

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

- 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
```

```

## [1] 1.093043152 1.689616994 0.884805477 0.211863660 1.862613862
## [6] 0.662489289 0.236936472 1.633571703 0.843848321 1.602925293
## [11] 0.757939738 0.230458652 0.258674321 0.716097899 1.816602912
## [16] 0.588140481 0.725372315 0.972511986 0.191187848 1.084424958
## [21] 0.376419627 1.785821954 0.321796915 0.743481342 2.556402160
## [26] 0.296785574 0.950781936 2.007425642 0.787860531 1.990355564
## [31] 0.771657146 0.487835611 0.439927363 0.299166758 2.458793143
## [36] 1.539673522 0.621522342 0.036212125 0.434148562 0.670421151
## [41] 0.152893409 0.379078278 0.992559302 0.128818655 2.297600220
## [46] 0.877647520 0.470188422 2.481886336 2.353166155 0.565129335
## [51] 0.088421597 0.439825484 0.244003590 0.930868112 2.580610337
## [56] 0.206203283 0.214442268 0.802758157 1.624611219 2.335880017
## [61] 1.888650616 4.446247894 0.308115865 0.589774406 0.128413713
## [66] 0.388819943 0.811585247 0.119506660 1.060264878 0.865024545
## [71] 6.831425173 2.322568749 0.701306793 0.034324786 1.780571442
## [76] 0.500918580 0.357798706 1.385213138 0.654897563 1.084045065
## [81] 0.549553042 1.461217446 3.071670829 0.283756861 0.603784163
## [86] 1.330430033 0.568238630 0.037797794 0.141869884 0.864386200
## [91] 0.156786138 2.368422749 0.232775290 1.800389368 0.896586028
## [96] 0.908587310 0.874766177 0.010380254 1.291285040 0.311406351
## [101] 1.677617446 2.071821604 0.559862695 0.596845553 0.062790969
## [106] 0.048583701 1.080234957 0.876506377 0.527440989 0.325880849
## [111] 0.957416845 0.315770773 2.296252383 1.394490844 0.031780350
## [116] 1.994828548 0.948401427 0.899124264 0.886353067 0.058138824
## [121] 0.009571081 0.068896279 0.236147809 0.143362758 0.143015016
## [126] 2.108355677 1.340117457 0.012943329 1.393105397 0.755091581
## [131] 0.462220063 0.354451240 3.228317432 0.742595366 2.019531270
## [136] 3.021670945 1.260681701 5.590736885 0.573619704 2.262791740
## [141] 0.586204026 0.789260904 0.297496201 0.014154489 0.402157736
## [146] 0.370380979 3.436287589 0.351569741 0.637971880 1.399006083
## [151] 0.063441525 0.193306252 0.812472942 0.497899038 1.021304230
## [156] 0.567004574 0.088610904 1.266940973 0.073655058 0.740443396
## [161] 1.720560986 0.043902504 2.015357481 0.501343219 0.482080551
## [166] 0.258970901 0.150115927 0.260295642 0.540910368 0.460193092
## [171] 0.444690448 1.922824263 1.880649391 0.433575676 0.558304015
## [176] 1.211635183 3.509490738 2.056768831 2.993090018 0.498825583
## [181] 0.347877809 0.321781320 0.721447906 0.284073626 1.771541017
## [186] 0.965197986 0.135151958 1.466056787 1.959568536 0.833890719
## [191] 0.251988066 2.133697692 0.042975280 0.185793609 0.136043672
## [196] 0.525694789 2.983045039 0.292510214 0.415056485 0.765859858

```

```

exp.1.mean <- mean(exp.1)
exp.1.mean

```

```

## [1] 0.9862977

```

```

exp.1.sd <- sd(exp.1)
exp.1.sd

```

```

## [1] 0.9902615

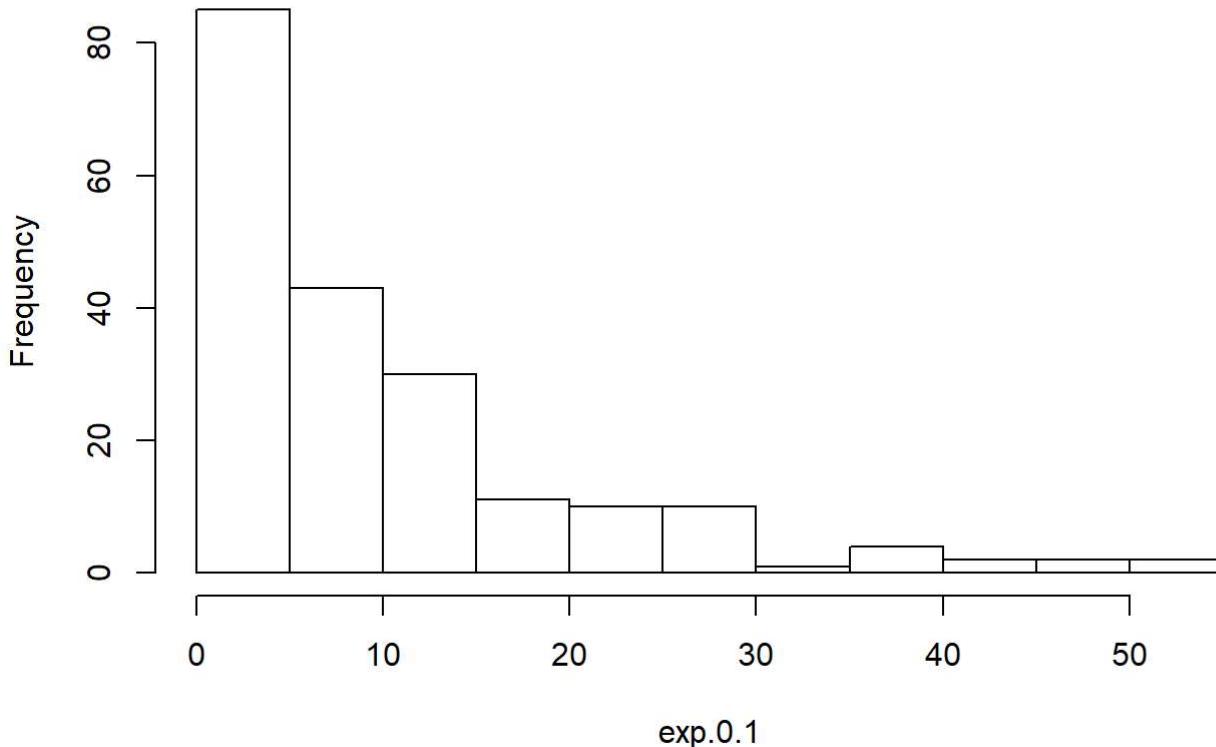
```

2. Повторите, использовав 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)

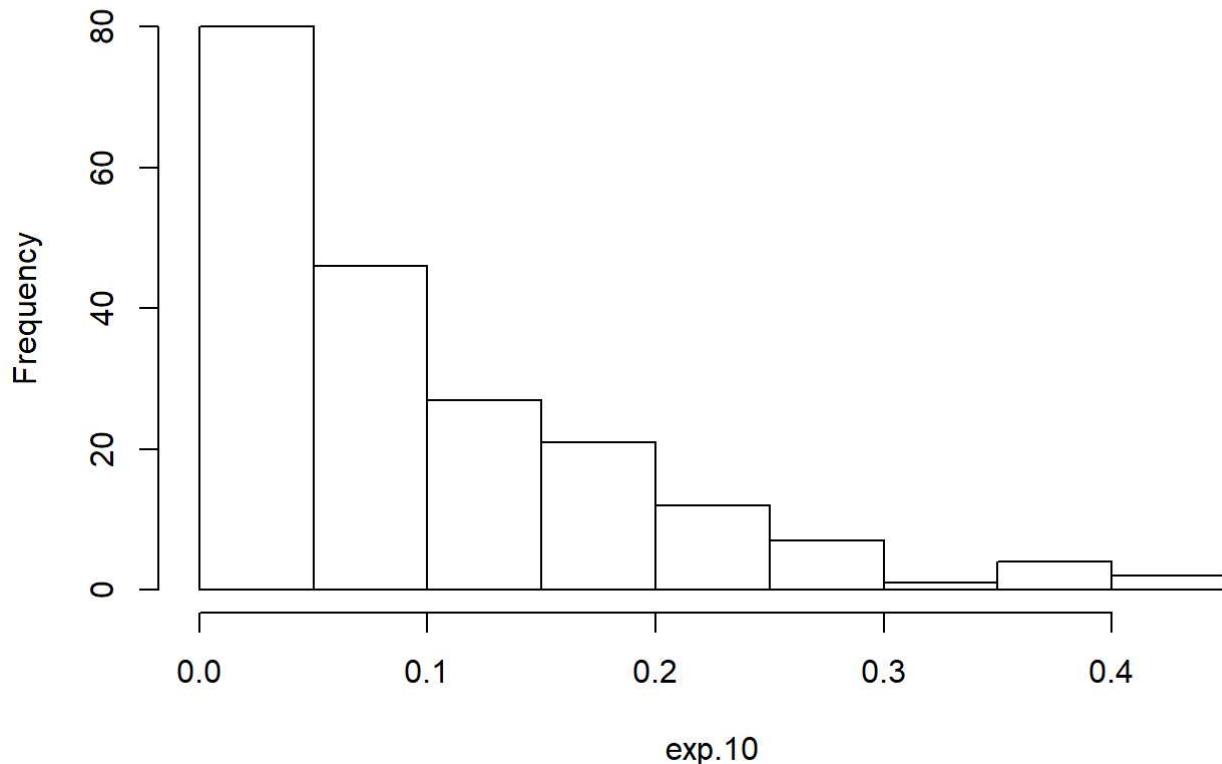
hist(exp.0.1)
```

Histogram of exp.0.1



```
hist(exp.10)
```

Histogram of exp.10



mean

```
exp.0.1.mean <- mean(exp.0.1)
exp.0.1.mean
```

```
## [1] 10.18243
```

```
exp.0.5.mean <- mean(exp.0.5)
exp.0.5.mean
```

```
## [1] 1.867569
```

```
exp.5.mean <- mean(exp.5)
exp.5.mean
```

```
## [1] 0.2042966
```

```
exp.10.mean <- mean(exp.10)
exp.10.mean
```

```
## [1] 0.0960838
```

sd

```
exp.0.1.sd <- sd(exp.0.1)  
exp.0.1.sd
```

```
## [1] 10.87487
```

```
exp.0.5.sd <- sd(exp.0.5)  
exp.0.5.sd
```

```
## [1] 1.961405
```

```
exp.5.sd <- sd(exp.5)  
exp.5.sd
```

```
## [1] 0.1949645
```

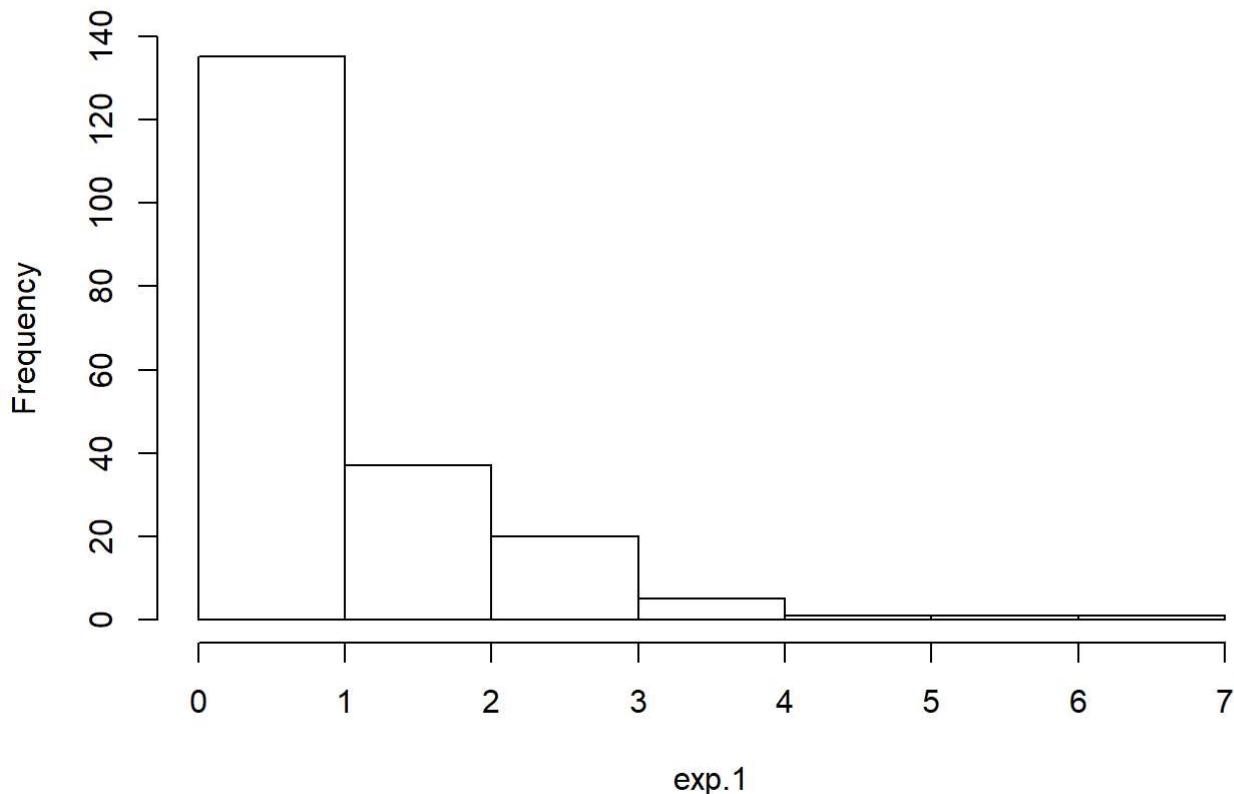
```
exp.10.sd <- sd(exp.10)  
exp.10.sd
```

```
## [1] 0.09098541
```

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)
```

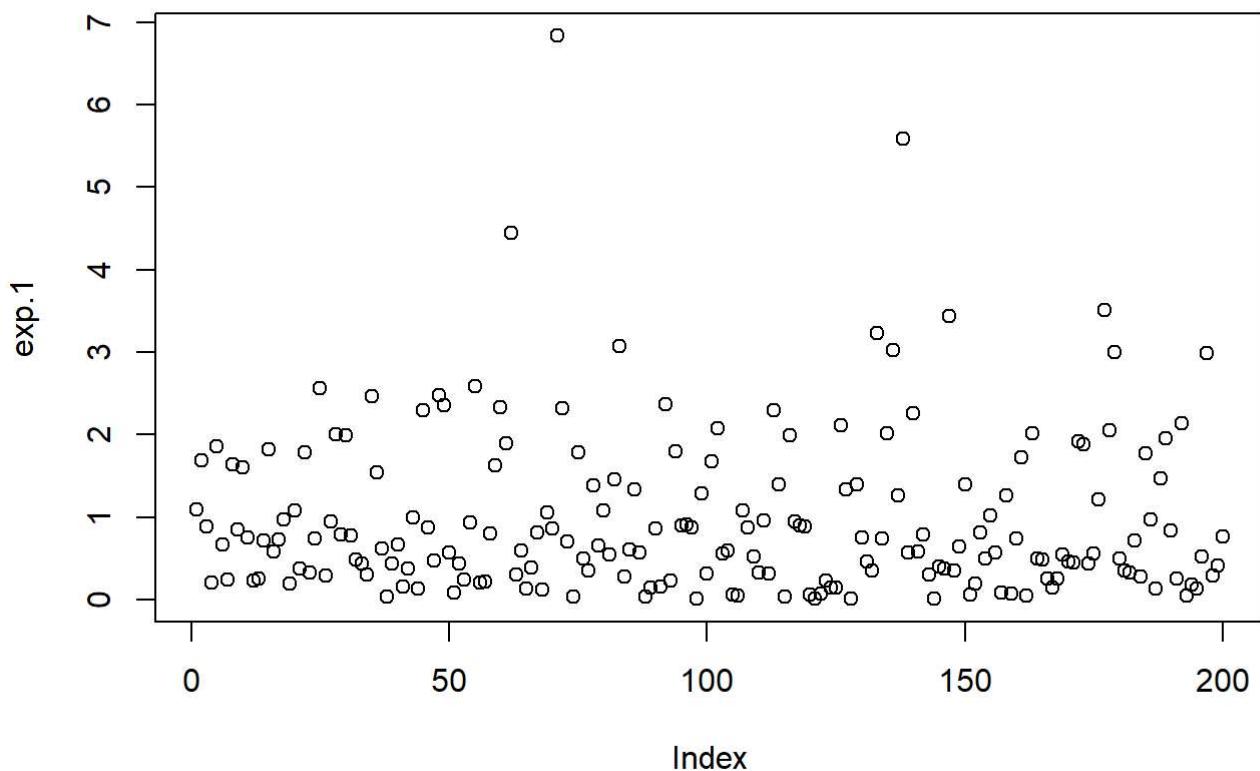
Histogram of exp.1



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.

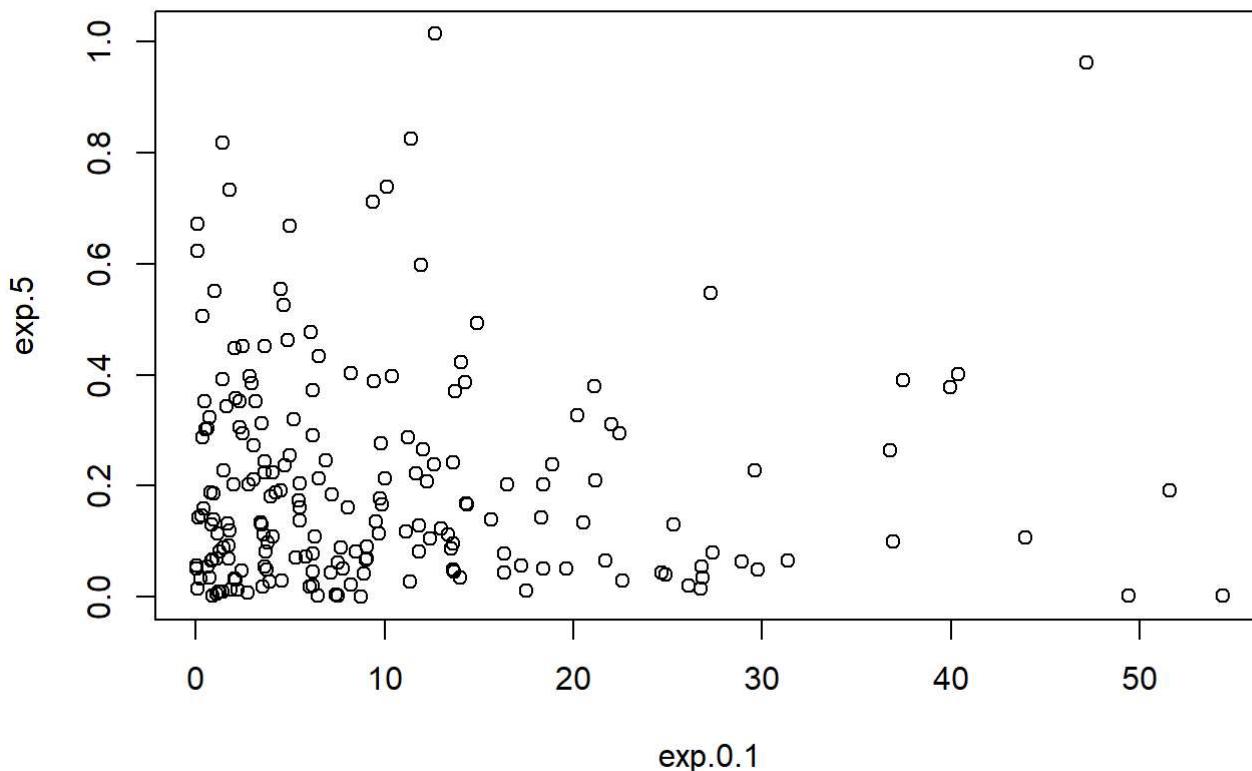
```
plot(exp.1)
```



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.means
```

```
## [1] 0.9862977 10.1824252 1.8675688 0.2042966 0.0960838
```

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

```
## [1] 0.99026154 10.87486525 1.96140459 0.19496452 0.09098541
```

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)
exp.means.repeat
```

```

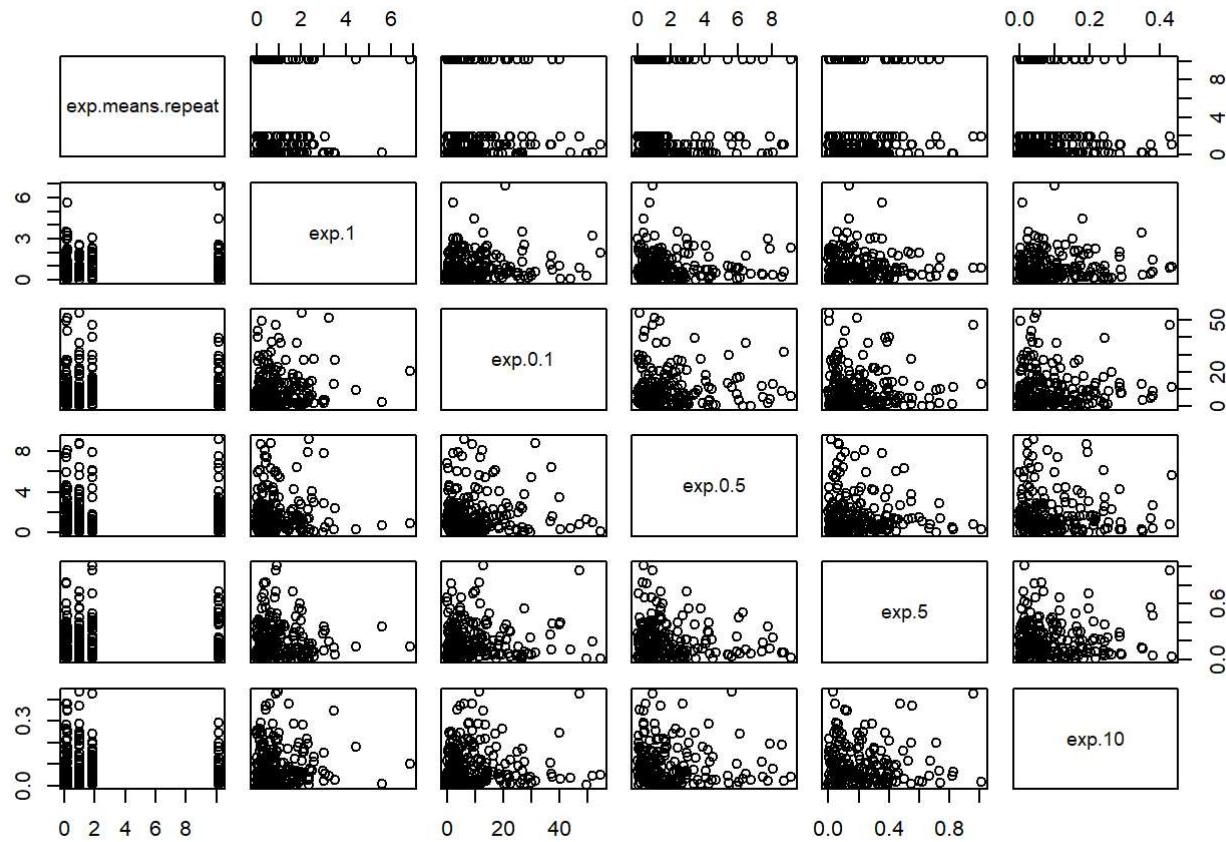
## [1] 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977
## [7] 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977
## [13] 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977
## [19] 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977
## [25] 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977
## [31] 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977
## [37] 0.9862977 0.9862977 0.9862977 0.9862977 0.9862977 10.1824252 10.1824252
## [43] 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252
## [49] 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252
## [55] 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252
## [61] 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252
## [67] 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252
## [73] 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252 10.1824252
## [79] 10.1824252 10.1824252 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688
## [85] 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688
## [91] 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688
## [97] 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688
## [103] 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688
## [109] 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688
## [115] 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688 1.8675688
## [121] 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966
## [127] 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966
## [133] 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966
## [139] 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966
## [145] 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966
## [151] 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966 0.2042966
## [157] 0.2042966 0.2042966 0.2042966 0.2042966 0.0960838 0.0960838
## [163] 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838
## [169] 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838
## [175] 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838
## [181] 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838
## [187] 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838
## [193] 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838 0.0960838
## [199] 0.0960838 0.0960838

```

```

# pairs: A matrix of scatterplots is produced.
pairs(~exp.means.repeat + exp.1 + exp.0.1 + exp.0.5 + exp.5 + exp.10) #formula

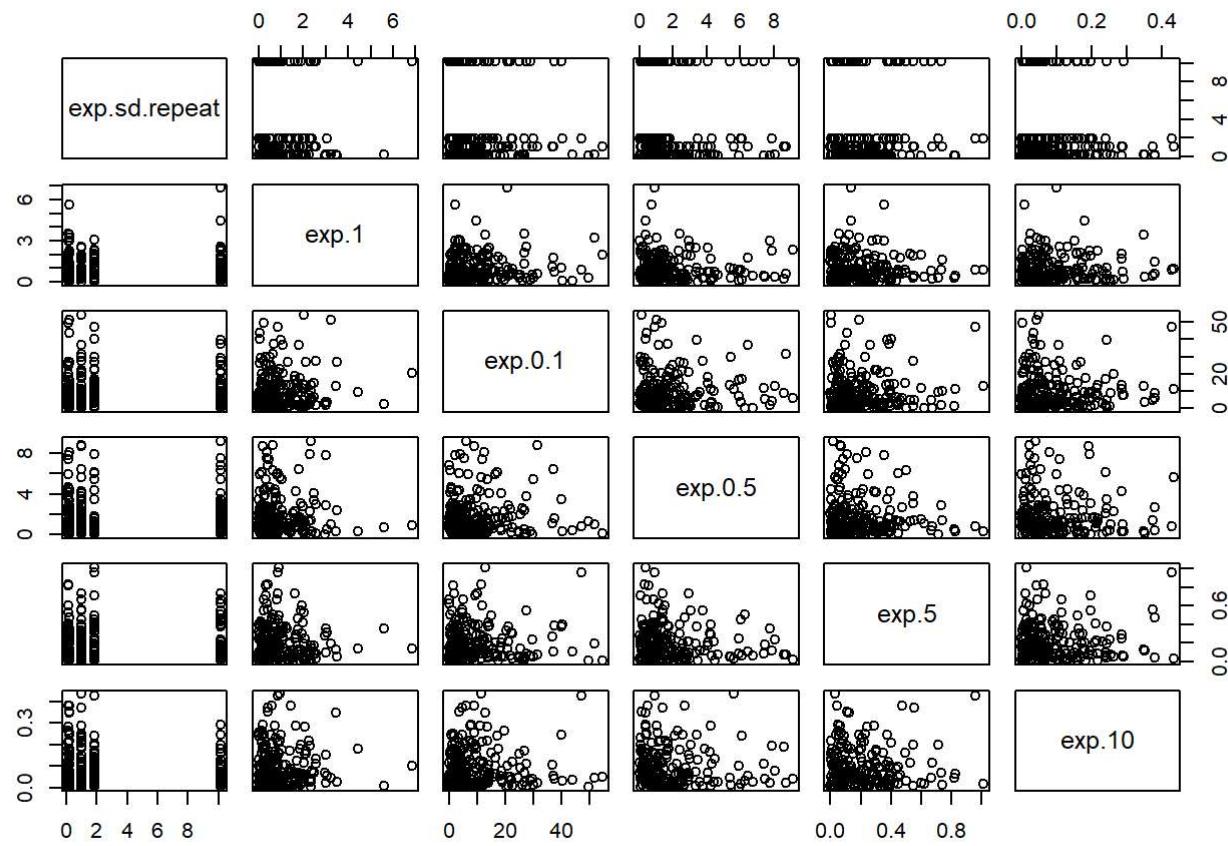
```



Не видно линейной связи

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

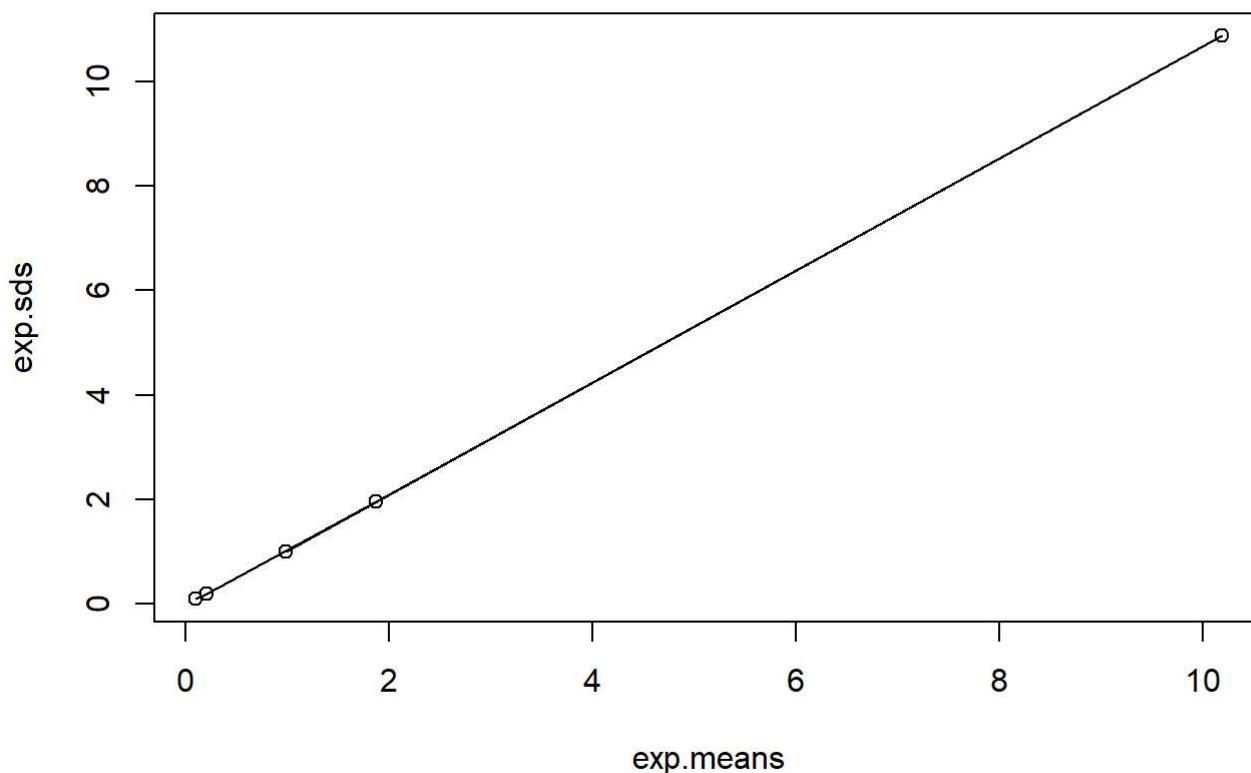
```
exp.sd.repeat <- rep(exp.means,each=40);
pairs(~exp.sd.repeat + exp.1 + exp.0.1 + exp.0.5 + exp.5 + exp.10)
```



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.mean
```

```
## [1] 0.9992001
```

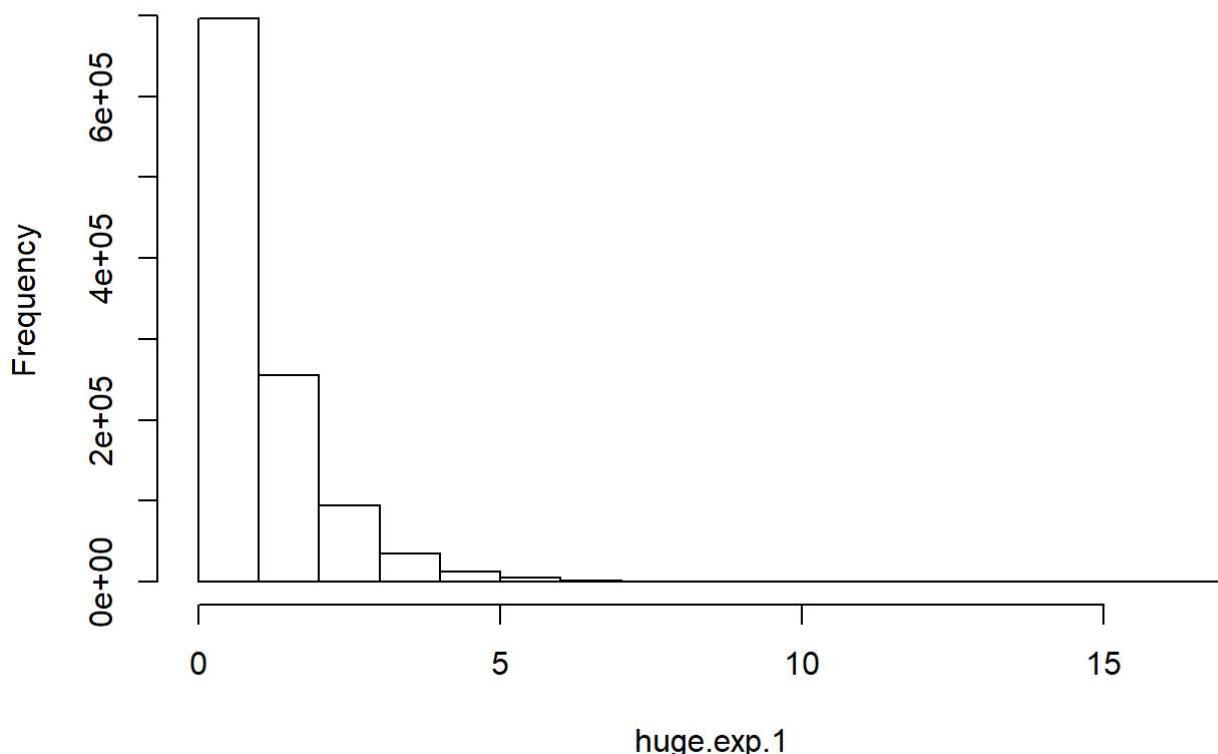
```
huge.exp.1.sd <- sd (huge.exp.1)
huge.exp.1.sd
```

```
## [1] 0.9994721
```

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^*e^{-x} __

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

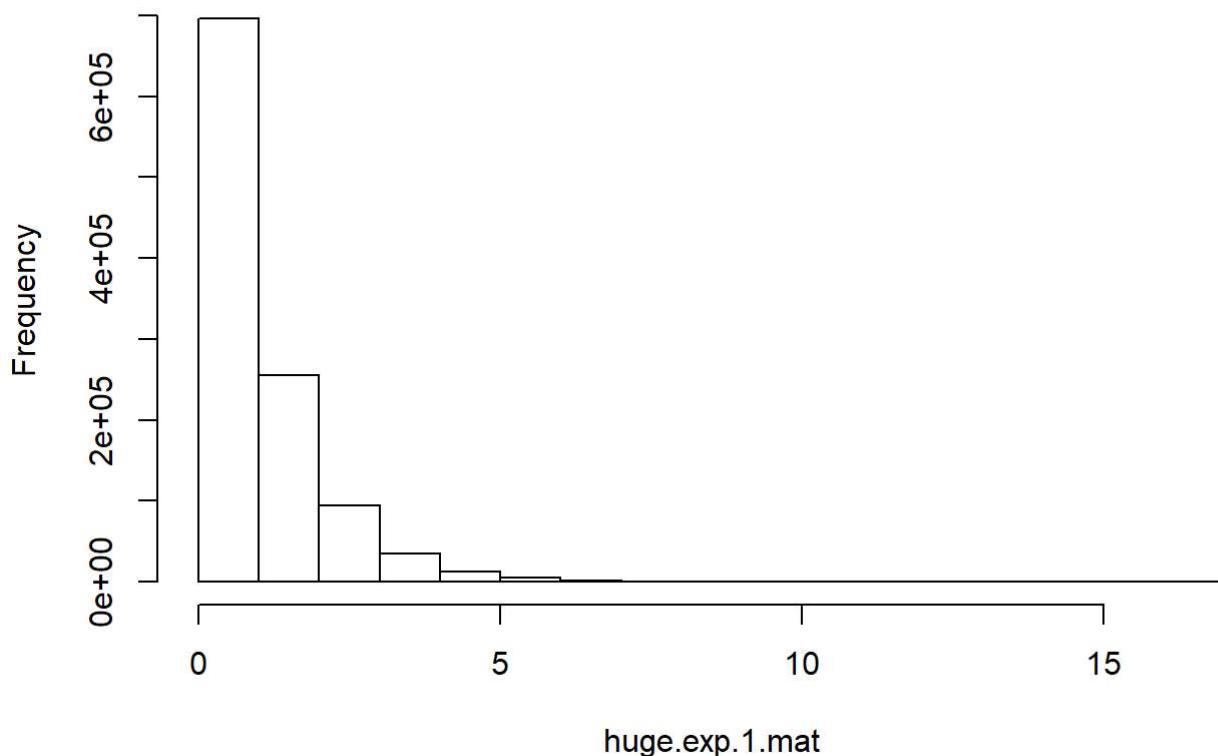
```
mean.greater.1 <- mean(huge.exp.1 > 1)  
mean.greater.1
```

```
## [1] 0.3677191
```

4. Получите матрицу *huge.exp.1.mat*, содержащую 1100 строк и 1000 колонок. Постройте гистограмму, использовав 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



```
## [1] 1.0506314 0.9662972 1.0142187 1.0570840 0.9828459 0.9935274
## [7] 0.9991601 0.9544631 1.0262087 0.9930322 0.9557260 1.0039254
## [13] 1.0092983 0.9809341 0.9994599 0.9309575 1.0356663 1.0160548
## [19] 0.9588072 0.9496915 1.0306684 0.9784776 0.9666463 1.0061170
## [25] 0.9718319 1.0373077 1.0118667 0.9988393 0.9631973 0.9584809
## [31] 0.9986189 1.0285850 1.0183153 0.9855397 0.9537568 0.9747608
## [37] 0.9947267 0.9991767 0.9979233 1.0055575 0.9845538 0.9857181
## [43] 0.9850340 0.9880246 1.0165738 0.9494333 0.9991090 0.9870504
## [49] 0.9610340 0.9947416 1.0387725 1.0123180 1.0142086 1.0163039
## [55] 0.9907253 0.9680168 1.0188181 0.9964620 0.9882581 1.0262207
## [61] 1.0121795 1.0084386 0.9580506 1.0598077 1.0014997 0.9982683
## [67] 1.0393343 0.9408167 0.9764138 0.9855956 1.0099214 1.0248983
## [73] 1.0182327 1.0067127 1.0360761 0.9885501 1.0045862 1.0089876
## [79] 0.9600563 0.9831776 1.0282301 1.0634422 1.0272239 0.9997818
## [85] 1.0085183 0.9832290 1.0470333 0.9993925 0.9949837 1.0031925
## [91] 0.9916218 0.9559129 0.9922458 1.0212100 1.0116160 1.0100067
## [97] 0.9966828 1.0286213 1.0580193 0.9677739 1.0483956 0.9408349
## [103] 0.9906795 1.0133105 1.0168817 0.9389116 1.0040682 1.0229644
## [109] 0.9850126 1.0578246 1.0317994 1.0561883 0.9743628 1.0039254
## [115] 0.9732598 0.9706537 1.0004479 1.0155318 0.9685370 0.9441544
## [121] 0.9986808 1.0078013 0.9928058 1.0045318 1.0470721 0.9578176
## [127] 0.9521941 0.9927038 0.9815628 0.9977937 0.9658591 0.9815654
## [133] 0.9556800 1.0344856 0.9859491 1.0199992 0.9800154 1.0016382
## [139] 1.0501942 0.9581802 0.9357954 1.0071312 1.0263382 0.9806092
## [145] 0.9948474 1.0053472 1.0166893 0.9943179 0.9470656 1.0260742
## [151] 0.9531698 1.0398133 0.9915990 0.9198481 0.9567542 1.0090161
## [157] 0.9953803 0.9891775 1.0165716 1.0265984 0.9590519 1.0016511
## [163] 1.0072268 1.0017531 0.9962401 1.0247860 0.9783925 1.0036222
## [169] 0.9785615 1.0862179 0.9676080 1.0014487 0.9811023 1.0225756
## [175] 1.0028200 1.0358317 1.0770947 1.0210263 0.9929671 1.0275455
## [181] 0.9958155 0.9515565 1.0309364 1.0021170 1.0259236 0.9750657
## [187] 1.0019737 1.0399601 1.0024798 0.9985185 0.9647751 0.9907804
## [193] 0.9867220 0.9871099 0.9903360 1.0536775 0.9975847 1.0455261
## [199] 1.0019117 0.9786260 1.0050977 0.9941254 0.9977892 0.9833576
## [205] 1.0077009 0.9771626 1.0062408 0.9170381 1.0042880 0.9666175
## [211] 1.0000639 0.9818435 1.0653140 0.9476022 1.0003014 0.9949671
## [217] 0.9666373 1.0159619 1.0375965 1.0759762 1.0326323 1.0220229
## [223] 0.9920903 0.9933555 0.9988206 0.9995733 0.9704998 0.9753255
## [229] 1.0022860 1.0097608 0.9181683 0.9679794 0.9687292 0.9565533
## [235] 0.9127689 1.0082389 1.0142208 0.9652050 1.0098295 1.0040247
## [241] 1.0381244 1.0273117 1.0064998 0.9976464 0.9662847 0.9873231
## [247] 0.9955212 1.0020838 0.9874770 0.9419300 1.0453125 1.0175566
## [253] 0.9849952 0.9997817 1.0045389 1.0316427 0.9693426 1.0428200
## [259] 1.0733887 0.9670629 1.0375908 1.0163006 0.9804975 0.9690288
## [265] 0.9966144 1.0265484 0.9299106 0.9864257 0.9851570 1.0276318
## [271] 1.0054785 1.0070680 1.0020491 1.0345305 0.9354896 1.0187179
## [277] 0.9850310 1.0460563 0.9876483 1.0308424 1.0144550 1.0454434
## [283] 1.0327721 1.0116524 1.0058220 0.9468925 1.0132010 0.9937876
## [289] 1.0494591 0.9735317 0.9566493 1.0251763 1.0325160 0.9741668
## [295] 1.0223099 0.9689542 0.9615376 1.0261095 1.0095856 0.9745659
## [301] 0.9909989 1.0217484 0.9686108 1.0208172 0.9488266 1.0155634
## [307] 0.9912336 0.9993415 0.9350551 0.9873356 0.9769581 1.0231860
## [313] 0.9734253 0.9960771 0.9710358 1.0691481 1.0097965 0.9883333
```

```

## [319] 0.9513080 1.0027666 1.0296641 0.9920280 1.0329409 1.0178756
## [325] 1.0132712 1.0003510 0.9643366 1.0442119 0.9937128 0.9869879
## [331] 0.9628639 0.9850259 0.9854061 1.0254432 1.0370007 0.9679014
## [337] 1.0462261 0.9969167 0.9856443 0.9920950 0.9881950 0.9749132
## [343] 0.9873303 1.0710240 1.0244954 0.9546684 1.0211630 0.9679041
## [349] 1.0010955 1.0432386 0.9998313 0.9252293 0.9957561 0.9912794
## [355] 0.9445845 0.9771059 0.9924471 1.0226345 0.9708363 1.0092218
## [361] 1.0265259 0.9608394 0.9793467 1.0032741 0.9414492 0.9854803
## [367] 1.0112688 0.9919539 0.9664131 0.9864752 0.9678761 0.9632544
## [373] 0.9631684 0.9837417 0.9509366 1.0208973 0.9990904 1.0115894
## [379] 0.9425078 1.0117222 1.0126579 0.9877869 1.0056696 0.9645355
## [385] 0.9962752 1.0517710 1.0481864 1.0214873 0.9853157 1.0218417
## [391] 0.9761578 1.0113294 1.0067746 1.0095886 1.0067174 1.0438826
## [397] 1.0178455 0.9995226 0.9617397 0.9685547 0.9372013 1.0164779
## [403] 1.0265691 0.9853165 0.9991189 0.9602149 0.9882779 0.9866362
## [409] 0.9974098 1.0052826 0.9889171 0.9724184 0.9701126 1.0001699
## [415] 1.0300099 0.9884979 1.0375716 0.9484939 1.0172708 1.0534123
## [421] 0.9698902 1.0175587 0.9681708 1.0506048 0.9698804 0.9951288
## [427] 0.9972451 0.9650827 0.9922649 1.0681591 1.0094147 0.9687082
## [433] 0.9928910 1.0597019 1.0089246 0.9743479 0.9887028 0.9994625
## [439] 1.0301163 1.0852563 1.0178274 0.9897649 1.0109944 0.9971498
## [445] 0.9694692 1.0158802 1.0199727 0.9610234 0.9557164 1.0168164
## [451] 1.0155143 1.0401855 1.0092559 1.0049623 1.0229748 0.9811259
## [457] 0.9877676 0.9907344 0.9471413 0.9935374 0.9624049 1.0269114
## [463] 0.9694206 0.9803900 1.0322498 0.9479307 1.0529164 0.9952591
## [469] 0.9819752 0.9826669 1.0022782 0.9868975 1.0149519 1.0478131
## [475] 1.0015879 1.0352207 1.0197337 0.9732999 1.0725358 1.0363861
## [481] 0.9963464 1.0233744 1.0421651 0.9875666 1.0387557 0.9822628
## [487] 0.9908481 1.0229735 1.0789594 0.9484949 1.0290336 1.0440197
## [493] 1.0019967 1.0063640 1.0163710 0.9906985 0.9861235 0.9663574
## [499] 0.9948672 1.0193397 0.9945878 1.0181749 0.9980156 1.0261613
## [505] 1.0157957 0.9789042 0.9878564 1.0119395 0.9801472 0.9733863
## [511] 1.0029249 0.9997397 0.9595655 1.0071025 0.9859893 1.0322525
## [517] 0.9813181 0.9995289 1.0293707 1.0170716 1.0222061 1.0253443
## [523] 0.9693184 1.0092850 0.9746863 0.9993244 0.9821867 1.0349458
## [529] 1.0034602 0.9636814 0.9784519 1.0126298 1.0167831 1.0186538
## [535] 1.0681837 1.0159689 1.0172222 0.9839991 1.0029975 0.9309896
## [541] 1.0161552 1.0557513 1.0137898 0.9985275 1.0162437 1.0366441
## [547] 1.0038919 0.9568821 1.0341000 1.0517495 0.9858852 1.0200167
## [553] 0.9445912 0.9551518 1.0082638 1.0072542 0.9093179 0.9580325
## [559] 0.9783392 1.0440029 0.9825660 1.0152082 0.9844922 1.0115476
## [565] 0.9955489 1.0101864 1.0291475 1.0145356 1.0087295 1.0344697
## [571] 1.0274757 1.0073042 0.9950870 1.0777561 0.9726245 1.0024927
## [577] 0.9710285 0.9878388 1.0071625 0.9767643 0.9995631 1.0147192
## [583] 0.9784584 1.0716644 0.9701059 1.0293763 0.9824197 0.9740320
## [589] 0.9947230 1.0635815 0.9916170 0.9763234 1.0275279 1.0107502
## [595] 0.9917838 1.0303055 0.9678218 0.9695252 0.9728650 1.0015921
## [601] 1.0272895 1.0441341 1.0175074 0.9891320 1.0414183 1.0406385
## [607] 1.0143229 1.0400699 0.9566977 0.9506884 1.0088237 1.0366054
## [613] 0.9991555 1.0502690 1.0156543 1.0029483 0.9776725 1.0033105
## [619] 0.9815436 0.9495833 1.0118661 1.0446167 1.0291288 1.0214410
## [625] 0.9967236 1.0173414 0.9977552 1.0105722 1.0231555 0.9658282
## [631] 1.0135185 1.0241013 1.0235355 1.0309269 1.0044427 1.0344433
## [637] 0.9982270 0.9435681 1.0142922 0.9548656 1.0080807 1.0198659

```

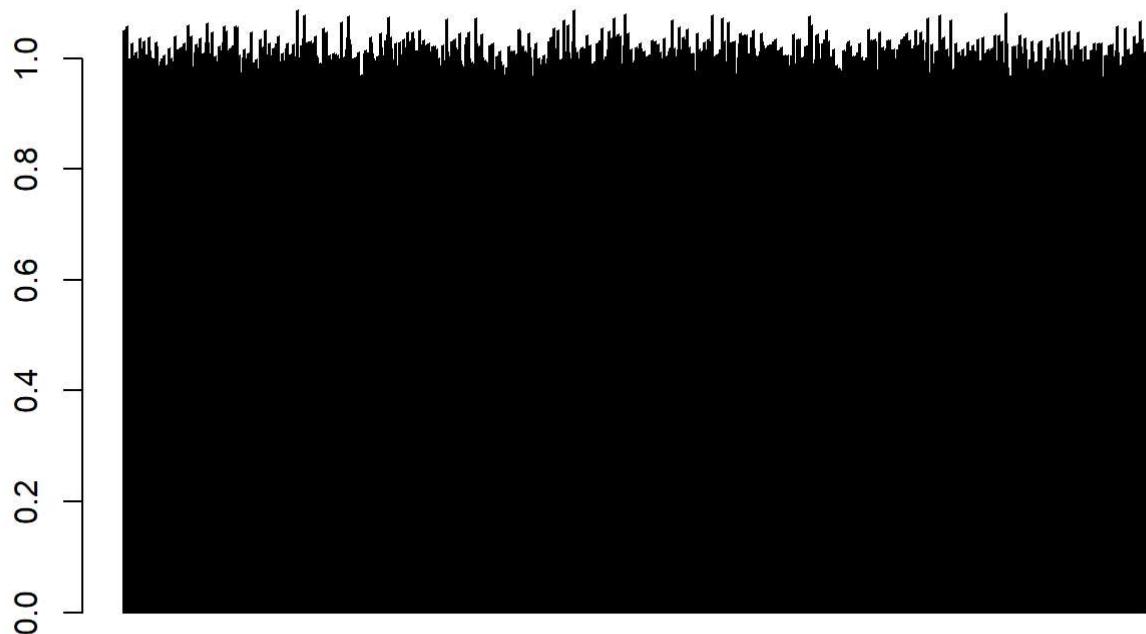
```

## [643] 0.9969183 1.0020836 1.0052931 0.9858040 0.9524703 1.0046228
## [649] 0.9848683 0.9860966 0.9257599 1.0050157 1.0419868 0.9452239
## [655] 0.9307112 0.9898893 1.0345430 0.9669899 1.0336833 0.9661189
## [661] 0.9970030 1.0010692 0.9732542 0.9869919 1.0211160 0.9974221
## [667] 0.9988784 1.0233933 1.0748908 1.0013417 1.0587028 0.9308858
## [673] 0.9884331 0.9909622 0.9979839 1.0409444 1.0433886 1.0239223
## [679] 0.9970368 0.9922318 0.9779434 1.0268717 0.9965136 1.0333230
## [685] 1.0089328 1.0501174 0.9999724 1.0295904 0.9920723 0.9897572
## [691] 1.0030149 1.0027745 1.0161106 0.9850407 0.9872300 0.9708819
## [697] 0.9804582 0.9797581 0.9529138 0.9433265 0.9767521 1.0141198
## [703] 1.0102549 0.9636690 0.9912480 1.0268887 1.0305881 0.9806937
## [709] 1.0205907 0.9668026 1.0028571 0.9683141 1.0023163 0.9923432
## [715] 0.9902999 1.0100855 0.9900474 1.0264910 0.9710537 1.0007264
## [721] 0.9947807 0.9941267 0.9763632 0.9965920 1.0016574 0.9689033
## [727] 1.0521562 0.9941969 1.0310866 1.0111015 1.0316321 1.0168085
## [733] 1.0335133 1.0287317 0.9780899 0.9637451 1.0465626 1.0070856
## [739] 0.9938319 1.0200044 0.9368398 1.0144662 1.0035068 1.0177635
## [745] 1.0323483 0.9980749 1.0312709 1.0130581 0.9885394 0.9658048
## [751] 1.0157921 0.9629270 0.9984848 0.9691841 1.0151093 0.9719195
## [757] 1.0241003 0.9845158 1.0334326 0.9811416 1.0353217 1.0390361
## [763] 1.0340260 1.0443344 1.0252696 0.9293126 1.0315560 1.0149624
## [769] 1.0157689 1.0150457 0.9637734 1.0234069 1.0497000 1.0163254
## [775] 1.0232797 0.9777016 1.0468966 0.9966142 1.0323794 0.9510969
## [781] 0.9523324 0.9963665 1.0283300 1.0720085 0.9594992 0.9629144
## [787] 0.9744575 1.0245197 0.9360677 0.9893990 0.9694795 1.0121852
## [793] 0.9731516 0.9721195 1.0145608 1.0763348 0.9712233 0.9501592
## [799] 1.0334177 1.0375240 0.9609583 0.9988750 1.0153448 0.9784746
## [805] 0.9915836 1.0035558 1.0676422 0.9772660 0.9736807 0.9596808
## [811] 0.9812665 1.0262343 1.0012685 0.9957240 1.0110477 1.0035141
## [817] 1.0036395 1.0160648 0.9893551 1.0036865 0.9984471 0.9901574
## [823] 1.0279538 0.9762549 0.9952311 1.0135895 0.9869396 1.0230001
## [829] 1.0112409 1.0106253 1.0110993 0.9707756 1.0341110 0.9842613
## [835] 0.9816722 0.9950763 0.9715033 1.0374662 0.9899138 1.0088100
## [841] 0.9978791 0.9746934 1.0162203 1.0114973 0.9287477 0.9846791
## [847] 1.0168204 0.9540906 0.9818100 1.0413704 0.9919423 0.9955429
## [853] 0.9476907 1.0415998 1.0096913 0.9992462 1.0292145 0.9900808
## [859] 0.9764826 1.0805677 1.0254777 0.9803782 0.9295202 0.9493932
## [865] 0.9685049 0.9374200 1.0207967 0.9759690 1.0207972 0.9776331
## [871] 0.9991230 0.9773739 1.0247397 1.0410959 1.0195395 0.9646586
## [877] 0.9996324 1.0090235 1.0350689 1.0144743 0.9724577 0.9814652
## [883] 0.9695987 0.9977479 0.9607728 1.0294185 1.0097101 0.9818524
## [889] 0.9802776 0.9914727 0.9940064 0.9507049 1.0268539 1.0292219
## [895] 1.0308754 0.9211786 0.9782479 0.9749211 0.9552556 0.9997926
## [901] 1.0194669 0.9759328 1.0132244 0.9728341 1.0347318 0.9897360
## [907] 0.9680074 0.9902119 0.9991845 1.0430436 1.0290751 1.0115369
## [913] 0.9963036 0.9485208 0.9758848 1.0456274 0.9960899 0.9863327
## [919] 0.9849098 0.9651557 0.9519502 1.0487070 1.0084392 1.0161947
## [925] 0.9890939 0.9957924 0.9973677 1.0225869 0.9526363 0.9881575
## [931] 1.0467915 0.9922572 0.9834838 1.0154254 0.9747307 0.9679309
## [937] 0.9939586 1.0202508 0.9949723 0.9752163 0.9903696 0.9983761
## [943] 1.0134305 0.9798859 0.9664352 1.0257658 0.9947124 0.9857610
## [949] 1.0138829 1.0261154 1.0224419 1.0171514 1.0265974 0.9658472
## [955] 0.9561873 0.9667746 1.0047466 1.0012268 1.0040583 0.9973018
## [961] 1.0234814 1.0125531 0.9631030 1.0251577 1.0006833 0.9799671

```

```
## [967] 0.9828383 1.0138725 1.0576528 1.0082964 0.9553690 0.9863707
## [973] 0.9698424 0.9756623 1.0033271 0.9726391 1.0533129 0.9890205
## [979] 0.9826147 1.0156717 0.9550786 0.9998255 1.0064612 0.9892574
## [985] 1.0394385 1.0202584 0.9499099 1.0261573 0.9887988 0.9699370
## [991] 1.0659430 0.9742740 1.0338970 0.9868711 1.0002585 1.0093474
## [997] 1.0116961 0.9637653 1.0235395 0.9733999
```

```
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(sum((x - 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.mean
```

```
## [1] 0.8858289
```

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

```
## [1] 0.4631495
```

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