Лабораторная работа №3

«Статистические гипотезы»

Выполнил: Николай Окуньков, 18ПИ-2.

Цель работы: проверка статистических гипотез с использованием встроенных в базовую версию пакета R функций, а также некоторого ряда критериев.

Теоретическая часть:

Непрерывное равномерное распределение – распределение случайной вещественной величины, принимающей значения, принадлежащие некоторому промежутку конечной длины. Характеризуется 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-Смирнова

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

Критерий Колмогорова

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

Критерий согласия Колмогорова – используется 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* – один из вариантов критерия Стьюдента, служащий 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, рассчитывающийся как отношение разницы между выборочным средним и известным значением к стандартной ошибке выборочного среднего. Исходя из свойств t-распределения, рассчитанное значение критерия можно интерпретировать таким образом: если данное 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. Данные генеральные средние оценивают 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 – приближение теоретического среднего распределения, которое основывается на выборке из него.

Экспоненциальное распределение – описывает интервалы времени между независимыми событиями, которые происходят 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 – непараметрический метод, позволяющий 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:

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

где $\boldsymbol{n_i}$ – число повторений в выборке числа $\boldsymbol{x_i}$; $\boldsymbol{p_i}$ – теоретическая вероятность появления числа $\boldsymbol{x_i}$; $\boldsymbol{n}$ – объём выборки; $\boldsymbol{k \leq n}$.

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

Используемые в данной работе функции:

lines(), mean(x), plot(), **seq(from, to, by =), runif(), ks.test(), sd(), t.test(), sapply(), abline(), cut()**, **hist( ), diff(), rexp(), chisq.test(), rnorm()**, **sort(x):**

lines() – функция служит для создания линий на графике;

lines(x, col = , lwb =)

(Параметры: **x** – значения; **col** – цвет; **lwd** – толщина линии.)

mean(x) – функция, находящая среднее арифметическое элементов объекта **x**;

plot() – функция, служащая для построения графиков;

(Параметры: **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.)

seq(from, to, by =) – функция, генерирующая последовательность чисел от **from** до **to** с шагом **by**;

runif() – функция для равномерного распределения;

runif(n, min = , max =)

(Параметры: **n** – объем выборки; **min** и **max** – нижний и верхний пределы распределения, которые должны быть представлены в виде целых чисел.)

ks.test() – функция для выполнения тестов Колмогорова-Смирнова. Возвращает значения статистики Колмогорова-Смирнова и строку с описанием гипотезы;

ks.test(x, y, …, alternative = c(“two.sided”, “less”, “greater”), exact = NULL)

(Параметры: **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 ”.**)

sd() – функция для вычисления стандартного отклонения значений в **x**;

sd(x, na.rm = FALSE)

(Параметры: **x** – числовой вектор; **na.rm –** логический аргумент, указывающий на то, нужно ли исключать пропущенные значения.)

t.test() – функция, выполняющая тест Стьюдента;

t.test(x, y=NULL, alternative = c("two.sided", "less", "greater"), conf.level = 0.95, mu=NULL, paired= FALSE, var.equal=FASLSE)

(Параметры: **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**; **var.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).)

sapply() – функция, используемая в случаях, когда необходимо применить какую-либо функцию к каждому компоненту списка, но результат вывести в виде вектора;

sapply(x, function, simplify = , USE.NAMES =)

(Параметры: **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**).)

abline() – функция добавляет одну или несколько прямых линий через текущий график;

abline(h = , v =)

(Параметры: **h** – значение для горизонтальной линии; **v** – значение для вертикальной линии.)

cut() – функция делит вектор на равные интервалы;

cut(x = , breaks =)

(Параметры: **x** – числовой вектор; **breaks** – либо количество интервалов, либо вектор точек, по которым нужно делить вектор **x**.)

hist( ) – функция для создания гистограмм частот значений переменной **x**;

hist(x, breaks =)

(Параметры: **x** – переменная; **breaks** – количество столбцов.)

diff() – функция для вычисления различий между всеми последовательными значениями вектора;

diff(x, lag = , differences =)

(Параметры: **x** – вектор значений; **lag –** указывает задержку; **differences -** позволяет указать порядок различий.)

rexp() – функция экспоненциального распределения;

rexp(N, rate =)

(Параметры: **N** – объем выборки; **rate** – параметр **λ**.)

chisq.test() – функция критерия согласия Пирсона;

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

(Параметры: **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.)

rnorm() – функция для случайной генерации совокупностей нормально распределенных чисел;

rnorm(n, mean = , sd =)

(Параметры: **n –** количество элементов выборки; **mean –** математическое ожидание; **sd –** среднеквадратическое отклонение.)

sort(x) – функция для сортировки элементов объекта x по возрастанию.

Ход работы:

Задание 1:

Для двух частей реализации выборки объема N=100 случайной величины, распределенной по равномерному закону в интервале [0, 1], с помощью критерия однородности Смирнова на уровне значимости α = 0.05 проверить нулевую гипотезу H_0: $\bar{x} = \bar{y}$, при альтернативной H_1: а) $\bar{x} \neq \bar{y}$, б) $\bar{x} < \bar{y}$, в) $\bar{x} > \bar{y}$.

z – выборка объемом 100; x – часть выборки z объемом 49, y – часть выборки z объемом 51.

```
N <- 100  # кол-во элементов в выборке
sign.lvl <- 0.05  # уровень значимости

# Создадим выбоки и разделим их на две части по 49 и 51 элементам
z <- runif(N, min = 0, max = 1); z # создаем выборку
x <- z[1:49]; x
y <- z[50:100]; y

# Проверка нулевой гипотезы с помощью критерия Колмагорова-Смирнова
h1.1 <- ks.test(X, Y, alternative = "two.sided"); h1.1
h1.2 <- ks.test(X, Y, alternative = "less"); h1.2
h1.3 <- ks.test(X, Y, alternative = "greater"); h1.3

# Проверка
h1.1.a$p.value > sign.lvl
h1.2$p.value > sign.lvl
h1.3$p.value > sign.lvl

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

Результат:

```
> # Проверка нулевой гипотезы с помощью критерия Колмагорова-Смирнова
> h1.1 <- ks.test(x, y, alternative = "two.sided"); h1.1

        Two-sample Kolmogorov-Smirnov test

data:  x and y
D = 0.17447, p-value = 0.3664
alternative hypothesis: two-sided

> h1.2 <- ks.test(x, y, alternative = "less"); h1.2

        Two-sample Kolmogorov-Smirnov test

data:  x and y
D^- = 0.17447, p-value = 0.2184
alternative hypothesis: the CDF of x lies below that of y

> h1.3 <- ks.test(x, y, alternative = "greater"); h1.3

        Two-sample Kolmogorov-Smirnov test

data:  x and y
D^+ = 0.067627, p-value = 0.7957
alternative hypothesis: the CDF of x lies above that of y

> # Проверка
> h1.1$p.value > sign.lvl
[1] TRUE
> h1.2$p.value > sign.lvl
[1] TRUE
> h1.3$p.value > sign.lvl
[1] TRUE
```

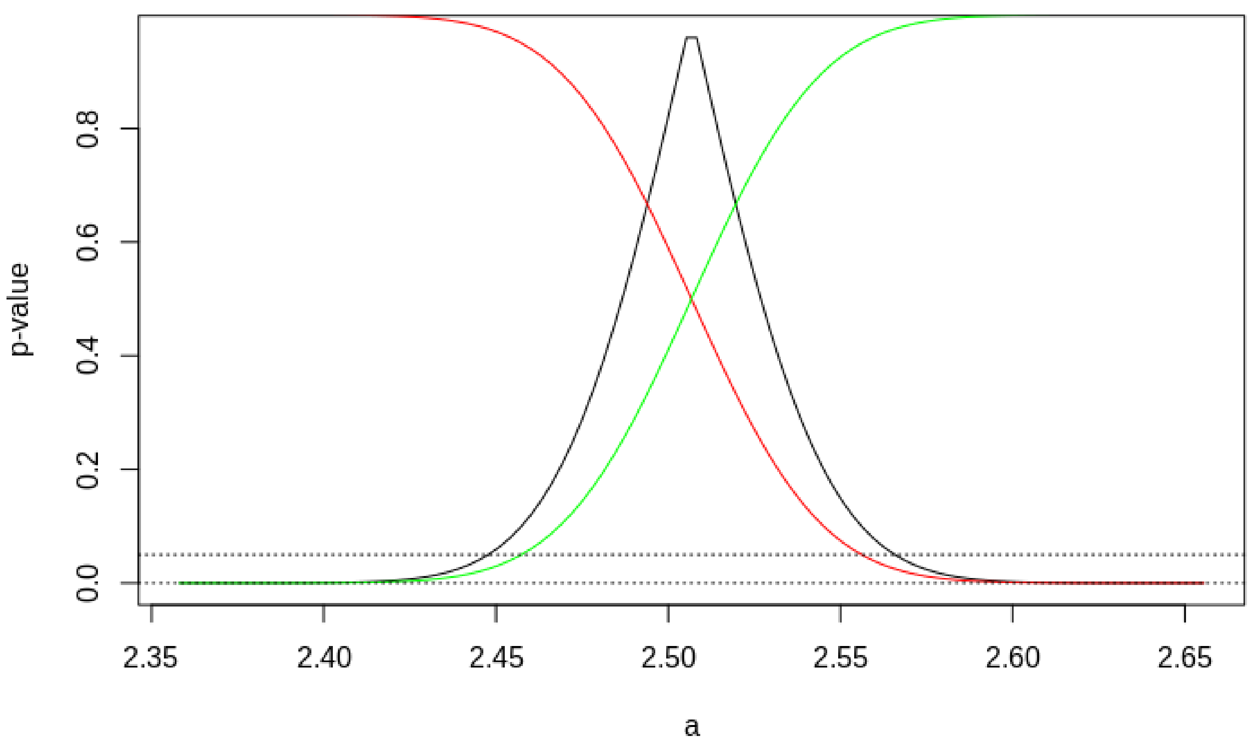
Задание 2:

Для реализации выборки объемом 100 из генеральной совокупности случайной величины, распределенной по равномерному закону в интервале [2, 3], проанализировать с помощью одновыборочного критерия Стьюдента зависимости для достигаемого уровня значимости α от значения параметра $a \in [\bar{x} - \frac{\hat{s}}{2}; \bar{x} + \frac{\hat{s}}{2}]$ для нулевой гипотезы H_0: $\bar{x} = a$ при альтернативной H_1: а) $\bar{x} \neq a$, б) $\bar{x} < a$, в) $\bar{x} > a$. Построить графики (на одном рисунке для разных альтернативных гипотез) зависимости величины p-value от величины параметра a. На графике также построить границу принятия нулевой гипотезы для заданного уровня значимости $\alpha = 0{,}05$.

```r
sapplytTest <- function(array, X, alt){
  res <- sapply(array, function(param) t.test(X, mu = param, alternative = alt)[[3]])
}

N <- 100  # кол-во элементов в выборке
sign.lvl <- 0.05  # уровень значимости

z <- runif(N, min = 2, max = 3); z # создаем выборку
x <- mean(z); x # выборочная средняя
s <- sd(z); s   # стандартное отклонение

# Найдем параметры a
a <- seq(x-s/2, x+s/2, length = N); a

n_eq <- sapplytTest(a, z, "two.sided"); n_eq
l    <- sapplytTest(a, z, "less"); l
g    <- sapplytTest(a, z, "greater"); g

plot(a, n_eq, type = "l", col = "black",
     xlab = "a",
     ylab = "p-value")
lines(a, l, type = "l", col = "red")
lines(a, g, type = "l", col = "green")
abline(h = c(0, sign.lvl), lty = 3)

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

Результат:

```
> # Найдем параметры а
> a <- seq(x-s/2, x+s/2, length = N); a
  [1] 2.314803 2.317706 2.320608 2.323511 2.326414 2.329316 2.332219 2.335121 2.338024 2.340927 2.343829 2.346732 2.349635 2.352537 2.355440 2.358342 2.361245 2.364148 2.367050 2.369953 2.372855
 [22] 2.375758 2.378661 2.381563 2.384466 2.387369 2.390271 2.393174 2.396076 2.398979 2.401882 2.404784 2.407687 2.410590 2.413492 2.416395 2.419297 2.422200 2.425103 2.428005 2.430908 2.433811
 [43] 2.436713 2.439616 2.442518 2.445421 2.448324 2.451226 2.454129 2.457031 2.459934 2.462837 2.465739 2.468642 2.471545 2.474447 2.477350 2.480252 2.483155 2.486058 2.488960 2.491863 2.494766
 [64] 2.497668 2.500571 2.503473 2.506376 2.509279 2.512181 2.515084 2.517987 2.520889 2.523792 2.526694 2.529597 2.532500 2.535402 2.538305 2.541207 2.544110 2.547013 2.549915 2.552818 2.555721
 [85] 2.558623 2.561526 2.564428 2.567331 2.570234 2.573136 2.576039 2.578942 2.581844 2.584747 2.587649 2.590552 2.593455 2.596357 2.599260 2.602163
> n_eq <- sapplytTest(a, z, "two.sided"); n_eq
  [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
> l    <- sapplytTest(a, z, "less"); l
  [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
> g    <- sapplytTest(a, z, "greater"); g
  [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 X, распределенной по экспоненциальному закону с параметром $\lambda = 0{,}2$, проверить нулевую гипотезу H_0: $F(x) = F_0(x, \lambda_{\text{оц}})$ при альтернативной H_1: $F(x) \neq F_0(x, \lambda_{\text{оц}})$ по критерию согласия Пирсона на уровне значимости $\alpha = 0{,}05$.

Здесь $F_0(x, \lambda_{\text{оц}})$ – функция плотности экспоненциального закона распределения с параметром $\lambda_{\text{оц}}$. $\lambda_{\text{оц}}$ – оценка параметра λ , полученная по данным выборки.

```
l <- 0.2  # параметр для експоненциальному распределению
N <- 200  # кол-во элементов в выборке
sign.lvl <- 0.05  # уровень значимости

x <- rexp(N, l); x # создаем выборку
breaks <- c(-Inf, 1, 2, 3, 4, 5, 10, 15, 20, 25, Inf)
table  <- table(cut(x, breaks = breaks)); table
p      <- diff(pexp(breaks, l)); p; sum(p)

# Проверка гитотезы с помощью критерия Пирсона
h <- chisq.test(x = table, p = p); h$p.value
h$p.value > sign.lvl

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

Результат:

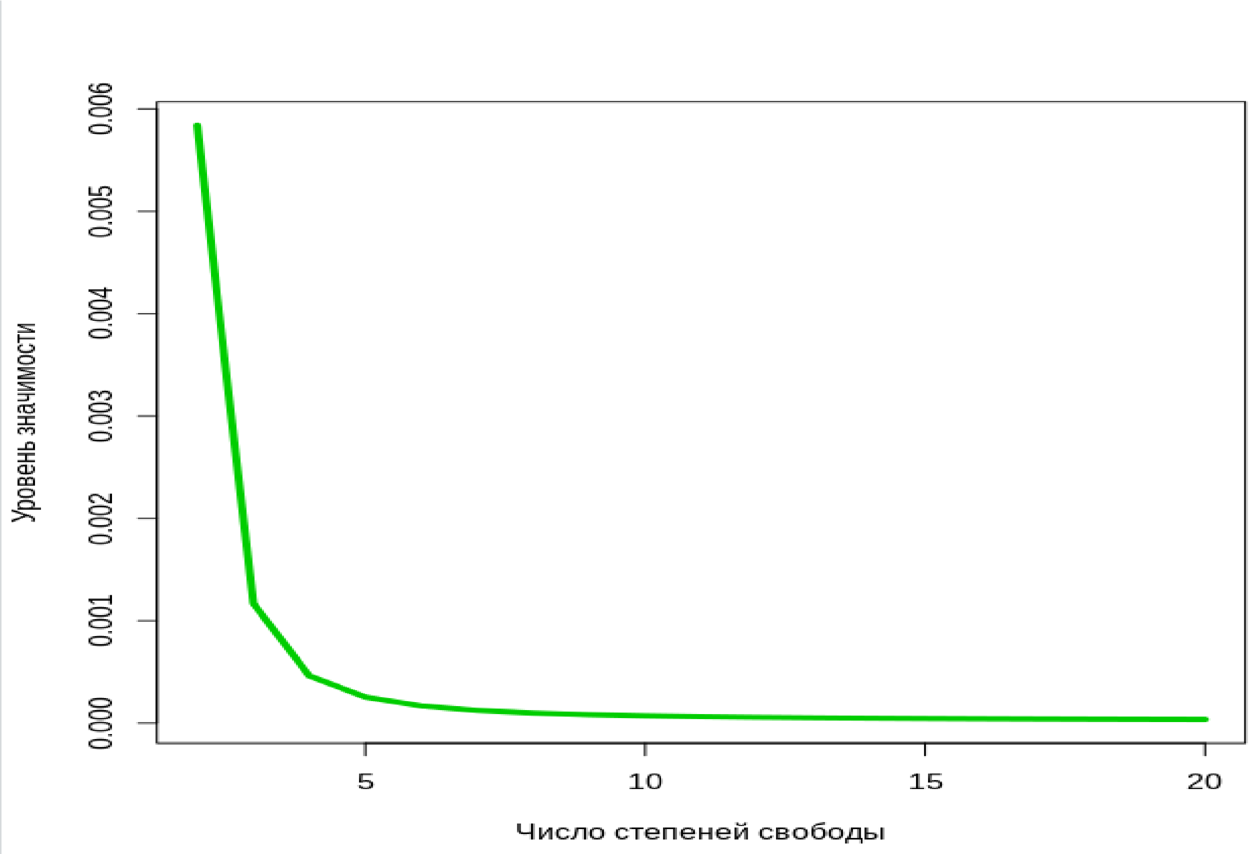
```
> x <- rexp(N, l); x # создаем выборку
  [1]  3.37887516  3.85023059  2.94606813  2.40141606  1.74671718  5.27049110  6.70707982  2.78822849  3.45937709
 [10] 16.67060722  4.09584624  5.19307471  1.49502680  1.97093744  0.95855523  2.23070199  2.79467698  7.53803069
 [19] 13.02063658  1.32219192  3.88168468  1.59423923  2.70178927  3.93379483  3.32946564  1.90054314  0.10366605
 [28]  5.20344672  0.88413209  3.04404825 17.56339930 10.74026530  0.01713163  9.14821209  1.10750918  6.67225906
 [37]  0.23488562  7.60480411  1.36635289  7.40976433  1.31819280  0.33683652  8.32941459  0.13481212  7.38995535
 [46]  0.61791211  0.36348477 16.04819458  6.54552861 18.32188281  9.44982543  2.51934949  4.24754291  1.23083911
 [55] 24.55113851  4.95086408  6.04736631  1.61763008  5.27876524 10.38409331  8.24688160  7.71207014 11.99483543
 [64]  2.92720058 11.54831156  2.40364161  6.07034032  4.36878469  0.22725833  3.61160804 14.33644757  5.12194998
 [73]  4.31343072  1.65661436  6.68500313  1.86344205  5.18433971  1.61701376  3.48148650  7.65791134 15.97060531
 [82]  6.73281220  4.26674736  3.12117946  5.06065459  2.38404528  2.58056240  6.33595322 13.34908617  2.32591292
 [91]  2.67234359  0.92412221  5.82914639  1.39359034 10.41672447 34.92591255 13.10272331  1.55299720 10.89731902
[100]  4.01812985  0.49273800  5.30243624  2.01276286  7.14640015  2.43905327  6.40997024  0.83266041 20.17365615
[109]  0.72116859  3.67339891  1.14174420  2.53767584  1.14241308  3.22120566  0.96424605  0.01770892  3.43383296
[118]  0.05497028  0.52766674  5.47820921  1.13781898  1.10230825  9.07717034  0.11583849  1.19158754  2.87201825
[127]  1.07481854  2.92341311  2.08509040  2.22102196  1.51126114  1.84460098  6.19786058 16.14336558  6.02120361
[136]  4.39286418  7.22956290  2.30957958  0.35205933  0.56200672  2.50521005  2.32696648  3.34889917  0.30628021
[145]  3.59212644  1.12581236  0.28218313  0.18785218  3.01449647  0.95323467  4.23065231  1.87216529  2.91568242
[154]  0.19008814  1.92556271  4.61087671 20.14373566  5.32196340  6.91783857  4.62619880  1.80544284 13.54621465
[163] 12.04325794 21.87206814 12.68905031  5.94715589  0.04458074  8.19900861  5.18417354  2.81614244  0.34107148
[172]  2.52843033  5.29276159  7.31032945 12.09217791  1.63528486  8.28450826  7.81907582 10.52021296  4.15017742
[181]  3.47341917  3.82889126  1.97244039  1.83520118  0.38429693  0.72742032  2.63519592  0.94879422  6.27862568
[190]  5.91399326  7.91438612  2.62603532  5.33261595  0.79641127  5.52961354  0.85730583  0.15206641  6.94881441
[199] 11.73618524  0.42413367
> breaks <- c(-Inf, 1, 2, 3, 4, 5, 10, 15, 20, 25, Inf)
> table  <- table(cut(x, breaks = breaks)); table

 (-Inf,1]    (1,2]    (2,3]    (3,4]    (4,5]   (5,10]  (10,15]  (15,20]  (20,25] (25, Inf]
       35       31       28       18       12       49       16        6        4         1
> 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
[10] 0.006737947
[1] 1
> # Проверка гипотезы с помощью критерия Пирсона
> h <- chisq.test(x = table, p = p); h$p.value
[1] 0.9406451
> h$p.value > sign.lvl
[1] TRUE
```

Задание 4:

Для центрированной выборки объемом 100 из генеральной совокупности случайной величины X, распределенной по нормальному закону с $a = 8$ и $sd = 2$, с помощью критерия согласия Колмогорова проанализировать зависимость достигнутого уровня значимости α от числа степеней свободы $m \in [2, 20]$ (построить график и проанализировать его) для нулевой гипотезы H_0: $F(x) = F_0(x, m)$ при альтернативной H_1: $F(x) \neq F_0(x, m)$, где $F_0(x, m)$ – теоретическая функция распределения Стьюдента с заданным числом степеней свободы m.

```r
sapplyksTest <- function(fdegrees, x){
  sapply(fdegrees, function(param) ks.test(x, "pt", param) [[2]])
}

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

# Создаем выборку нормального распределения
x <- rnorm(N, mean = mean, sd = sd); x

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

lvls <- sapplyksTest(fdegrees, x)
plot(fdegrees, lvls, type = "l", col = 3, lwd = 4,
     xlab = "Число степеней свободы",
     ylab = "Уровень значимости")
```

Результат:



Вывод

В ходе работы мной были проверены статистические гипотезы на разных наборах данных (выборках), что помогло мне усвоить теоретический материал, используемый в данной работе, узнать о новых функциях пакета R и ознакомиться с механизмом их работы.