

Практикум на ЭВМ, отчёт по заданию 4
Роор Даниил, группа 316

Далее текст программы будет написан моноширинным шрифтом FreeMono, перемежаясь с комментариями, написанными немоноширинным шрифтом, и построенными графиками

```
library(dplyr)

#FUNCTIONS
#####
filterByColumnValue = function (df, column, value) {
  dplyr::filter(df, !!as.symbol(column)==value)
  # !! discards quotes
  # from column string
}

df = read.csv('COVID-19-in-Italy.csv')
```

ЗАДАНИЕ 1. В примере с гистограммой и плотностью воспользоваться функцией legend для отображения подписей к графикам ядерных оценок плотностей.

```
X <- rnorm(n = 100, mean = 15, sd = 5)

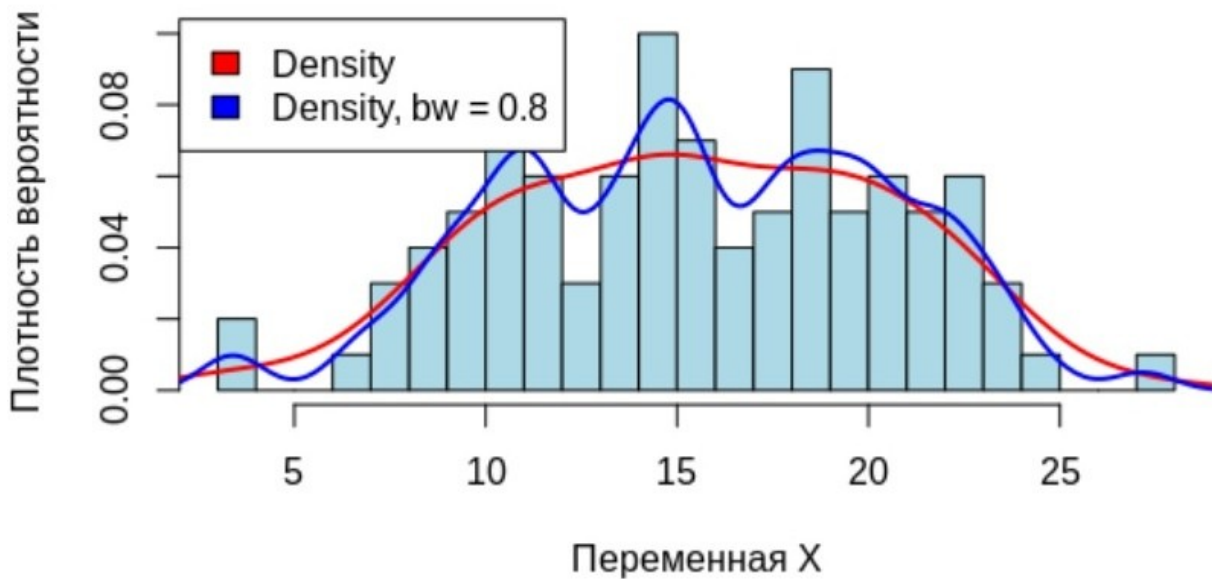
hist(X, breaks = 20, freq = FALSE, col = "lightblue",
      xlab = "Переменная X",
      ylab = "Плотность вероятности",
      main = "Гистограмма, совмещенная с кривой плотности")

sign = c('Density', 'Density, bw = 0.8')
Colfill <- c('red', 'blue') #2:(2 + length(sign))

lines(density(X), col = "red", lwd = 2)
lines(density(X, bw = 0.8), col = "blue", lwd = 2)

legend('topleft', sign, fill = Colfill)
```

Гистограмма, совмещенная с кривой плотности



ЗАДАНИЕ 2. Реализовать аппроксимацию с помощью ядерных оценок для своих данных.

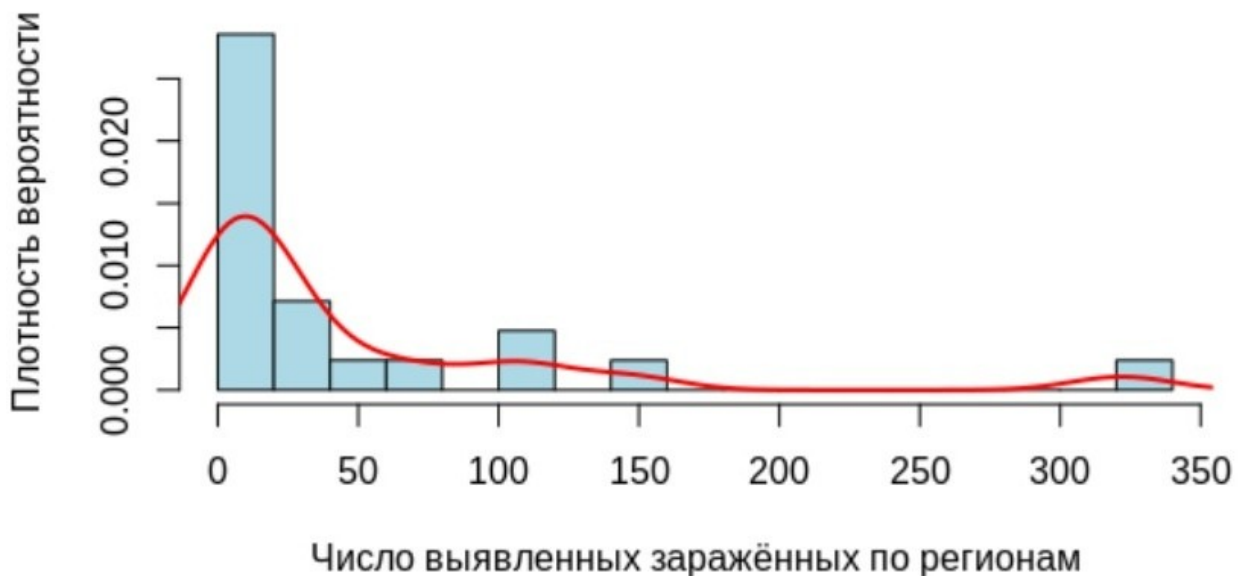
Найдем ядерную оценку плотности числа зараженных 10 марта. Выборкой являются значения этого числа в разных регионах

```
march10 = filterByColumnValue(df, 'date', '2020-03-10 18:00:00')
infected = march10$new_positive

hist(infected, breaks = 20, freq = FALSE, col = "lightblue",
     xlab = "Число выявленных заражённых по регионам",
     ylab = "Плотность вероятности",
     main = "Гистограмма, совмещенная с кривой плотности")

lines(density(infected), col = "red", lwd = 2)
```

Гистограмма, совмещенная с кривой плотности



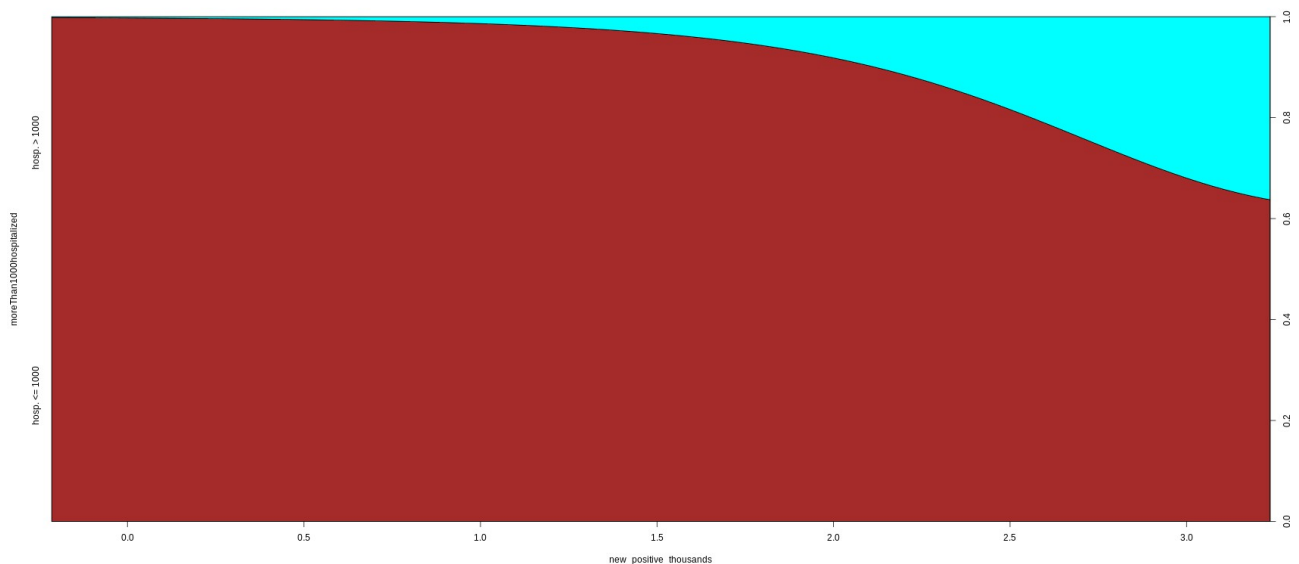
ЗАДАНИЕ 3. На примере собственных данных продемонстрировать применение `cdplot` и `boxplot`.

Для `cdplot` создадим новую категориальную переменную - флаг, показывающий, превышает ли количество госпитализированных людей 1000 человек. График будет для значений числа выявленных больных изображать вероятность равенства категориальной переменной тому или иному значению (то есть мы исследуем связь между числом больных и госпитализированных)

```
df$moreThan1000hospitalized = df$hospitalized_with_symptoms >
10000
df$moreThan1000hospitalized = factor(df$moreThan1000hospitalized)

df$new_positive_thousands = df$new_positive / 1000

png(file = 'cdplot.png', width = 1700, height = 800)
cdplot(moreThan1000hospitalized ~ new_positive_thousands, col =
c("brown", "cyan"),
      yaxlabels = c('hosp. <= 1000', 'hosp. > 1000'), bw = 0.8,
data = df)
dev.off()
```



На boxplot изобразим распределение числа заражённых за сутки по регионам. Категориальной переменной для boxplot будет регион Италии.

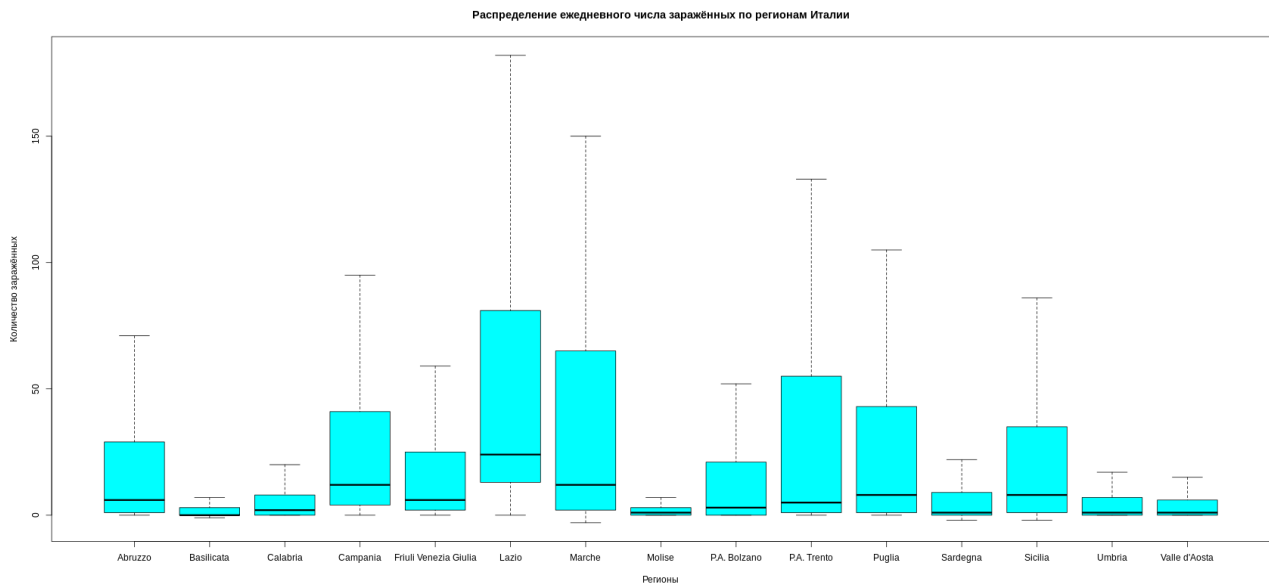
В некоторых регионах слишком много зараженных по сравнению с другими регионами. Это портит график, сравнить между собой остальные регионы из-за этого труднее, так что не будем их рассматривать.

```
ejected = df[df$region != 'Lombardia' & df$region != 'Piemonte'
             & df$region != 'Emilia-Romagna' & df$region !=
             'Veneto'
             & df$region != 'Liguria' & df$region != 'Toscana', ]
```

Убираем не встречающиеся (после отбрасывания выше) уровни категориальной переменной region, чтобы на графике не было регионов, данные по которым мы удалили.

```
ejected$region = droplevels(ejected$region)

png(file = 'boxplot.png', width = 1700, height = 800)
boxplot(new_positive ~ region,
        xlab = "Регионы",
        ylab = "Количество заражённых",
        main = "Распределение числа заражённых за сутки по
регионам Италии",
        col = "cyan", data = ejected,
        outline = FALSE) #убираем выбросы с графика
dev.off()
```



ЗАДАНИЕ 4. Используя функцию `pie`, нарисовать пример круговой диаграммы для собственных данных.

Нарисуем круговую диаграмму для суммарного количества случаев заболевания по регионам по данным на 29.07.2020

```
july29 = df[df$region != 'Lombardia'
            & df$region != 'Sardegna'
            & df$date == "2020-07-29 17:00:00",]
```

```
cases = july29$total_cases
```

Напишем для каждого региона количество случаев и число процентов от общего числа случаев

```
lbl = sprintf("%s (%d, %.1f%s)", july29$region, cases, 100 *
cases/sum(cases), "%")
```

```
png(file = 'pieplot.png', width = 1700, height = 800)
pie(cases, labels = lbl, radius = 1,
    main = 'Суммарное число заражённых по регионам Италии
(29.07.2020)')
dev.off()
```

Суммарное число заражённых по регионам Италии (29.07.2020)

