

lab1

Лабораторная работа 1

Исходные данные

Экспоненциальное распределение определяется следующей функцией распределения:

$$F(x) = 1 - e^{(-\lambda \cdot x)}$$

Функция `rexp` генерирует случайные переменные при помощи экспоненциального распределения `rexp(n=42, rate=5)` вернёт 42 экспоненциально-распределённых числа с параметром (λ) равным 5. Если второй аргумент не указан, будет использовано значение 1 т.е. получим “стандартное экспоненциальное распределение”.

Задание 1

1. Сгенерируйте 200 случайных значений из стандартного экспоненциального распределения и сохраните в вектор `exp.1`. Найдите среднее и стандартное отклонение из данных этого вектора.

```
exp.1 <- rexp(200)
exp.1.mean <- mean(exp.1)
exp.1.sd <- sd(exp.1)
```

2. Повторите, используя следующие значения параметра λ : 0.1, 0.5, 5, 10, и сохраните результаты в векторы: `exp.0.1`, `exp.0.5`, `exp.5`, `exp.10`.

```
exp.0.1 <- rexp(200, 0.1)
exp.0.5 <- rexp(200, 0.5)
exp.5 <- rexp(200, 5)
exp.10 <- rexp(200, 10)

exp.0.1.mean <- mean(exp.0.1)
exp.0.5.mean <- mean(exp.0.5)
exp.5.mean <- mean(exp.5)
exp.10.mean <- mean(exp.10)

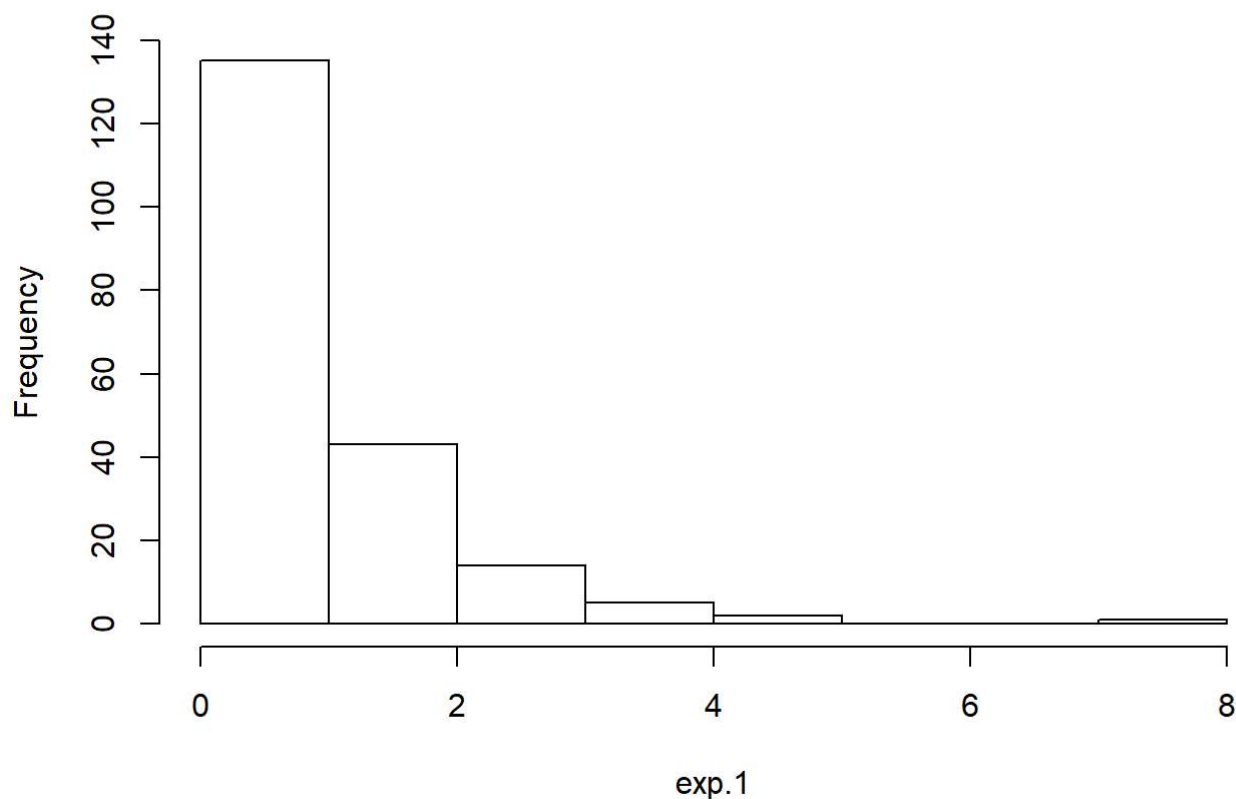
exp.0.1.sd <- sd(exp.0.1)
exp.0.5.sd <- sd(exp.0.5)
exp.5.sd <- sd(exp.5)
exp.10.sd <- sd(exp.10)
```

3. Функция `plot()` — общая функция для визуализации данных. Функция `hist()` принимает данные и раскладывает их по корзинам. Чтобы использовать эти функции нужно для начала передать им данные, которые хотим визуализировать.

4. Используйте функцию `hist()` для построения гистограммы полученного ранее стандартного распределения.

```
hist(exp.1)
```

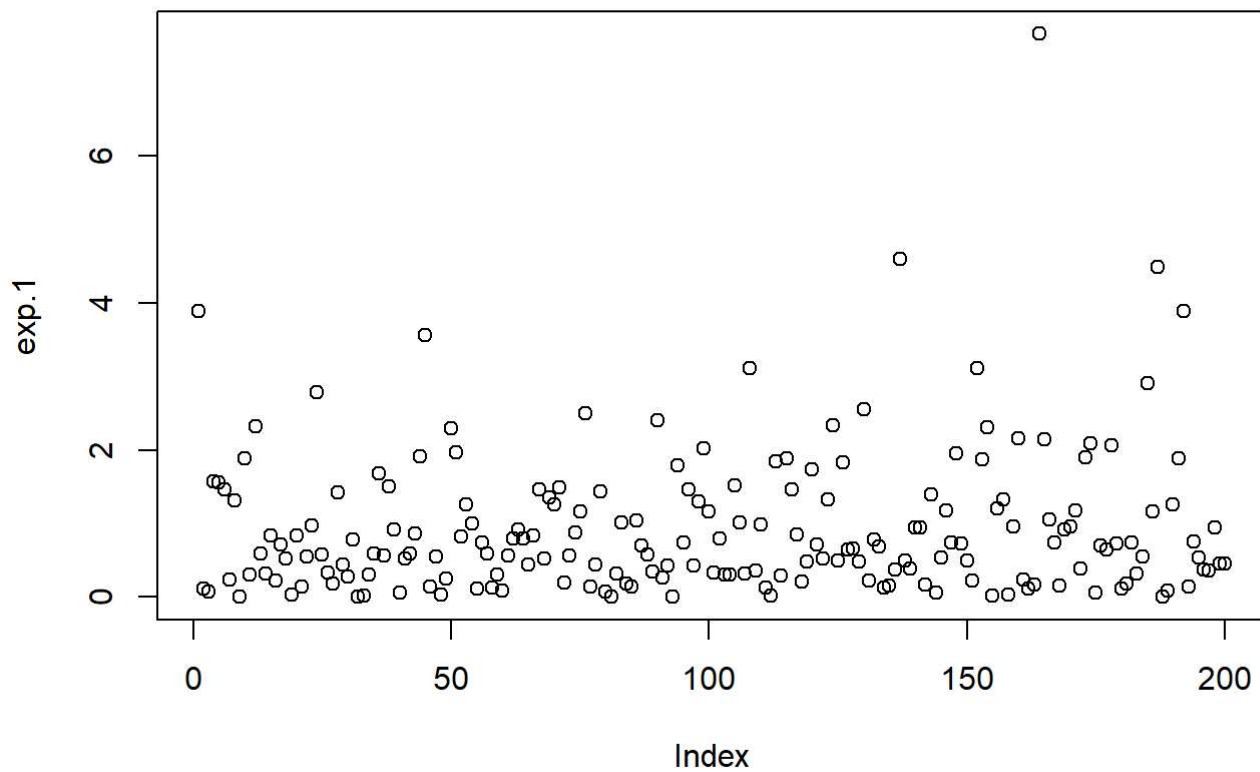
Histogram of exp.1



Получена гистограмма стандартного распределения. Лучше увеличить количество наблюдений, сейчас их слишком мало.

5. Используйте функцию `plot()` с тем же самым вектором для построения графика значений в порядке следования.

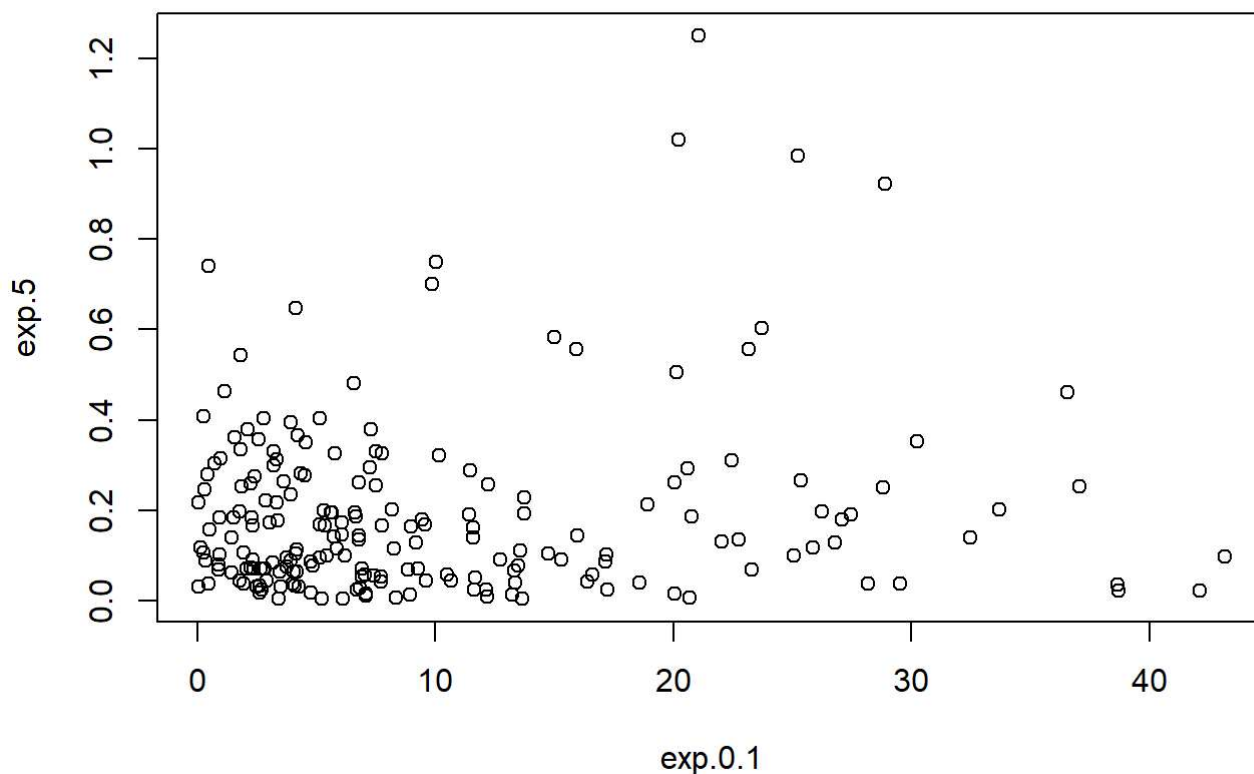
```
plot(exp.1)
```



Видно, что распределение неравномерно.

6. Используйте функцию `plot()` с любыми двумя полученными ранее векторами, чтобы получить диаграмму рассеяния (scatterplot (<https://www.mathsisfun.com/data/scatter-xy-plots.html>)) этих векторов относительно друг друга.

```
plot(exp.0.1,exp.5)
```



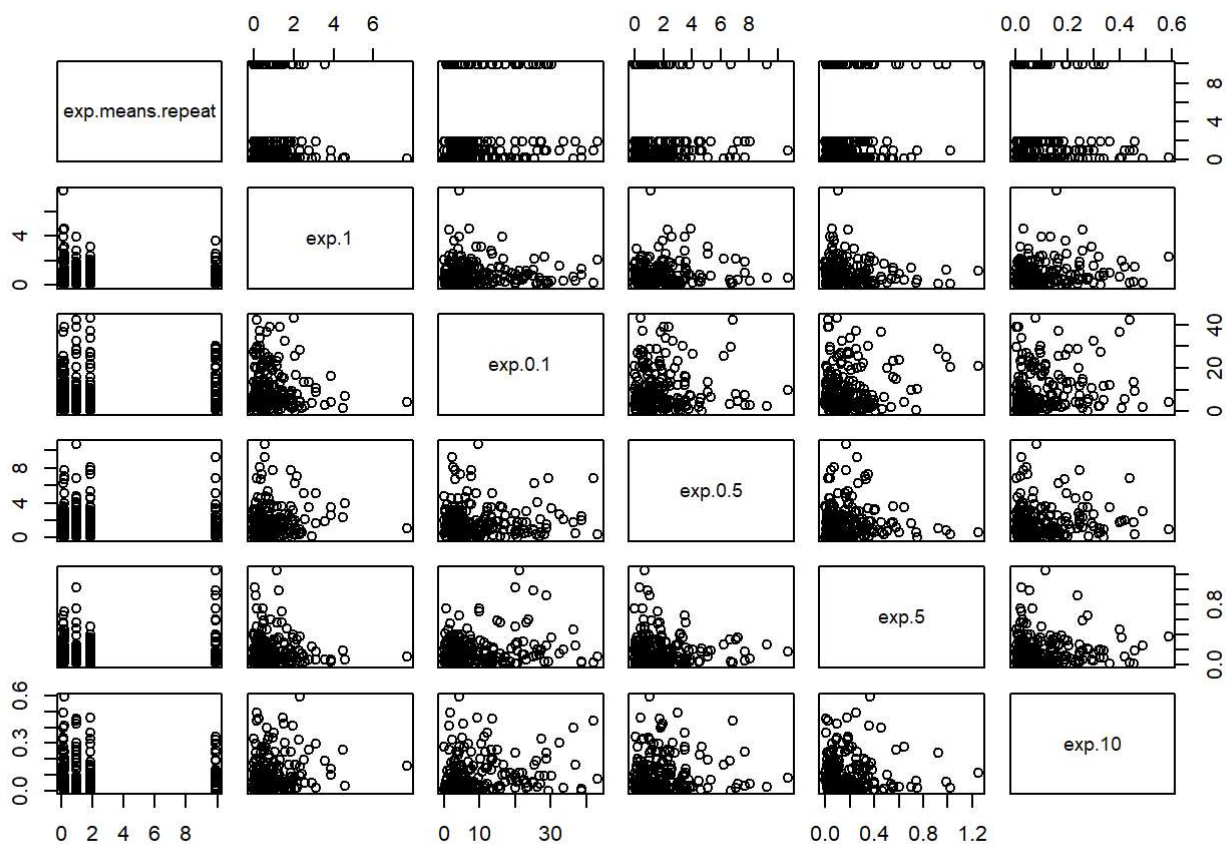
На графике нет линейной зависимости.

7. Теперь сравним свойства всех наших векторов. Для начала создадим вектор, содержащий средние значения каждого из 5 полученных ранее распределений в порядке их создания. Полученный вектор сохраним в переменную `exp.means`.

```
exp.means <- c(exp.1.mean, exp.0.1.mean, exp.0.5.mean , exp.5.mean ,exp.10.mean)
exp.sds <- c(exp.1.sd, exp.0.1.sd, exp.0.5.sd, exp.5.sd, exp.10.sd)
```

Далее необходимо создать следующие диаграммы рассеяния: 8. Вектор с пятью средними относительно каждого из пяти векторов с различными значениями λ .

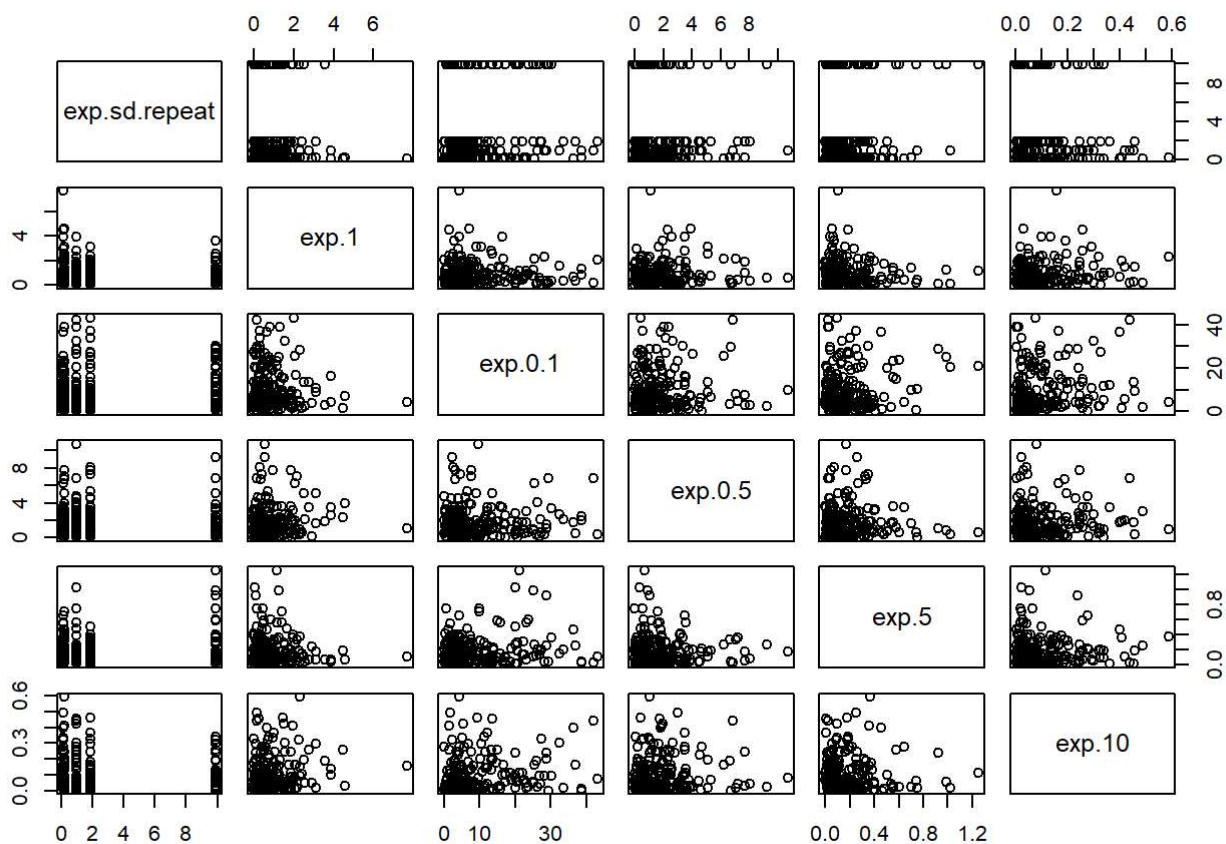
```
# ? размерности не совпадают, повторим элементы
exp.means.repeat <- rep(exp.means,each = 40);
# pairs: A matrix of scatterplots is produced.
pairs(~exp.means.repeat + exp.1 + exp.0.1 + exp.0.5 + exp.5 + exp.10) #formula
```



Не видно линейной связи

9. Вектор с пятью стандартными отклонениями относительно каждого из пяти векторов с различными значениями λ .

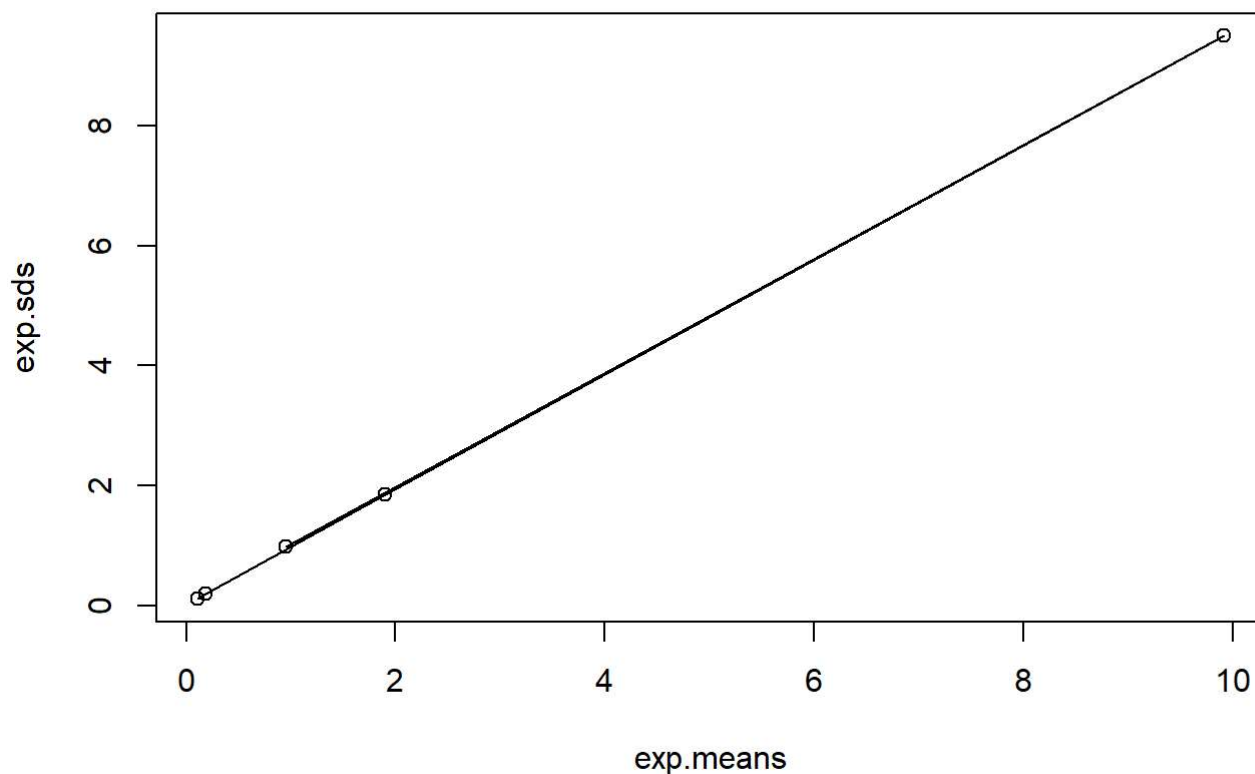
```
exp.sd.repeat <- rep(exp.means,each=40);
pairs(~exp.sd.repeat + exp.1 + exp.0.1 + exp.0.5 + exp.5 + exp.10)
```



Снова не видно линейной связи

10. Вектор с пятью средними относительно вектора с пятью стандартными отклонениями.

```
plot(exp.means,exp.sds)
lines(exp.means,exp.sds)
```



Видна линейная зависимость

Задание 2

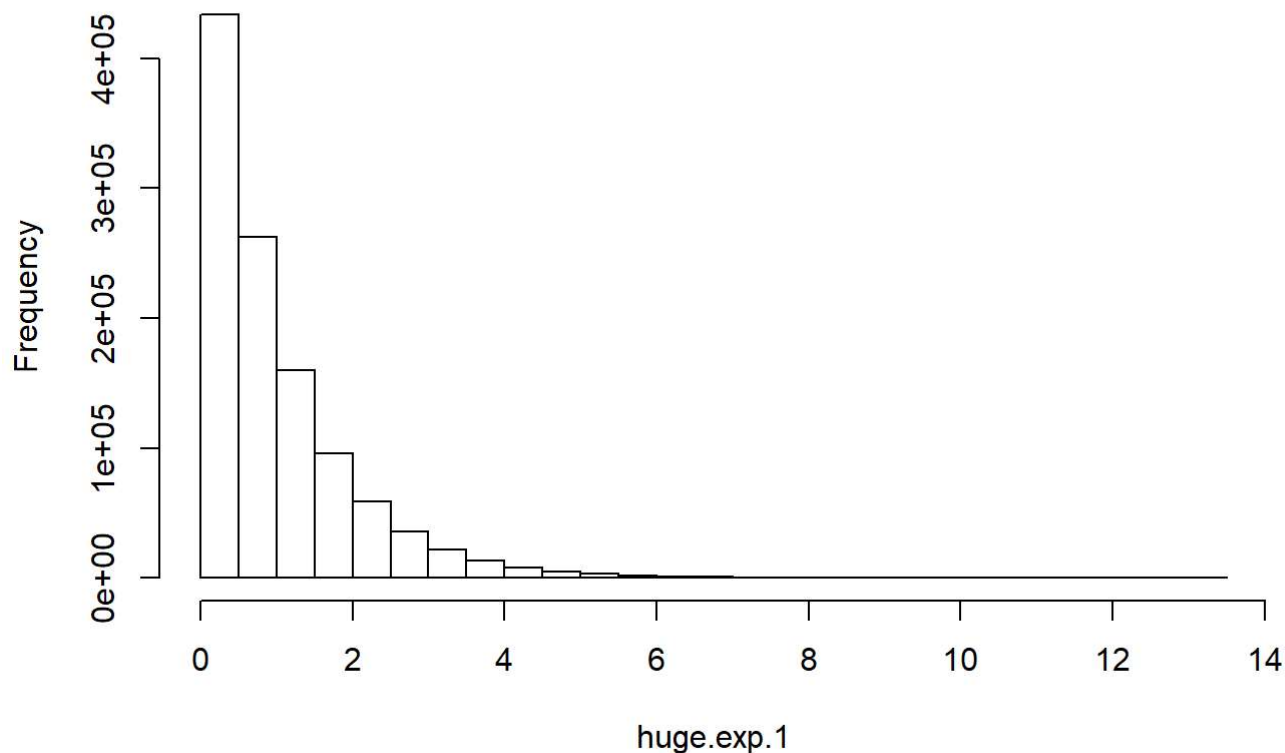
R достаточно производителен, чтобы вычислять относительно большие объёмы данных. 1. Для демонстрации сгенерируйте 1100000 (1 млн. 100 тысяч) чисел из стандартного экспоненциального распределения и сохраните их в вектор *huge.exp.1*. Получите среднее и стандартное отклонение.

```
huge.exp.1 <- rexp(1100000)
huge.exp.1.mean <- mean(huge.exp.1)
huge.exp.1.sd <- sd (huge.exp.1)
```

2. Получите гистограмму *huge.exp.1*. Она соответствует функции $1 - e^{(-x)}$? Должна ли?

```
hist(huge.exp.1)
```

Histogram of huge.exp.1



_Гистограмма не должна соответствовать функции распределения $1 - e^{-x}$. Она соответствует функции плотности распределения $x \cdot e^{-x}$ _

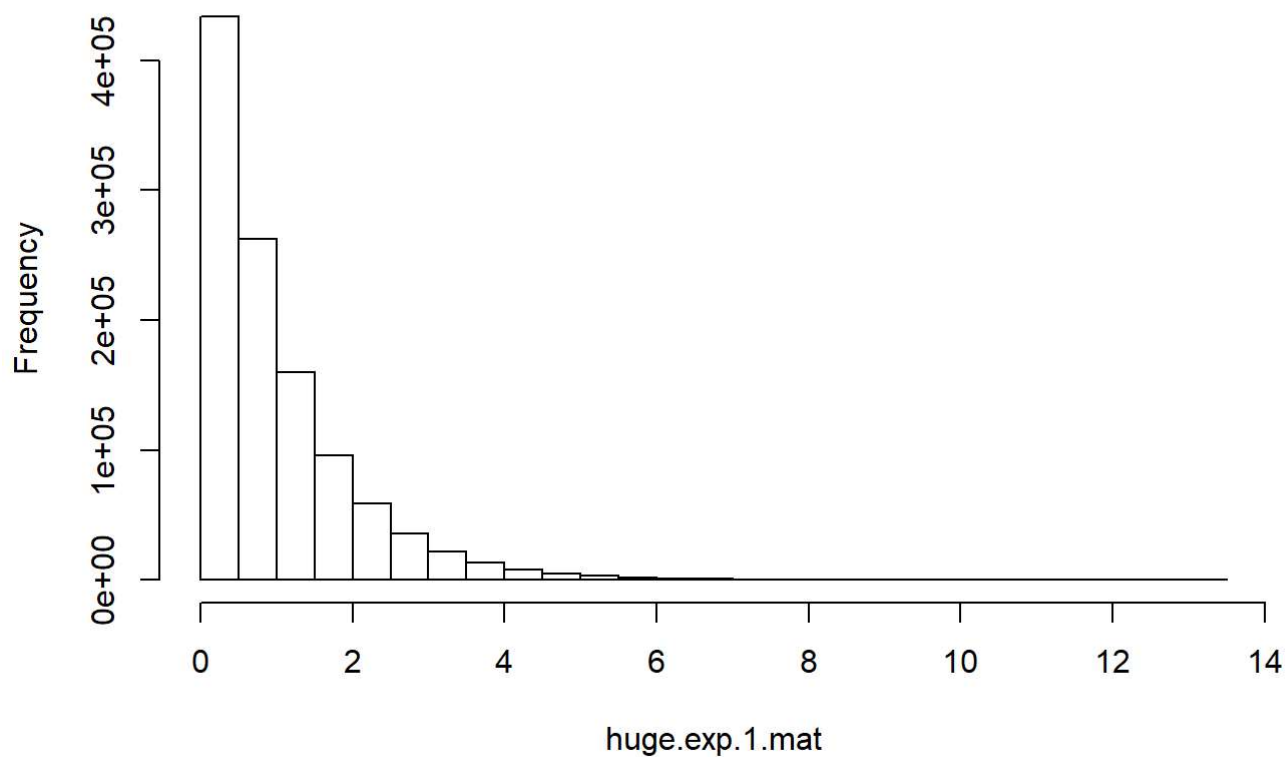
3. Получите среднее всех чисел из *huge.exp.1* строго больших 1. Для проверки выборки может потребоваться новый вектор.

```
mean.greater.1 <- mean(huge.exp.1 > 1)
```

4. Получите матрицу *huge.exp.1.mat*, содержащую 1100 строк и 1000 колонок. Постройте гистограмму, используя полученную матрицу. Что можно сказать о полученных данных?

```
huge.exp.1.mat <- matrix(huge.exp.1, nrow = 1100, ncol = 1000)
hist(huge.exp.1.mat)
```


Histogram of huge.exp.1.mat



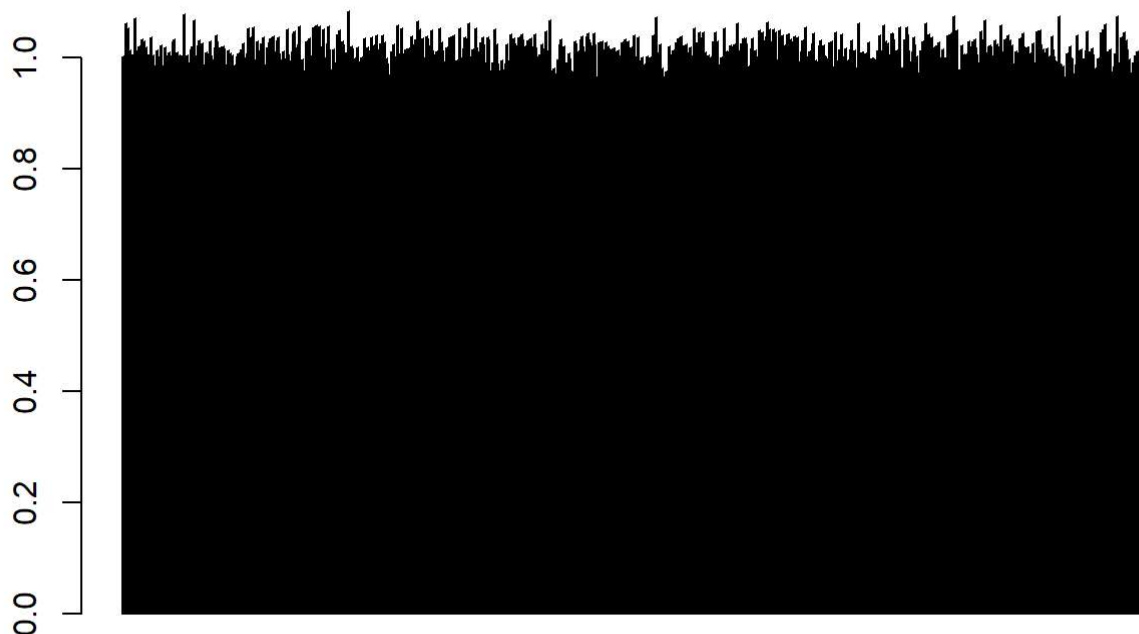
Гистограмма получилась такой же

5. Рассчитайте среднее 137-й колонки из *huge.exp.1.mat*.

```
mean.huge.mat.1.col137 <- colMeans(huge.exp.1.mat)[137]
```

6. Найдите среднее всей 1000 колонок и сохраните их в вектор. Отобразите диаграмму средних значений колонок. Объясните почему форма диаграммы отличается от виденных ранее.

```
mean.huge.mat.1.allCol <- colMeans(huge.exp.1.mat)
barplot(mean.huge.mat.1.allCol)
```



Получилась диаграмма стандартного распределения. ЦПТ: сумма большого количества независимых случайных величин имеет распределение, близкое к нормальному

7. Рассчитайте квадрат каждого значения из *huge.exp.1* и найдите среднее и стандартное отклонение полученного вектора квадратов. Объясните отличие от соответствующих показателей вектора *huge.exp.1*. Вам может помочь формула, по которой R рассчитывает стандартное отклонение:
- $$\sqrt{\text{sum}((x - \text{mean}(x))^2) / (n - 1)}.$$

```
huge.exp.1.sqrt <- sqrt(huge.exp.1)
huge.exp.sqrt.mean <- mean(huge.exp.1.sqrt)
huge.exp.sqrt.sd <- sd(huge.exp.1.sqrt)
```

Так как *x* находится в числителе, то при его увеличении увеличиваются и среднее, и стандартное отклонения