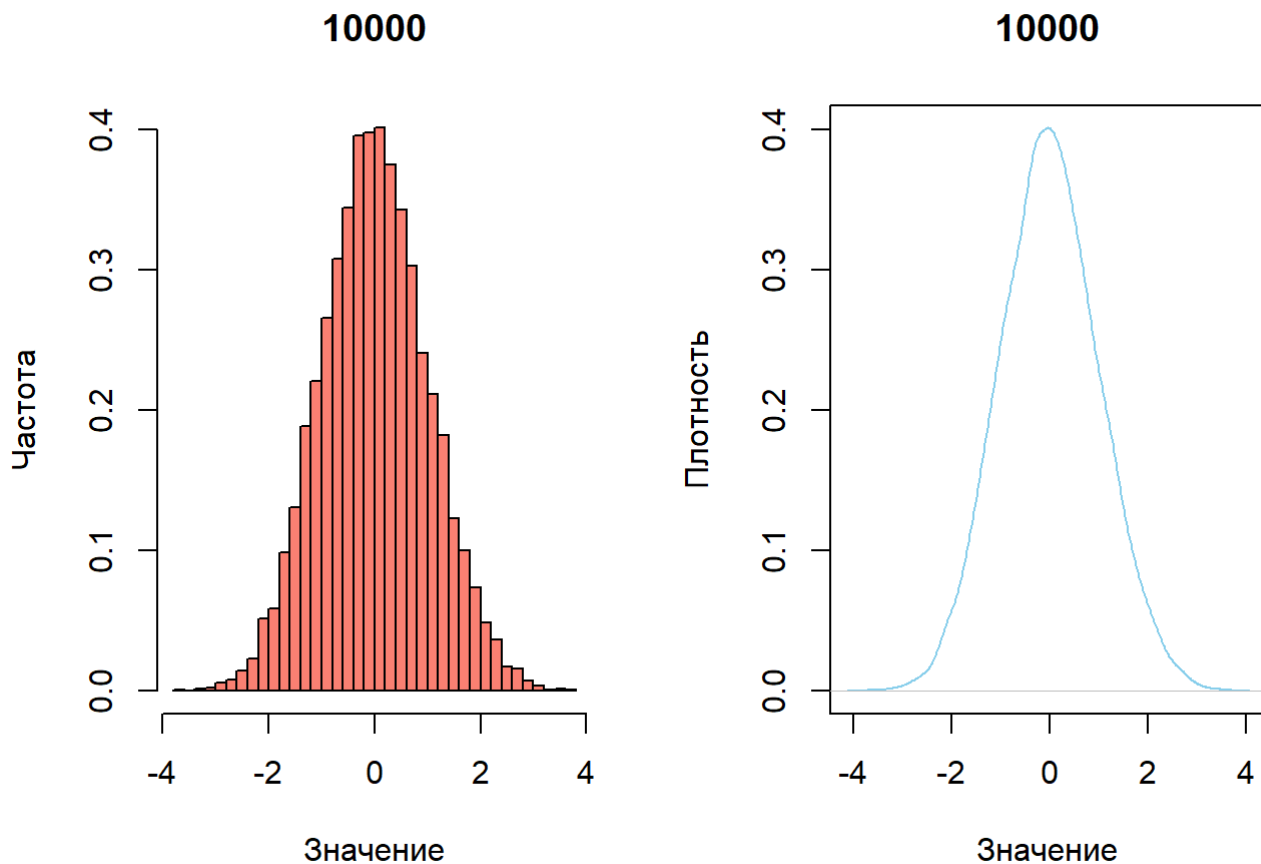


**6000**



**6000**





Из гистограмм и графиков плотностей видно, что с увеличением размера выборки эмпирическое распределение все больше и больше приближается к теоретическому нормальному распределению

## 1.2. density()

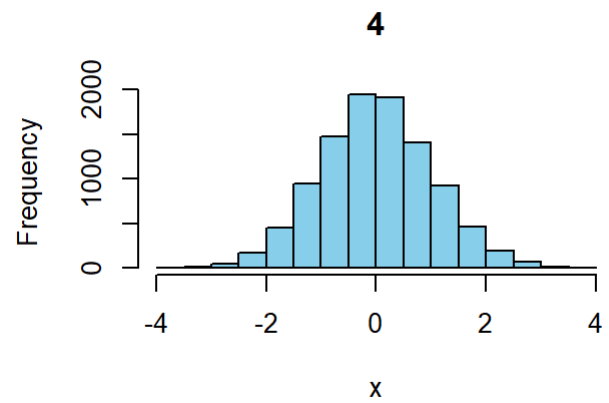
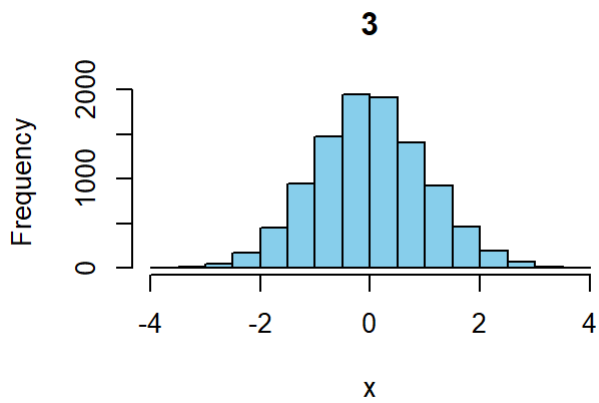
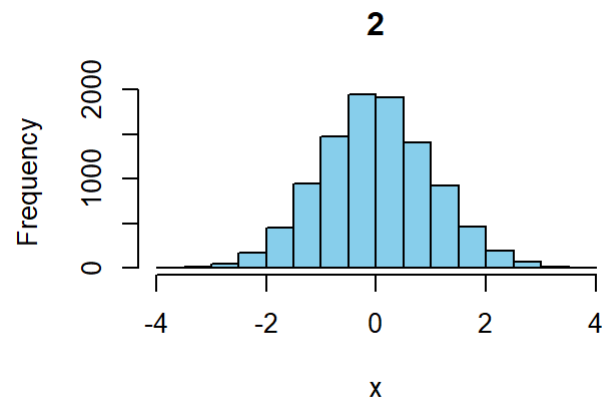
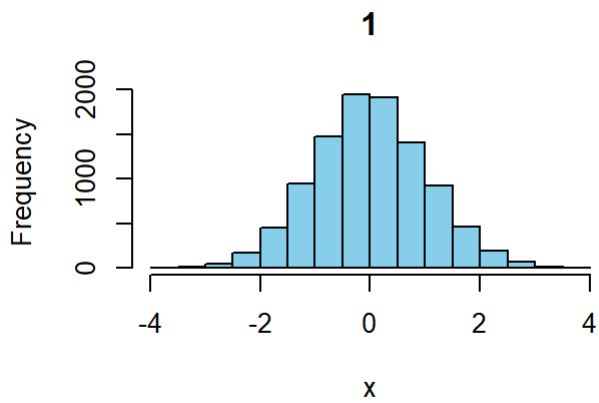
Функция строит ядерную оценку плотности

Параметры:

1. **x** - выборка, для которой строится оценка (численный вектор)
2. **bw** - ширина окна, ядро подбирается таким образом, чтобы отклонение равнялось этому параметру
3. **adjust** - число, на которое умножается bw
4. **kernel** - ядро, на основе которого строится оценка (функция)
5. **weights** - вектор весов для элементов выборки той же длины, что и x
6. **give.Rkern** - если TRUE, то оценка не строится, возвращаемое значение функции - ширина окна для выбранного ядра
7. **n** - число равномерно распределенных точек, в которых оценивается плотность
8. **from, to** - левая и правая границы выборки, на которой нужно оценить плотность

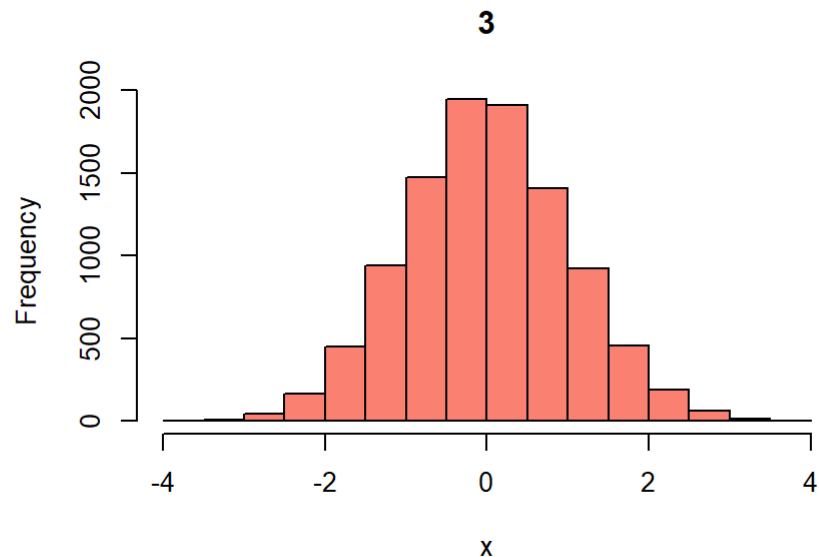
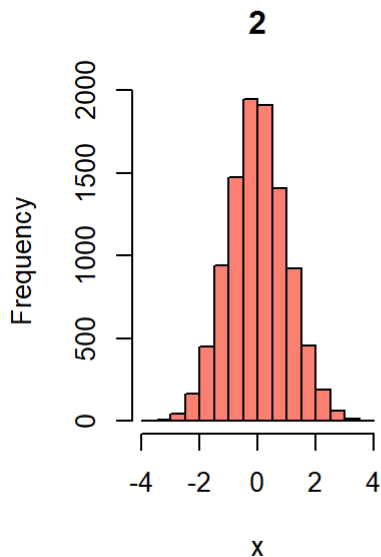
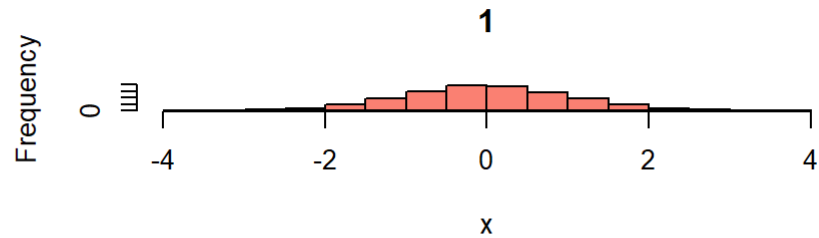
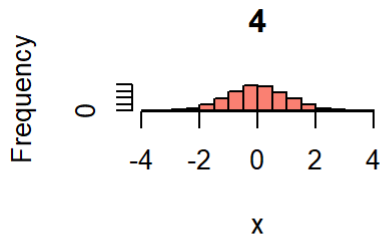
## 1.3. layout и par(mfrow)

```
par(mfrow=c(2,2))
for (i in 1:4){
  hist(x, main=i, col='skyblue')
}
```



для **layout** можно задать порядок вывода, высоту и ширину графиков

```
layout(matrix(c(4,2,1,3), ncol=2), widths = c(1,2), heights = c(1,2))
for (i in 1:4){
  hist(x, main=i, col='salmon')
}
```



## 1.4 boxplot()

Построение диаграммы размаха

1. **formula** - в виде  $y \sim grp$ , где  $y$  - численный вектор, который необходимо разбить на группы по переменной  $grp$
2. **data** - *data.frame* или *list*, откуда берутся переменные для *formula*
3. **subset** - численный вектор, задающийся, если график нужно построить по подмножеству данных
4. **na.action** - действие, совершающееся при встрече NA значения
5. **xlab, ylab** - подписи к осям
6. **range** - определяет, насколько далеко могут выходить "усы" из "ящика"
7. **width** - ширина ящиков
8. **varwidth** - если TRUE, то ширина ящиков пропорциональна квадратному корню из числа входящих наблюдений
9. **names** - имена, которые будут подписаны под "ящиками"

## 1.5 barplot()

Построение столбчатой диаграммы

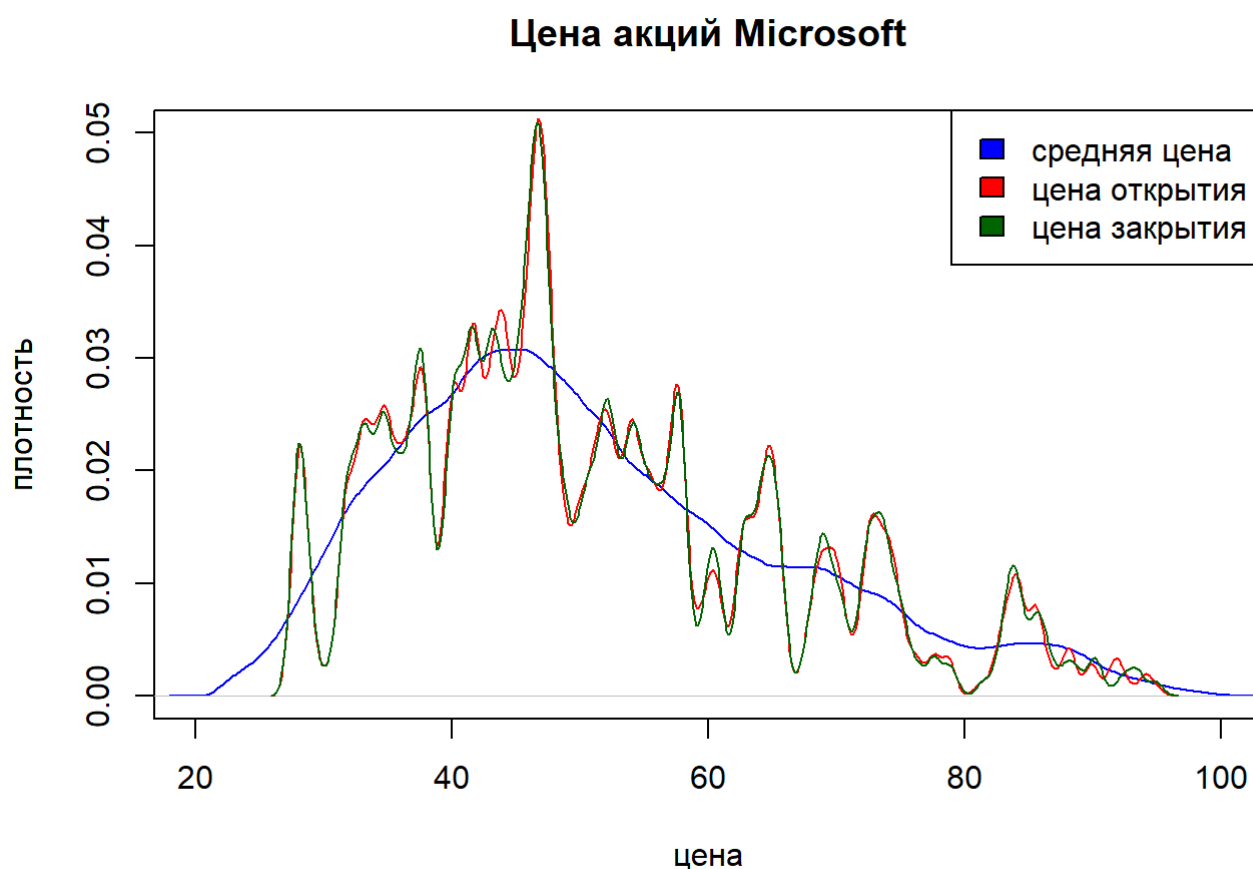
1. **height** - значения, отвечающие за высоту столбцов диаграммы, матрица или вектор
2. **width** - ширина столбцов, одно число или вектор
3. **space** - размер отступа между столбцами, одно число или вектор
4. **names.arg** - имена, подписанные под столбцами, вектор строк
5. **legend.text** - вектор описаний для построения легенды
6. **beside** - если TRUE, то столбцы, относящиеся к одной группе будут расположены рядом друг с другом, если FALSE, то друг на друге (параметр актуален только в случае матрицы в параметре *height*)

7. **horiz** - если TRUE, то столбцы будут расположены горизонтально
8. **density** - плотность линий штриховки столбцов, число или вектор
9. **angle** - угол линий штриховки
10. **col** - цвета столбцов
11. **border** - цвета границы столбцов
12. **formula** - в виде  $y \sim grp$ , где  $y$  - численный вектор, который необходимо разбить на группы по переменной  $grp$
13. **data** - *data.frame* или *list*, откуда берутся переменные для *formula*
14. **subset** - - численный вектор, задающийся, если график нужно построить по подмножеству данных
15. **na.action** - действие, совершающееся при встрече NA значения

## 2. Ядерная аппроксимация данных

```
data <- read.csv(file = '..\\dataset.csv')
```

```
msft <- subset(data, data['Name'] == 'MSFT')
msft$mean <- (msft$open + msft$close)/2
my_plot <- plot(density(msft$mean, kernel='epanechnikov'), main='Цена акций Microsoft', col=
'blue', xlab = 'цена', ylab = 'плотность', xlim=c(20,100), ylim=c(0, 0.05))
lines(density(msft$close, bw=0.5), col='red')
lines(density(msft$open, bw=0.5), col='darkgreen')
legend('topright', legend=c('средняя цена', 'цена открытия', 'цена закрытия'), fill=c('blue',
'red', 'darkgreen'))
```



Из графика видно, что фазы колебаний цены открытия и цены закрытия на больших амплитудах (больших изменениях) совпадают, но при незначительных колебаниях могут расходиться

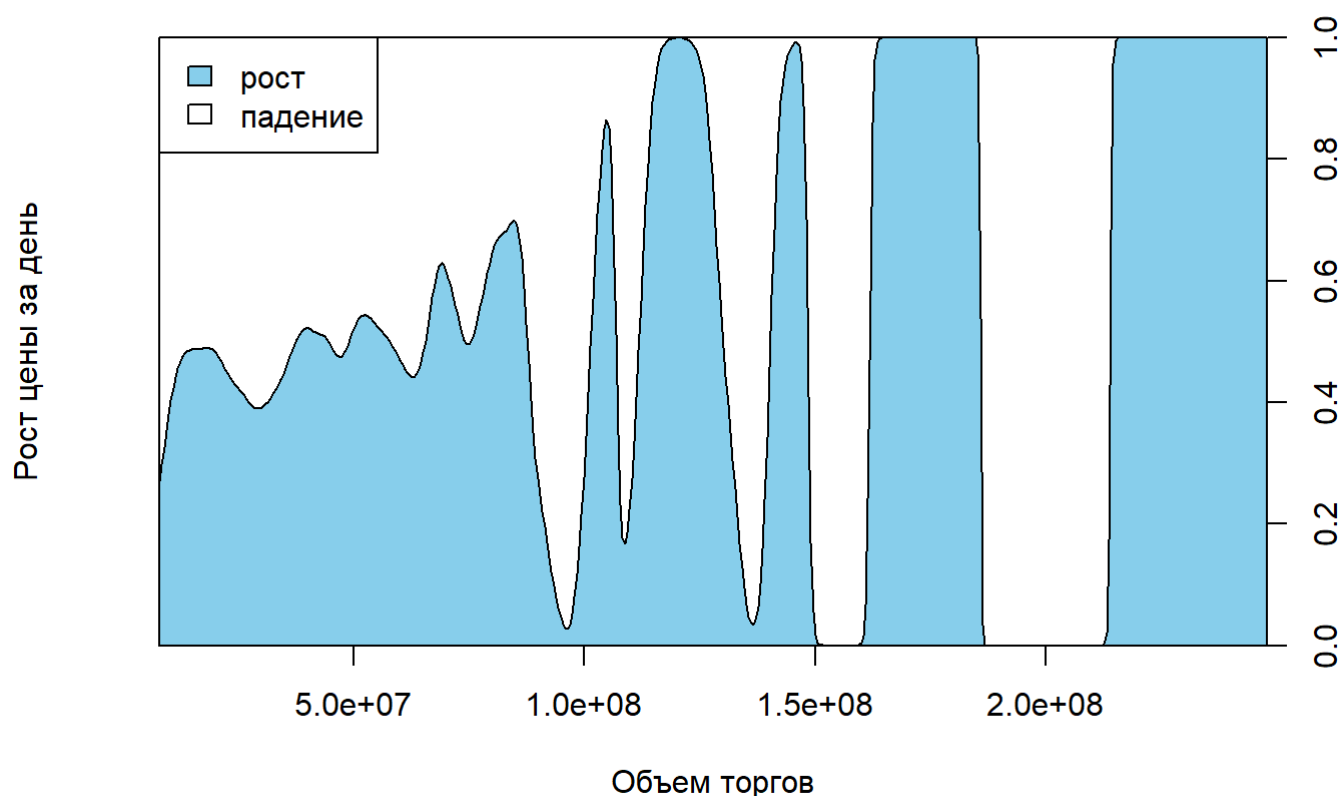


## 3. cdplot() и boxplot()

### 3.1 cdplot()

Исследуем зависимость объема торгов от разницы цен открытия и закрытия (роста или падения цены)

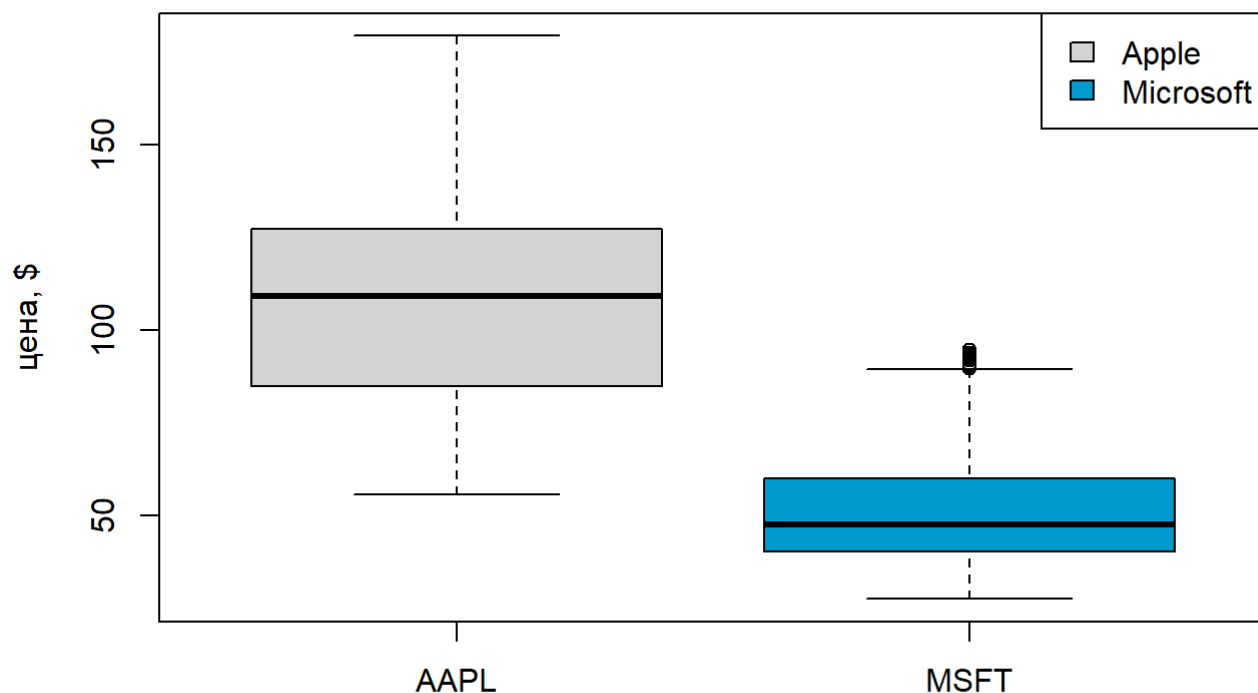
```
msft$date <- as.Date(msft$date)
msft$factor <- factor((msft$open - msft$close) > 0)
cdplot(msft$volume, msft$factor, col=c('skyblue', 'white'), bw='SJ', ylab='Рост цены за день',
, xlab='Объем торгов', yaxlabels = c('', ''))
legend('topleft', legend=c('рост', 'падение'), fill=c('skyblue', 'white'))
```



Из графика можно заключить, что сделок в дни роста совершается больше

### 3.2 boxplot

```
data$mean <- (data$open + data$close)/2
apple <- subset(data, data['Name'] == 'AAPL')
msft <- subset(data, data['Name'] == 'MSFT')
boxplot(apple$mean, msft$mean, col=c('lightgrey', 'deepskyblue3'), names=c('AAPL', 'MSFT'), y
lab = 'цена, $')
legend('topright', legend=c('Apple', 'Microsoft'), fill=c('lightgrey', 'deepskyblue3'))
```



## 4. pie()

```
data$date <- as.Date(data$date)
day <- subset(data, data['date']=='2015-01-09')
day <- day[order(-day$volume),] #сортируем по объему продаж
sub_day <- day[1:10,]
plt <- palette.colors(10, palette='Tableau')
pie(sub_day$volume, labels=sub_day$Name, clockwise=TRUE, init.angle = 180, col=plt, radius=0.5, main='Топ-10 компаний по объему продаж акций 01.09.2015')
names <- c('Bank of America', 'Apple', 'General electric', 'Cisco', 'Starbucks', 'Pfizer', 'Microsoft', 'Ford', 'Intel', 'Comcast')
legend('right', legend=names, fill=plt)
```

## Топ-10 компаний по объему продаж акций 01.09.2015

