Лабораторная работа №3 «Статистические гипотезы»
Выполнил: Саратовцев Артем 18Пи-1
Цель работы: проверка статистических гипотез с использованием встроенных в базовую версию пакета R функций, а также некоторого ряда критериев.
ТЕОРИЯ:
Непрерывное равномерное распределение – это распределение случайной вещественной величины, принимающей значения, принадлежащие некоторому промежутку конечной длины, характеризующееся тем, что плотность вероятности на этом промежутке почти везде постоянна.
Критерий однородности Смирнова – используется для проверки гипотезы о принадлежности двух независимых выборок одному закону распределения, то есть о том, что два эмпирических распределения соответствуют одному и тому же закону.

Критерий Колмогорова

$$\lambda = \sqrt{n}\, sup|F_э\ (x_i) - F_т\ (x_i)|$$

Критерий Колмогорова-Смирнова

$$\lambda = \sqrt{\frac{n_1 n_2}{n_1 + n_2}}\, sup|F_{э1}\ (x_i) - F_{2э}\ (x_i)|$$

Критерий согласия Колмогорова – предназначен для проверки гипотезы о принадлежности выборки некоторому закону распределения, то есть проверки того, что эмпирическое распределение соответствует предполагаемой модели.

Распределение Стьюдента – это однопараметрическое семейство абсолютно непрерывных распределений.
Одновыборочный t-критерий – этот вариант критерия Стьюдента служит для проверки нулевой гипотезы о равенстве среднего значения генеральной совокупности, из которой была взята выборка, некоторому известному значению. В общем виде проверка этой гипотезы выполняется при помощи t-критерия, который рассчитывается как отношение разницы между выборочным средним и известным значением к стандартной ошибке выборочного среднего. Рассчитанное значение критерия мы можем далее интерпретировать следующим образом, исходя из свойств t-распределения: если это значение попадает в область отклонения нулевой гипотезы, то мы вправе отклонить проверяемую нулевую гипотезу. Область отклонения нулевой гипотезы для критерия Стьюдента определяется заранее принятым уровнем значимости и числом степеней свободы.

$$T_S = \frac{\bar{x} - a}{\hat{s}_x}\sqrt{n}$$

Двухвыборочный t-критерий – при сравнении двух выборок проверяемая нулевая гипотеза состоит в том, что обе эти выборки происходят из нормально распределенных генеральных совокупностей с одинаковыми средними значениями. Эти генеральные средние мы оцениваем при помощи выборочных средних значений. В знаменателе приведенной формулы находится стандартная ошибка разницы между выборочными средними.

$$T_S = (\bar{x} - \bar{y})\sqrt{\frac{mn(m+n-2)}{(m+n)((m-1)\hat{s}_x^2 + (n-1)\hat{s}_y^2)}}$$

Значение **p-value** – вероятность получить для данной вероятностной модели распределения значений случайной величины такое же или более экстремальное значение статистики (среднего арифметического, медианы и др.), по сравнению с ранее наблюдаемым, при условии, что нулевая гипотеза верна.

Выборочное среднее – это приближение теоретического среднего распределения, основанное на выборке из него.

Экспоненциальное распределение – описывает интервалы времени между независимыми событиями, происходящими со средней интенсивностью. Количество наступлений такого события за некоторый отрезок времени описывается дискретным распределением Пуассона:

$$P_n(m) \approx \frac{\lambda^m}{m!}e^{-\lambda}$$

Критерий согласия Пирсона – это непараметрический метод, который позволяет оценить значимость различий между фактическим (выявленным в результате исследования) количеством исходов или качественных характеристик выборки, попадающих в каждую категорию, и теоретическим количеством, которое можно ожидать в изучаемых группах при справедливости нулевой гипотезы.

$$\mathrm{X}^2(k-1) = \sum_{i=1}^{k}\frac{n}{p_i}\left(\frac{n_i}{n} - p_i\right)^2$$

n_i - число повторений в выборке числа x_i

p_i - теоретическая вероятность появления числа x_i

n – объем выборки; $k \le n$

Уровень значимости статистического теста – допустимая для данной задачи вероятность ошибки первого рода, то есть вероятность отклонить нулевую гипотезу, когда на самом деле она верна.

Функция **lines()**:

lines(x, col = , lwb =)

значения:

x – значения,

col – цвет,

lwd – толщина линии,

служит для создания линий на графике.

Функция **mean(x)**:

находит среднее арифметическое элементов объекта **x**.

Функция **plot()**:

значения:

x, **y –** координаты точек на графике,

type – указывает на тип графика (в данной работе используется тип “l”, lines),

xlab – заголовок для оси x,

ylab – заголовок для оси y,

main – общее название графика,

служит для построения графиков.

Функция **seq(from, to, by =)**:

генерирует последовательность чисел от **from** до **to** с шагом **by**.

Функция **runif()**:

runif(n, min = , max =)

значения:
n – объем выборки,
min и **max** – нижний и верхний пределы распределения, которые должны быть представлены в виде целых чисел,
равномерное распределение.

Функция **ks.test()**:
ks.test(x, y, …, alternative = c("two.sided", "less", "greater"), exact = NULL)
значения:
x – вектор, содержащий выборку,
y – вектор, содержащий вторую выборку, или символьная строка с именем распределения,
… – параметры распределения,
alternative – символьный аргумент, обозначающий тип альтернативной гипотезы (принимает одно из значений: **"two.sided"** (по умолчанию), **"less"** или **"greater"**),
exact – NULL или логическое значение, обозначающее требуется ли точное вычисление p-value (не используется, если alternative = "less" или alternative = " greater "),
выполняет тесты Колмогорова-Смирнова. Возвращает значения статистики Колмогорова-Смирнова и строку с описанием гипотезы.

Функция **sd()**:
sd(x, na.rm = FALSE)
значения:
x – числовой вектор,
na.rm – логический аргумент, указывающий на то, нужно ли исключать пропущенные значения,
вычисляет стандартное отклонение значений в **x**.

Функция **t.test()**:
t.test(x, y=NULL, alternative = c("two.sided", "less", "greater"), conf.level = 0.95, mu=NULL, paired= FALSE, var.equal=FASLSE)
значения:
t.test(formula, data, subset, na.action), где **x, y** – числовые векторы, первой и второй выборок,
alternative – альтернативная гипотез (может быть одна из **"two sided"** (по умолчанию) - двусторонняя критическая область, **"greater"** - правосторонняя критическая область или **"less"** – левосторонняя критическая область),
conf.level – доверительная вероятность для возвращаемого доверительного интервала,
paired – признак парного теста, проверяется гипотеза для x и y, оба вектора должны иметь одну и ту же длину,
formula – формула вида **lhs ~ rhs** , где **lhs** - числовой вектор, **rhs** - фактор с двумя классами,
data – матрица или таблица данных, из которых берутся данные для **formula,**
subset – вектор, определяющий используемое подмножество наблюдений,
na.action – функция, которая вызывается, как только в данных встретилось значение **NA**,
vat.equal – логическая переменная, указывающая, следует ли рассматривать две дисперсии как равные (если **TRUE**, то объединенная дисперсия используется для оценки дисперсии, в противном случае используется приближение Уэлча с приближенными степенями свободы),
тест Стьюдента.

Функция **sapply()**:
sapply(x, function, simplify = , USE.NAMES =)
значения
x – имя матрицы, массива или таблицы данных,
function – имя применяемой функции,

simplify – логический аргумент (нужно ли представлять выводимый результат в виде матрицы (TRUE) или вектора,

USE.NAMES – логический аргумент (если данный аргумент принимает значение TRUE и x символьного типа, то в качестве названий для вывода используется x),

используется в случаях, когда необходимо применить какую-либо функцию к каждому компоненту списка, но результат вывести в виде вектора.

Функция **abline()**:

abline(h = , v =)

значения:

h – значение для горизонтальной линии,

v – значение для вертикальной линии,

добавляет одну или несколько прямых линий через текущий график.

Функция **cut()**:

cut(x = , breaks =)

значения:

x – числовой вектор,

breaks – либо количество интервалов, либо вектор точек, по которым нужно делить вектор **x**,

делит вектор на равные интервалы.

Функция **hist( )**:

hist(x, breaks =)

значения:

x – переменная,

breaks – количество столбцов,

служит для создания гистограмм частот значений переменной x.

Функция **diff()**:

diff(x, lag = , differences =)

значения:

x – вектор значений,

lag – указывает задержку,

differences - позволяет указать порядок различий,

вычисляет различия между всеми последовательными значениями вектора.

Функция **rexp()**:

rexp(N, rate =)

значения:

N – объем выборки,

rate – параметр **λ**,

экспоненциальное распределение.

Функция **chisq.test()**:

chisq.test(x, y = NULL, p = rep(1/length(x), length(x)), rescale.p = FALSE)

значения:

x – вектор или матрица

y – вектор (игнорируется, если **x** матрица),

p – вектор, содержащий вероятности (должен иметь такую же длину, что и **x),**

rescale.p – логическое значение (если TRUE, то **p** при необходимости нормируется так, чтобы сумма его компонент была равна 1,

критерий согласия Пирсона.

Функция **rnorm()**:

rnorm(n, mean = , sd =)

значения:

n – количество элементов выборки,

mean – математическое ожидание,

sd – среднеквадратическое отклонение,

служит для случайной генерации совокупностей нормально распределенных чисел.

Функция **sort(x)**:

сортирует элементы объекта x по возрастанию.

Функции **rm()** и **gc()**:

нужды для удаления обьектов.

<u>ХОД РАБОТЫ</u>:

Задание 1

Для двух частей реализации выборки объема 100 случайной величины, распределенной по равномерному закону, с помощью критерия однородности Смирнова проверить гипотезы.

```
# task 1

N <- 100       # Кол-во элементов в выборке
Point <- 49    # Точка для разделения выборок на 2 части
min <- 0       # Левая граница интервала для выборки
max <- 1       # Правая граница интервала для выборки
alpha <- 0.05  # Уровень значимости

Z <- runif(N, min = min, max = max); Z
X <- Z[1:Point]; X        # Часть выборки Z объемом 49
Y <- Z[(Point+1):N]; Y    # Часть выборки Z объемом 51

# С помощью критерия однородности Смирнова проверяем
# нулевую гипотезу о равенстве средних при альтернативной:
H1.a <- ks.test(X, Y, alternative = "two.sided"); H1.a  # не равны (X != Y)
H1.b <- ks.test(X, Y, alternative = "less"); H1.b       # меньше   (X < Y)
H1.c <- ks.test(X, Y, alternative = "greater"); H1.c    # больше   (X > Y)

H1.a$p.value > alpha
H1.b$p.value > alpha
H1.c$p.value > alpha

# Очиска памяти
rm(list = ls())
gc()
```

Результат выполнения:

```
> # С помощью критерия однородности Смирнова проверяем
> # нулевую гипотезу о равенстве средних при альтернативной:
> H1.a <- ks.test(X, Y, alternative = "two.sided"); H1.a  # не равны (X != Y)

        Two-sample Kolmogorov-Smirnov test

data:  X and Y
D = 0.19408, p-value = 0.2522
alternative hypothesis: two-sided

> H1.b <- ks.test(X, Y, alternative = "less"); H1.b       # меньше   (X < Y)

        Two-sample Kolmogorov-Smirnov test

data:  X and Y
D^- = 0.19408, p-value = 0.1522
alternative hypothesis: the CDF of x lies below that of y

> H1.c <- ks.test(X, Y, alternative = "greater"); H1.c    # больше   (X > Y)

        Two-sample Kolmogorov-Smirnov test

data:  X and Y
D^+ = 0.058824, p-value = 0.8412
alternative hypothesis: the CDF of x lies above that of y

> H1.a$p.value > alpha
[1] TRUE
> H1.b$p.value > alpha
[1] TRUE
> H1.c$p.value > alpha
[1] TRUE
```

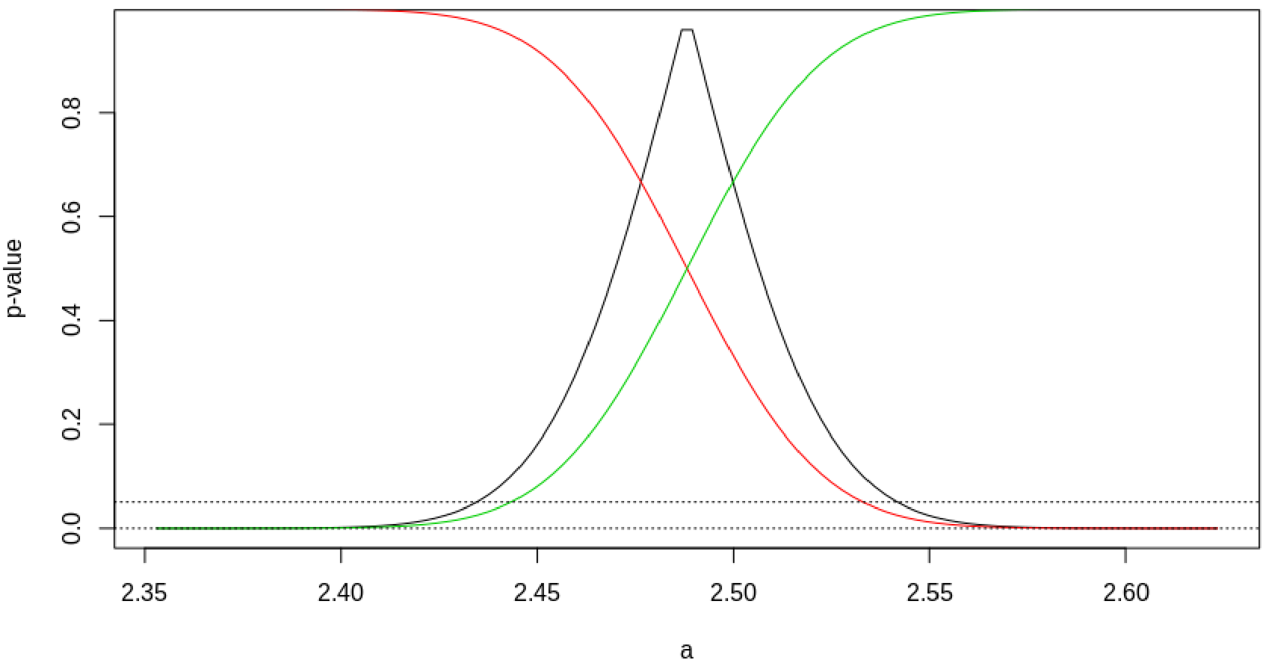
Задание 2

Для реализации выборки объемом 100 из генеральной совокупности случайной величины, распределенной по равномерному закону, проанализировать с помощью одновыборочного критерия Стьюдента зависимости для достигаемого уровня значимости α для нулевой гипотезы, а также построить графики.

```
# task 2

# Фукнция для применения t.test к вектору значений
# Возвращает уровень значимости по критерию Стьюдента
tTest <- function(a, x, alternative){
  #res <- sapply(a, function(param) t.test(x, mu = param, alternative = alternative)$p.value)
  res <- sapply(a, function(param) t.test(x, mu = param, alternative = alternative)[[3]])
  return(res)
}


N <- 100       # Кол-во элементов в выборке
min <- 2       # Левая граница интервала для выборки
max <- 3       # Правая граница интервала для выборки
alpha <- 0.05  # Уровень значимости

Z <- runif(N, min = min, max = max); Z
X <- mean(Z); X   # Выборочная средняя
S <- sd(Z); S     # Стандартное отклонение

# Найдем границы интервала для параметра a
left  <- X - S/2; left
right <- X + S/2; right

a <- seq(left, right, length = N); a

not_equal <- tTest(a, Z, "two.sided"); not_equal
less <- tTest(a, Z, "less"); less
greater <- tTest(a, Z, "greater"); greater

plot(a, not_equal, type = "l", col = 1,
     xlab = "a",
     ylab = "p-value")
lines(a, less, type = "l", col = 2)
lines(a, greater, type = "l", col = 3)

abline(h = c(0, alpha), lty = 3)

# Очиска памяти
rm(list = ls())
gc()
```

Результат выполнения:

```
> N <- 100       # Кол-во элементов в выборке
> min <- 2       # Левая граница интервала для выборки
> max <- 3       # Правая граница интервала для выборки
> alpha <- 0.05  # Уровень значимости
> Z <- runif(N, min = min, max = max); Z
  [1] 2.736909 2.366441 2.638997 2.063583 2.311951 2.964389 2.318436 2.314955 2.076560 2.642904 2.383344 2.184371 2.673484 2.181238 2.224746 2.833832 2.772994 2.070240 2.228028 2.262799
 [21] 2.759166 2.501112 2.405545 2.037675 2.825486 2.059745 2.223627 2.307087 2.916041 2.620456 2.541059 2.242658 2.337542 2.626507 2.448588 2.008499 2.717595 2.561186 2.809195 2.490953
 [41] 2.673548 2.548108 2.724308 2.001738 2.805322 2.935184 2.517647 2.452874 2.825854 2.515898 2.600750 2.717140 2.383191 2.125327 2.328519 2.196489 2.208766 2.524372 2.196217 2.801364
 [61] 2.061645 2.847921 2.595703 2.277846 2.454096 2.515212 2.670293 2.118627 2.370371 2.619279 2.069907 2.547056 2.988038 2.895119 2.128920 2.040950 2.150322 2.279510 2.209875 2.746956
 [81] 2.525183 2.115221 2.745189 2.769886 2.709582 2.470398 2.231987 2.761744 2.635100 2.741084 2.591802 2.711331 2.838856 2.701237 2.939975 2.472133 2.786902 2.692466 2.597651 2.419669
> X <- mean(Z); X   # Выборочная средняя
[1] 2.488176
> S <- sd(Z); S     # Стандартное отклонение
[1] 0.2702809
> # Найдем границы интервала для параметра a
> left  <- X - S/2; left
[1] 2.353035
> right <- X + S/2; right
[1] 2.623316
> a <- seq(left, right, length = N); a
  [1] 2.353035 2.355765 2.358496 2.361226 2.363956 2.366686 2.369416 2.372146 2.374876 2.377606 2.380336 2.383067 2.385797 2.388527 2.391257 2.393987 2.396717 2.399447 2.402177 2.404907
 [21] 2.407638 2.410368 2.413098 2.415828 2.418558 2.421288 2.424018 2.426748 2.429478 2.432209 2.434939 2.437669 2.440399 2.443129 2.445859 2.448589 2.451319 2.454049 2.456780 2.459510
 [41] 2.462240 2.464970 2.467700 2.470430 2.473160 2.475890 2.478620 2.481351 2.484081 2.486811 2.489541 2.492271 2.495001 2.497731 2.500461 2.503191 2.505922 2.508652 2.511382 2.514112
 [61] 2.516842 2.519572 2.522302 2.525032 2.527762 2.530493 2.533223 2.535953 2.538683 2.541413 2.544143 2.546873 2.549603 2.552333 2.555064 2.557794 2.560524 2.563254 2.565984 2.568714
 [81] 2.571444 2.574174 2.576904 2.579635 2.582365 2.585095 2.587825 2.590555 2.593285 2.596015 2.598745 2.601475 2.604206 2.606936 2.609666 2.612396 2.615126 2.617856 2.620586 2.623316
> not_equal <- tTest(a, Z, "two.sided"); not_equal
  [1] 2.481396e-06 3.760572e-06 5.672954e-06 8.517341e-06 1.272565e-05 1.891817e-05 2.797970e-05 4.116362e-05 6.023274e-05 8.764812e-05 1.268195e-04 1.824340e-04 2.608829e-04 3.708085e-04
 [15] 5.237965e-04 7.352421e-04 1.025415e-03 1.420752e-03 1.955397e-03 2.673009e-03 3.628827e-03 4.891996e-03 6.548097e-03 8.701827e-03 1.147972e-02 1.503278e-02 1.953882e-02 2.520430e-02
 [29] 3.226546e-02 4.098832e-02 5.166743e-02 6.462296e-02 8.019596e-02 9.874155e-02 1.206200e-01 1.461858e-01 1.757748e-01 2.096895e-01 2.481844e-01 2.914496e-01 3.395964e-01 3.926431e-01
 [43] 4.505044e-01 5.129835e-01 5.797680e-01 6.504314e-01 7.244389e-01 8.011587e-01 8.798777e-01 9.598217e-01 9.598217e-01 8.798777e-01 8.011587e-01 7.244389e-01 6.504314e-01 5.797680e-01
 [57] 5.129835e-01 4.505044e-01 3.926431e-01 3.395964e-01 2.914496e-01 2.481844e-01 2.096895e-01 1.757748e-01 1.461858e-01 1.206200e-01 9.874155e-02 8.019596e-02 6.462296e-02 5.166743e-02
 [71] 4.098832e-02 3.226546e-02 2.520430e-02 1.953882e-02 1.503278e-02 1.147972e-02 8.701827e-03 6.548097e-03 4.891996e-03 3.628827e-03 2.673009e-03 1.955397e-03 1.420752e-03 1.025415e-03
 [85] 7.352421e-04 5.237965e-04 3.708085e-04 2.608829e-04 1.824340e-04 1.268195e-04 8.764812e-05 6.023274e-05 4.116362e-05 2.797970e-05 1.891817e-05 1.272565e-05 8.517341e-06 5.672954e-06
 [99] 3.760572e-06 2.481396e-06
> less <- tTest(a, Z, "less"); less
  [1] 9.999988e-01 9.999981e-01 9.999972e-01 9.999957e-01 9.999936e-01 9.999905e-01 9.999860e-01 9.999794e-01 9.999699e-01 9.999562e-01 9.999366e-01 9.999088e-01 9.998696e-01 9.998146e-01
 [15] 9.997381e-01 9.996324e-01 9.994873e-01 9.992896e-01 9.990223e-01 9.986635e-01 9.981856e-01 9.975540e-01 9.967260e-01 9.956491e-01 9.942601e-01 9.924836e-01 9.902306e-01 9.873978e-01
 [29] 9.838673e-01 9.795058e-01 9.741663e-01 9.676885e-01 9.599020e-01 9.506292e-01 9.396900e-01 9.269071e-01 9.121126e-01 8.951552e-01 8.759078e-01 8.542752e-01 8.302018e-01 8.036785e-01
 [43] 7.747478e-01 7.435082e-01 7.101160e-01 6.747843e-01 6.377806e-01 5.994207e-01 5.600612e-01 5.200892e-01 4.799108e-01 4.399388e-01 4.005793e-01 3.622194e-01 3.252157e-01 2.898840e-01
 [57] 2.564918e-01 2.252522e-01 1.963215e-01 1.697982e-01 1.457248e-01 1.240922e-01 1.048448e-01 8.788739e-02 7.309292e-02 6.031001e-02 4.937078e-02 4.009798e-02 3.231148e-02 2.583371e-02
 [71] 2.049416e-02 1.613273e-02 1.260215e-02 9.769408e-03 7.516390e-03 5.739860e-03 4.350913e-03 3.274048e-03 2.445998e-03 1.814413e-03 1.336504e-03 9.776985e-04 7.103758e-04 5.127074e-04
 [85] 3.676211e-04 2.618983e-04 1.854042e-04 1.304415e-04 9.121702e-05 6.340975e-05 4.382406e-05 3.011637e-05 2.058181e-05 1.398985e-05 9.459086e-06 6.362827e-06 4.258671e-06 2.836477e-06
 [99] 1.880286e-06 1.240698e-06
> greater <- tTest(a, Z, "greater"); greater
  [1] 1.240698e-06 1.880286e-06 2.836477e-06 4.258671e-06 6.362827e-06 9.459086e-06 1.398985e-05 2.058181e-05 3.011637e-05 4.382406e-05 6.340975e-05 9.121702e-05 1.304415e-04 1.854042e-04
 [15] 2.618983e-04 3.676211e-04 5.127074e-04 7.103758e-04 9.776985e-04 1.336504e-03 1.814413e-03 2.445998e-03 3.274048e-03 4.350913e-03 5.739860e-03 7.516390e-03 9.769408e-03 1.260215e-02
 [29] 1.613273e-02 2.049416e-02 2.583371e-02 3.231148e-02 4.009798e-02 4.937078e-02 6.031001e-02 7.309292e-02 8.788739e-02 1.048448e-01 1.240922e-01 1.457248e-01 1.697982e-01 1.963215e-01
 [43] 2.252522e-01 2.564918e-01 2.898840e-01 3.252157e-01 3.622194e-01 4.005793e-01 4.399388e-01 4.799108e-01 5.200892e-01 5.600612e-01 5.994207e-01 6.377806e-01 6.747843e-01 7.101160e-01
 [57] 7.435082e-01 7.747478e-01 8.036785e-01 8.302018e-01 8.542752e-01 8.759078e-01 8.951552e-01 9.121126e-01 9.269071e-01 9.396900e-01 9.506292e-01 9.599020e-01 9.676885e-01 9.741663e-01
 [71] 9.795058e-01 9.838673e-01 9.873978e-01 9.902306e-01 9.924836e-01 9.942601e-01 9.956491e-01 9.967260e-01 9.975540e-01 9.981856e-01 9.986635e-01 9.990223e-01 9.992896e-01 9.994873e-01
 [85] 9.996324e-01 9.997381e-01 9.998146e-01 9.998696e-01 9.999088e-01 9.999366e-01 9.999562e-01 9.999699e-01 9.999794e-01 9.999860e-01 9.999905e-01 9.999936e-01 9.999957e-01 9.999972e-01
 [99] 9.999981e-01 9.999988e-01
```



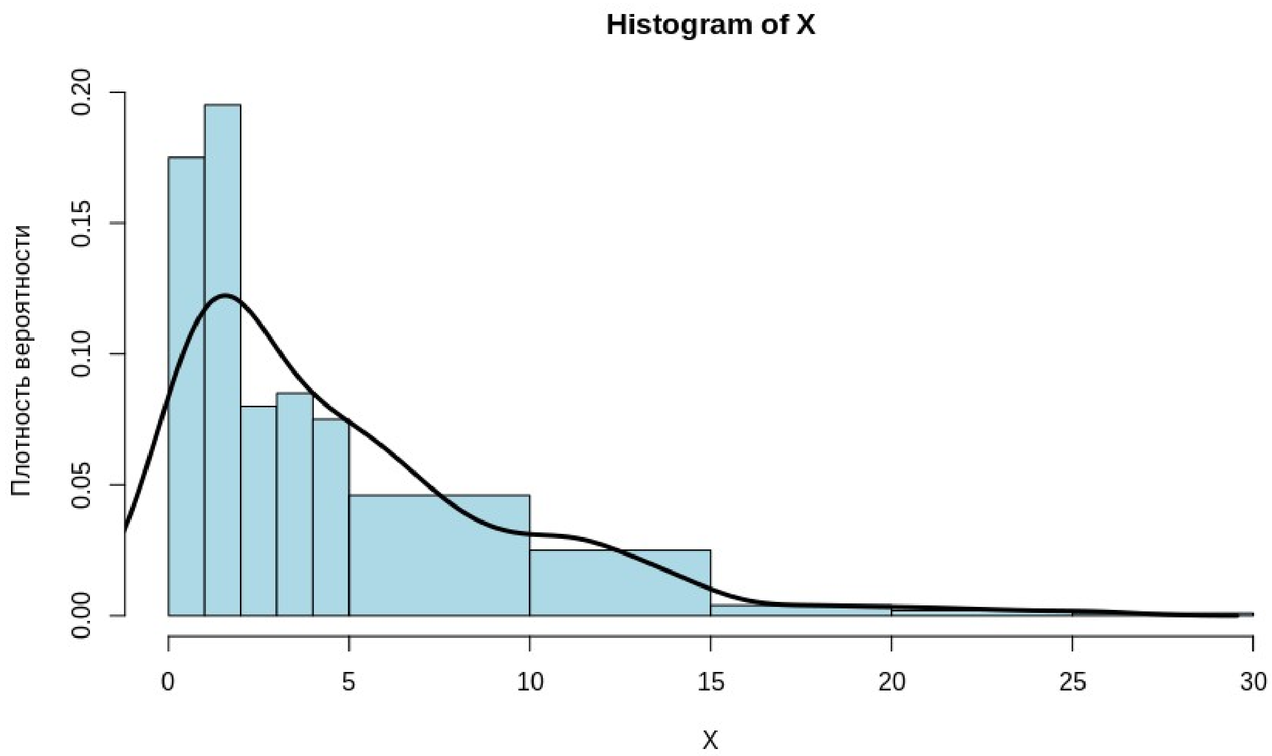
Задание 3

Для выборки объемом 200 из генеральной совокупности случайной величины, распределенной по экспоненциальному закону с параметром, проверить гипотезы.

```r
# task 3

l <- 0.2        # Параметр для експоненциальному распределению
N <- 200        # Кол-во элементов в выборке
alpha <- 0.05  # Уровень значимости

X <- rexp(N, l); X  # Создаем выборку

# Задаем границы интервалы и создаем таблицу представлений в виде интервалов
breaks <- c(0, 1, 2, 3, 4, 5, 10, 15, 20, 25, 30)
tbl    <- table(cut(X, breaks = breaks)); tbl

# Рисуем гистограмму экспоненциального распредления случайной величины
hist(X, breaks = breaks,
     col = "lightblue",
     xlab = "X",
     ylab = "Плотность вероятности")
lines(density(X), lwd = 3)

breaks <- c(-Inf, 1, 2, 3, 4, 5, 10, 15, 20, 25, Inf)
p      <- diff(pexp(breaks, l)); p; sum(p)

# Критерий Хи-квадрат (Критерий Пирсона)
res <- chisq.test(x = tbl, p = p)

# Проверка
res$p.value > alpha

# Очиска памяти
rm(list = ls())
gc()
```

Результат выполнения:

```
> # Задаем границы интервалы и создаем таблицу представлений в виде интервалов
> breaks <- c(0, 1, 2, 3, 4, 5, 10, 15, 20, 25, 30)
> tbl    <- table(cut(X, breaks = breaks)); tbl

  (0,1]   (1,2]   (2,3]   (3,4]   (4,5]  (5,10] (10,15] (15,20] (20,25] (25,30]
     35      39      16      17      15      46      25       4       2       1


> breaks <- c(-Inf, 1, 2, 3, 4, 5, 10, 15, 20, 25, Inf)
> p      <- diff(pexp(breaks, l)); p; sum(p)
 [1] 0.181269247 0.148410707 0.121508410 0.099482672 0.081449523 0.232544158 0.085548215 0.031471429 0.011577692 0.006737947
[1] 1


> # Критерий Хи-квадрат (Критерий Пирсона)
> res <- chisq.test(x = tbl, p = p)


> # Проверка
> res$p.value > alpha
[1] TRUE
```

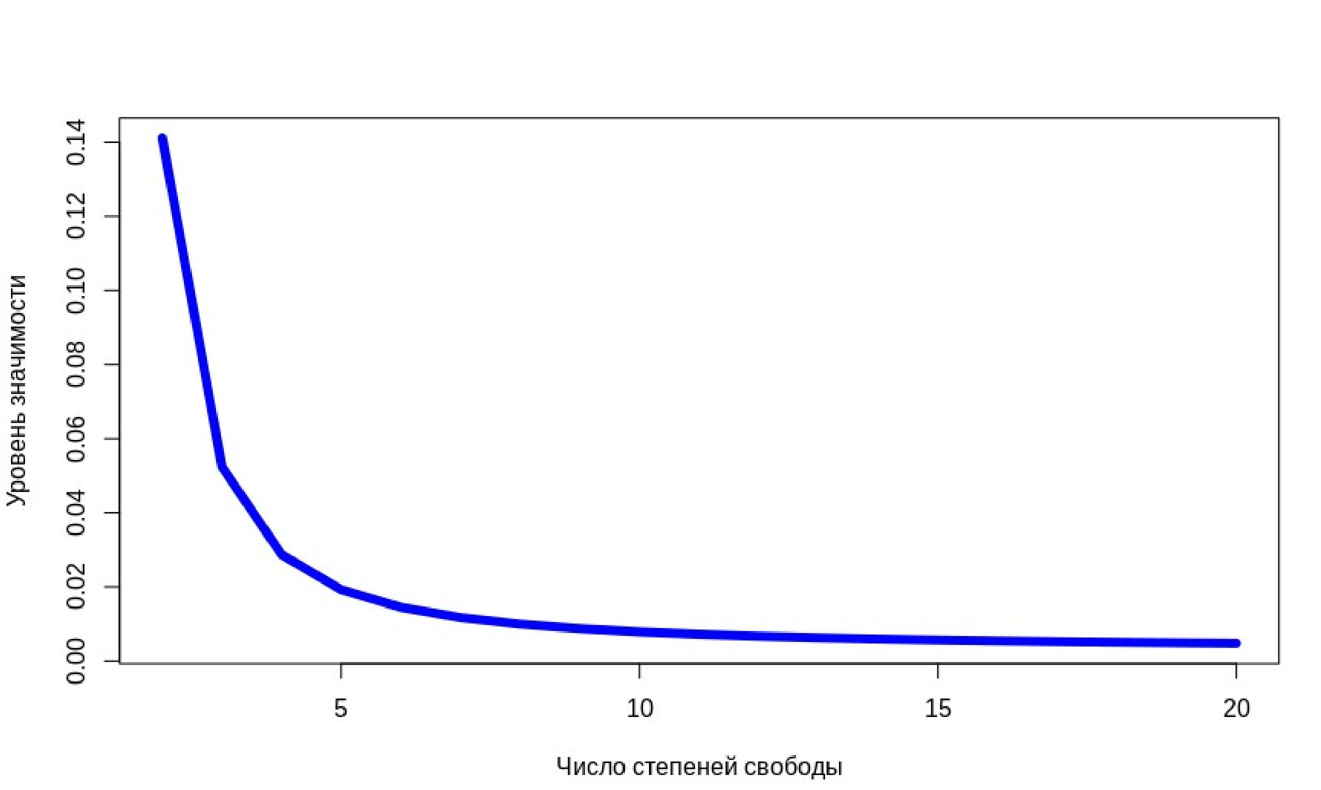


Задание 4

Для центрированной выборки объемом 100 из генеральной совокупности случайной величины, распределенной по нормальному закону, с помощью критерия согласия Колмогорова проанализировать зависимость достигнутого уровня значимости α от числа степеней свободы (построить график и проанализировать его) для нулевой гипотезы.

```
# task 4

# Фукнция для применения ks.test к вектору значений
# Возвращает достигнутый уровень значимости от числа степеней свободы
ksTest <- function(m, X) {
  sapply(m, function(param) ks.test(X, "pt", param) [[2]])
}

N  <- 100      # Кол-во элементов в выборке
a  <- 8        # Мат ожидание
sd <- 2        # Стандартное отклонение
m  <- c(2: 20) # Число степеней свободы

# Создаем выборку нормального распределения
X <- rnorm(N, a, sd); X

# Центрируем выборку
X <- sort(X - mean(X)); X

levels <- ksTest(m, X)

plot(m, levels, type = "l", col = 4, lwd = 6,
     xlab = "Число степеней свободы",
     ylab = "Уровень значимости")
```

Результат выполнения:



<u>ВЫВОД</u>: в ходе работы я осуществил проверки статистических гипотез на разных наборах данных (выборках), что помогло мне закрепить теоретический материал из математической статистики и на практике разобраться в новых для меня функциях стандартного пакета R, а также познакомиться с механизмом их работы.

Конец