МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ФГБОУ ВО «ИГУ»
Кафедра физико-химической биологии, биоинженерии и биоинформатики

УТВЕРЖДАЮ
Декан биолого-почвенного факультета
А.Н. Матвеев
«21» марта 2025 г.

ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

для проведения текущего контроля и промежуточной аттестации по дисциплине:

Б1.О.25 **«СПЕЦИАЛЬНЫЕ ГЛАВЫ МАТЕМАТИКИ»**

Специальность: 06.05.01 «Биоинженерия и биоинформатика»

Специализация: «Биоинженерия и биоинформатика»

Квалификация выпускника: биоинженер и биоинформатик

Форма обучения: очная с элементами электронного обучения и дистанционных образовательных технологий

Согласовано с УМК биолого-почвенного факультета
Протокол № 5 от 21 марта 2025 г.
Председатель _____ А.Н. Матвеев

Рекомендовано кафедрой физико-химической биологии, биоинженерии и биоинформатики
Протокол № 12 от 19 марта 2025 г.
Зав. кафедрой _____ В.П. Саловарова

Иркутск 2025 г.

ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

Разработан для учебной дисциплины Б1.О.25 «СПЕЦИАЛЬНЫЕ ГЛАВЫ МАТЕМАТИКИ» специальности 06.05.01 «Биоинженерия и биоинформатика», специализация «Биоинженерия и биоинформатика». Фонд оценочных материалов (ФОМ) включает оценочные материалы для проведения текущего контроля, промежуточной аттестации в форме экзамена.

Оценочные материалы соотнесены с требуемыми результатами освоения образовательной программы 06.05.01 «Биоинженерия и биоинформатика», в соответствии с содержанием рабочей программы учебной дисциплины Б1.О.25 «Специальные главы математики» с учетом ОПОП.

Нормативные документы, регламентирующие разработку ФОМ:

- статья 2, часть 9 Федерального закона «Об образовании в Российской Федерации», ФЗ-273, от 29.12.2012 г.;
- ФГОС ВО по специальности 06.05.01 «Биоинженерия и биоинформатика», утвержденный приказом Министерства науки и высшего образования Российской Федерации 12 августа 2020 г. № 973.

1. Компетенции, формируемые в процессе изучения дисциплины (2 курс, 3 семестр)

ОПК-2: Способен использовать специализированные знания фундаментальных разделов математики, физики, химии и биологии для проведения исследований в области биоинженерии, биоинформатики и смежных дисциплин (модулей)
ОПК-3: Способен проводить экспериментальную работу с организмами и клетками, использовать физико-химические методы исследования макромолекул, математические методы обработки результатов биологических исследований

Компетенция	**Индикаторы компетенций**	**Результаты обучения**	**Формы и методы контроля и оценки**
ОПК-2 Способен использовать специализированные знания фундаментальных разделов математики, физики, химии и биологии для проведения исследований в области биоинженерии, биоинформатики и смежных дисциплин (модулей)	*ИДК ОПК-2.1* Демонстрирует специализированные знания в области фундаментальных разделов математики, физики, химии, биологии и перспективы междисциплинарных исследований	Знать: литературу по теме, владеть навивками, анализа информации сети «интернет» для поиска и освоения новых методов анализа данных и информационных технологий. Уметь: выбирать оптимальные методы и программы для решения задач в области анализа биологической информации по разным разделам биологических дисциплин. Владеть: методами построение анализа биологических систем с применением методов анализа функций, векторной алгебры, численных методов.	**Текущий контроль**: - письменная работа (решение самостоятельных заданий) - **Промежуточная аттестация**: экзамен
	ИДК ОПК-2.2 Умеет использовать навыки проведения исследований в области биоинженерии, биоинформатики с	Знать: базовые алгоритмы в языка программирования R программирования, процедуры и функции обработки и визуализации статистических данных и результатов моделирования, реализации численных методов.	**Текущий контроль**: - письменная работа (решение самостоятельных заданий)

	учетом специализированных фундаментальных знаний	Уметь: анализировать входные и выходные данные алгоритмов и моделей описания биологических систем, обрабатывать и визуализировать статистические данные и результаты моделирования с помощью базовых средств языка программирования R. Владеть: навыками анализа сложных данных в различных отраслях биологии и биоинформатики.	- **Промежуточная аттестация**: экзамен
	ИДК ОПК-2.3 Владеет методами химии, физики и математического моделирования для проведения исследований в области биоинженерии, биоинформатики	Знать: классификацию основных типов математических моделей и математических функций для описания и исследования биологических систем и биологических процессов. Уметь: осуществлять интерпретацию результатов математического моделирования и математических расчетов. Владеть: методами анализа комплексных биологических данных с использованием различных вычислительных и численных методов	**Текущий контроль**: - письменная работа (решение самостоятельных заданий) - **Промежуточная аттестация**: экзамен
ОПК-3 Способен проводить экспериментальную работу с организмами и клетками, использовать физико-химические методы исследования макромолекул, математические методы обработки результатов биологических исследований	*ИДК ОПК-3.1* Проводит экспериментальную работу с организмами и клетками с использованием физико-химических методов исследования макромолекул	Знать: основные математические понятия и методы, применимые для анализа биологических макромолекул. Уметь: адекватно выбрать математический метод поведения биологических систем на молекулярном уровне. Владеть: основными принципами формализации сложных биологических систем в виде математических моделей клеточных и биохимических процессов	**Текущий контроль**: - письменная работа (решение самостоятельных заданий) - **Промежуточная аттестация**: экзамен
	ИДК ОПК-3.2 Демонстрирует практические навыки математических методов обработки	Знать: цель, основные задачи и области применения математички методов в рамках направления подготовки. Уметь: формализовать исследуемую биологическую	**Текущий контроль**: - письменная работа (решение самостоятельных заданий)

	результатов экспериментальных исследований	систему и биологический процесс в виде математической модели, использовать биологические данные для проверки и тестирования математички моделей, кластеризации и систематизации биологических данных. Владеть: методами анализа и исследования разработанных математических моделей для описания различных биологических процессов и биосистем, кластеризации и систематизации биологических объектов.	- **Промежуточная аттестация**: экзамен
	ИДК ОПК-3.3 Владеет опытом применения методов для исследования макромолекул, обработки результатов биологических исследований, прогнозирования перспектив и социальных последствий своей профессиональной деятельности.	Знать: особенности и основные свойства биологических систем, описываемых с помощью математических методов, методов кластеризации и систематизации биологических объектов. Уметь: выбирать адекватные методы для анализа биологических данных систематизации и кластеризации биологических объектов. Владеть: навыками совершенствования своих профессиональных качеств в области построения математических моделей и анализа и систематизации биологических данных.	**Текущий контроль**: - письменная работа (решение самостоятельных заданий) - **Промежуточная аттестация**: экзамен

2. Оценочные материалы текущего контроля

В рамках дисциплины «Спецглавы математики» используются следующие формы текущего контроля - письменная работа по решению самостоятельных заданий (все формулировки заданий для самостоятельного решения с необходимыми сопроводительными материалами выложены на образовательном портале ИГУ в темах курса «Специальные главы математики»);

Перечень письменных работ для самостоятельного выполнения по разделам – темам дисциплины.

Задание по теме 1:

У вас есть текстовый файл `ice_cream_sales.txt`, содержащий данные о продажах мороженого в зависимости от температуры воздуха. Файл имеет следующую структуру:

```
Temperature,Sales
20,150
22,175
25,200
28,220
30,240
32,250
35,260
```

Где:

- `Temperature` - температура воздуха в градусах Цельсия.
- `Sales` - количество проданного мороженого.

Используя язык программирования R, необходимо:

1. Загрузить данные из файла `ice_cream_sales.txt` в датафрейм.
2. Построить график зависимости продаж мороженого от температуры.
3. Добавить заголовок графика "Зависимость продаж мороженого от температуры".
4. Обозначить оси графика: "Температура (°C)" и "Продажи".

Ответ:

```
# Укажите путь к вашему файлу
file_path <- "ice_cream_sales.txt"

# 1. Загрузка данных из файла
data <- read.csv(file_path)

# 2. Построение графика зависимости продаж от температуры
plot(data$Temperature, data$Sales,
    main = "Зависимость продаж мороженого от температуры", # Заголовок графика
    xlab = "Температура (°C)", # Подпись оси X
    ylab = "Продажи", # Подпись оси Y
    type = "p", # Тип графика - точки
    pch = 16,     # Тип точек
    col = "blue" # Цвет точек
)

# Можно добавить линию тренда (по желанию)
model <- lm(Sales ~ Temperature, data = data)
abline(model, col = "red")

#Добавление легенды (по желанию)
legend("topleft",
```

```
       legend = c("Данные", "Линия тренда"),
       col = c("blue", "red"),
       pch = c(16, NA),
       lty = c(NA, 1),
       cex = 0.8)
```

Задание по теме 2:

Задание:

Напишите программу на языке R, которая:

1. Использует цикл `for`, чтобы перебрать числа от 1 до 10 (включительно).
2. Для каждого числа вычисляет его квадрат и куб.
3. Выводит в консоль таблицу, содержащую число, его квадрат и его куб. Таблица должна иметь заголовки столбцов: "Число", "Квадрат", "Куб".

Ответ:

```
# Задание: Вычисление квадратов и кубов чисел от 1 до 10

# Вывод заголовков таблицы
cat("Число  Квадрат  Куб\n")

# Цикл for для перебора чисел от 1 до 10
for (i in 1:10) {
  # Вычисление квадрата и куба числа
  квадрат <- i * i
  куб <- i * i * i

  # Форматированный вывод строки таблицы
  cat(sprintf("%5d  %7d  %5d\n", i, квадрат, куб))
}
```

Задание по теме 3:

Задание:

Напишите функцию на языке R, которая принимает следующие аргументы:

1. `func`: Математическая функция, заданная как выражение R (например, `x^2` или `sin(x)`).
2. `from`: Нижняя граница интервала по оси X.
3. `to`: Верхняя граница интервала по оси X.
4. `title`: Заголовок графика (строка).
5. `x_label`: Подпись оси X (строка).
6. `y_label`: Подпись оси Y (строка).

Функция должна:

1. Сгенерировать вектор значений x в заданном интервале `from` до `to` с шагом 0.1.
2. Вычислить значения функции `func` для каждого значения x.
3. Построить график функции, используя полученные значения x и y.
4. Установить заголовок графика (`title`), подписи осей X (`x_label`) и Y (`y_label`).

Ответ:

```
# Функция для построения графика математической функции
plot_function <- function(func, from, to, title, x_label, y_label) {
  # 1. Генерация вектора значений x
  x <- seq(from, to, by = 0.1)

  # 2. Вычисление значений функции func для каждого x
  y <- eval(parse(text = func)) #Важно использовать eval и parse, потому что func передаётся как строка
```

```
  # 3. Построение графика
  plot(x, y,
       type = "l",  # Тип графика - линия
       main = title,  # Заголовок графика
       xlab = x_label, # Подпись оси X
       ylab = y_label, # Подпись оси Y
       col = "blue"  # Цвет линии
  )
}

# Пример использования функции:
plot_function("x^2", -5, 5, "График функции y = x^2", "X", "Y")

plot_function("sin(x)", 0, 2*pi, "График функции y = sin(x)", "X", "Y")

plot_function("exp(-x)", -2, 5, "График функции y = e^(-x)", "X", "Y")

# Пример использования с более сложной функцией, требующей обработки деления на ноль
# Создадим вектор y, заполненный NA (Not Available)
# Далее заполним вектор y значениями функции. При делении на ноль в y будет NA, что является допустимым.
plot_function("1/(x-1)", -2, 3, "График функции y = 1/(x-1)", "X", "Y") # Будет проблема при x = 1, т.к. деление на 0.
```

Задание по теме 3. Раздел для самостоятельного изучения.

1. Установите пакет `ggplot2` в R (при необходимости).
2. Используя пакет `ggplot2`, постройте boxplot для данных `iris$Sepal.Length`, сгруппированных по виду `iris$Species`.
3. Добавьте заголовок графика: "Boxplot длины чашелистика по видам".
4. Обозначьте оси графика: "Вид" (x-axis) и "Длина чашелистика (см)" (y-axis).
5. Измените цвет boxplots на зеленый.

Ответ:

```
# 1. Установка пакета ggplot2 (если он еще не установлен)
if(!require(ggplot2)){install.packages("ggplot2")}

# 2. Загрузка пакета ggplot2
library(ggplot2)

# 3. Построение Boxplot с использованием ggplot2
ggplot(iris, aes(x = Species, y = Sepal.Length, fill = Species)) +  # Указываем данные и переменные
  geom_boxplot(fill = "green") +  # Добавляем boxplot с зеленым цветом
  labs(title = "Boxplot длины чашелистика по видам",  # Добавляем заголовок
       x = "Вид",                  # Обозначаем ось x
       y = "Длина чашелистика (см)") + # Обозначаем ось y
  theme_minimal()          # Добавляем минималистичную тему (по желанию)
```

Задание по теме 4:

1. Сгенерируйте 100 случайных чисел, распределенных нормально, со средним значением 5 и стандартным отклонением 2. Назовите этот набор данных `data1`.

2. Сгенерируйте 100 случайных чисел, распределенных экспоненциально, с параметром rate = 0.2 (что соответствует среднему значению 5). Назовите этот набор данных `data2`.
3. Постройте гистограммы распределения для обоих наборов данных на одной координатной плоскости. Используйте разные цвета для каждой гистограммы и добавьте легенду.
4. Протестируйте каждый набор данных на соответствие нормальному распределению, используя тест Шапиро-Уилка.
5. Сравните средние значения `data1` и `data2`, используя:
   - t-тест Стьюдента (предполагается нормальность распределения, но будет проверено).
   - U-тест Манна-Уитни (непараметрический тест, не требует нормальности распределения).
6. Интерпретируйте результаты тестов.

Ответ:

```
# 1. Генерация данных
set.seed(123) # Фиксируем seed для воспроизводимости результатов

data1 <- rnorm(100, mean = 5, sd = 2)
data2 <- rexp(100, rate = 0.2)  # rate = 1/mean

# 2. Загрузка необходимых пакетов (если еще не установлены)
if(!require(ggplot2)){install.packages("ggplot2")}
if(!require(gridExtra)){install.packages("gridExtra")} #Пакетик для размещения нескольких графиков на одной плоскости

library(ggplot2)
library(gridExtra)

# 3. Построение гистограмм
hist1 <- ggplot(data.frame(x = data1), aes(x = x)) +
  geom_histogram(fill = "blue", alpha = 0.5, bins = 15) +  # alpha - прозрачность
  ggtitle("Нормальное распределение") +
  xlab("Значение") +
  ylab("Частота")

hist2 <- ggplot(data.frame(x = data2), aes(x = x)) +
  geom_histogram(fill = "red", alpha = 0.5, bins = 15) +
  ggtitle("Экспоненциальное распределение") +
  xlab("Значение") +
  ylab("Частота")

grid.arrange(hist1, hist2, ncol = 2) #Выводим гистограммы на одну плоскость

# 4. Тест Шапиро-Уилка на нормальность
shapiro.test(data1)
shapiro.test(data2)

# 5. Сравнение средних значений
#   a) T-тест Стьюдента
t.test(data1, data2)

#   b) U-тест Манна-Уитни
wilcox.test(data1, data2)
```

```
# 6. Интерпретация результатов
#  - Смотрим на p-value тестов Шапиро-Уилка. Если p-value < 0.05, то распределение
#    статистически значимо отличается от нормального.
#  - Смотрим на p-value t-теста и U-теста. Если p-value < 0.05, то средние значения
#    статистически значимо различаются.
```

Задание по теме 5:

1. Файл df_y.tsv содержит информацию выборке, распределённой по неизвестному закону распределения. Необходимо, путем анализа суммы квадратов отклонений, определить наиболее оптимальный вариант распределения, для описания выборки и соответствующей ей генеральной совокупности. При разбиении вариационного рада выборки на интервалы необходимо использовать рекомендации темы 1.
2. Для тестирования необходимо использовать законы равномерного, экспоненциального, нормального, логнормального и гамма распределений.
3. В финале необходимо построить гистограмму распределения в шкале плотности вероятностей с отображением кривых плотности вероятности для всех типов тестируемых распределений.

Ответ:

```
#установка поддержки русского языка
Sys.setlocale("LC_ALL","Russian_Russia") #установка подержки русского языка

#установка рабочей директории
setwd("H:\\Предметы\\Предметы\\Специальные_ главы_математики\\")

#загрузка таблицы данных в виде data frame
my_data<-read.csv("df_x.tsv", header=T, sep="\t", check.names = FALSE)
my_data

#////////////////////////////////////////////////////////////////////////////////////////////////////////////
#построение гистограммы и таблицы характеристик распределения по интервалам для первого ряда наблюдения (ряда nolm)

val<-my_data$value
length(val)

#формирование классов для разбиении на интервалы

k<-8 #количество интервалов, (смотрим в лекции 1)

dx<-(max(val)-min(val))/k #ширина одного интервала

my_breaks <- seq(min(val), max(val), by = dx) #задание диапазона литералов в ручную


hist(val,
    breaks = my_breaks,
    main = "Гистограмма плотности распределения для ряда x",
    xlab = "Значения",
    ylab = "Плотность вероятности",
    freq = FALSE,
    col = "lightgreen")

hist_data <- hist(val, plot = FALSE)
```

```
#получения статистики распределения по интервалам

n<-length(val)#размер выборки
m_value<-mean(val)#наблюдаемое среднее
sd_value<-sd(val)#наблюдаемое стандартное отклонение

n_interval<-length(hist_data$counts) #количество интервалов $counts - частота попадания элементов в интервал
num_iterval<-c(1:n_interval) #нумерация интервалов

in_min<-rep(0, n_interval) #вектор для хранения нижних границ интервалов
in_max<-rep(0, n_interval) #вектор для хранения верхних границ интервалов

#заполнение данных о границах интервалов
hist_data$breaks #вектор границ интервалов
#заполнение нижних границ
for(i in 1:(n_interval)){
  in_min[i]<-hist_data$breaks[i]
 }
#заполнение верхних границ
for(i in 2:(n_interval+1)){
  in_max[i-1]<-hist_data$breaks[i]
 }

#вектор значения плотности распределения для интервалов
den<-hist_data$density

#вектор наблюдаемой вероятности попадания элемента в интервал
P_observ<-rep(0, n_interval)
for(i in 1:(n_interval)){
  P_observ[i]<-den[i]*dx
 }

#вектор ожидаемой вероятности (в предположении равномерного распределения) попадания элемента в интервал
P_expected_uniform<-rep(0, n_interval)
for(i in 1:(n_interval)){
  P_expected_uniform[i]<-punif(in_max[i], min=min(val), max=max(val))-punif(in_min[i], min=min(val), max=max(val))
 }

#вектор ожидаемой вероятности (в предположении экспоненциального распределения) попадания элемента в интервал
P_expected_exp<-rep(0, n_interval)
for(i in 1:(n_interval)){
  P_expected_exp[i]<-pexp(in_max[i], rate=1/m_value)-pexp(in_min[i], rate=1/m_value)
 }


#вектор ожидаемой вероятности (в предположении нормального распределения) попадания элемента в интервал
```

```
P_expected_normal<-rep(0, n_interval)
for(i in 1:(n_interval)){
  P_expected_normal[i]<-pnorm(in_max[i], mean=m_value, sd=sd_value)-pnorm(in_min[i], mean=m_value, sd=sd_valeu)
 }

#вектор ожидаемой вероятности (в предположении логнормального распределения) попадания элемента в интервал
mu_log<-log(m_value^2/(sd_value^2+m_value^2)^0.5) #расчет логсреднего
sd_log<-(log(sd_value^2/m_value^2+1))^0.5        #расчет логстандартного поклонения

P_expected_lognormal<-rep(0, n_interval)
for(i in 1:(n_interval)){
  P_expected_lognormal[i]<-plnorm(in_max[i], meanlog=mu_log, sdlog=sd_log)-plnorm(in_min[i], meanlog=mu_log, sdlog=sd_log)
 }


#получение итоговой таблицы с характеристики распределения элементов по интервалам
x_data<-data.frame(num_iterval, in_min, in_max, P_observ, P_expected_uniform, P_expected_exp, P_expected_normal, P_expected_lognormal)
x_data

#расчет SSD

#для равномерного распределения
SSum<-rep(0, n_interval)
for(i in 1:(n_interval)){
  SSum[i]<-(x_data$P_observ[i] - x_data$P_expected_uniform[i])^2
 }

SSD_uniform<-sum(SSum) #сумма квадратов для равномерного распределения

#для равномерного распределения
SSum<-rep(0, n_interval)
for(i in 1:(n_interval)){
  SSum[i]<-(x_data$P_observ[i] - x_data$P_expected_exp[i])^2
 }

SSD_exp<-sum(SSum) #сумма квадратов для равномерного распределения

#для нормального распределения
SSum<-rep(0, n_interval)
for(i in 1:(n_interval)){
  SSum[i]<-(x_data$P_observ[i] - x_data$P_expected_normal[i])^2
 }

SSD_normal<-sum(SSum) #сумма квадратов для нормального распределения

#для логнормального распределения
SSum<-rep(0, n_interval)
for(i in 1:(n_interval)){
```

```
 SSum[i]<-(x_data$P_observ[i] - x_data$P_expected_lognormal[i])^2
}

SSD_lognormal<-sum(SSum) #сумма квадратов для логнормального распределения

#анализ результата оценок SSD

Distribution<-c("uniform", "exp", "normal", "lognormal")
SSD<-c(SSD_uniform, SSD_exp, SSD_normal, SSD_lognormal)

df_ssd<-data.frame(Distribution, SSD)
df_ssd #вывод результата на экран

#////////////////////////////////////////////////////////////////////////////////////
#построение гистограммы и функций плотности вероятности для тестируемых типов распределений

hist(val,
    breaks = my_breaks,
    main = "Гистограмма плотности распределения для ряда x",
    xlab = "Значения",
    ylab = "Плотность вероятности",
    freq = FALSE,
    col = "lightgreen")


#добавление кривой графика ожидаемой функции плотности вероятности для равномерного распределения
curve(dunif(x, min = min(x), max = max(x)),
     col = "blue",
     lwd = 4,
     add = TRUE)

#добавление кривой графика ожидаемой функции плотности вероятности для экспоненцтального распределения
curve(dexp(x, rate = 1/m_value),
     col = "goldenrod1",
     lwd = 4,
     add = TRUE)

#добавление кривой графика ожидаемой функции плотности вероятности для нормального распределения
curve(dnorm(x, mean = m_value, sd = sd_value),
     col = "red",
     lwd = 4,
     add = TRUE)

#добавление кривой графика ожидаемой функции плотности вероятности для логнормального распределения
curve(dlnorm(x, meanlog = mu_log, sdlog = sd_log),
     col = "black",
     lwd = 4,
```

add = TRUE)

Задание по теме 6:
Задание по исследованию внутригрупповых и межгрупповых дистанций.
Файл human_data.tsv содержит таблицу, строки которой представляют собой вектора многомерных статических данных, характеризующих ряды показателей людей из различных населённых пунктов.
Для анализа необходимо выбрать строки. Характеризующие женящих из городов Красноярск и Владивосток – две группы в многомерном массиве данных.
Для выбранных групп необходимо провести нормировку данных Min-Max – ранжированием от 0 до 1, и рассчитать матрицу несходства между анализируемым объектами по метрике Брея_кертиса.
Необходимо на одной странице в пределах разбиения на эквивалентные интервалы визуализировать в виде гистограммы плотности распределения (с наложение кривой эмпирической плотности вероятности) дистанций в каждой группе и между группами.
Необходимо, путем анализа суммы квадратов отклонений, определить наиболее оптимальный вариант распределения, закона распределения дистанций в каждой группе и между группами. Анализ необходимо провести с использованием расчетов эмпирической (кумулятивной) и ожидаемой функций распределения вероятностей. Для тестирования необходимо использовать законы равномерного, экспоненциального, нормального, логнормального и гамма распределений.
В финале необходимо построить четыре графика: 1) три гистограммы распределения в шкале плотности вероятностей с отображением кривых плотности вероятности для внутригрупповых и межгрупповых дистанций; 2) график эмпирической функции распределения с наложением графиком ожидаемых функций распределения всех типов тестируемых распределений для дистанций группы 1; 3) график эмпирической функции распределения с наложением графиком ожидаемых функций распределения всех типов тестируемых распределений для дистанций группы 2; 3) график эмпирической функции распределения с наложением графиком ожидаемых функций распределения всех типов тестируемых распределений для межгрупповых дистанций.
Опираюсь на центральную предельную теорему, необходимо оценить 95% доверительные интервалы для средних внутригрупповых дистанций (в каждой группе) и средних межгрупповых дистанций.

Ответ:
```
install.packages("vegan")

#подключение библиотеки vegan
library(vegan)

#установка рабочей директории
setwd("E:\\Предметы\\Предметы_осень_2025\\Специальные_
главы_математики\\Занятие_8")


#установка поддержки русского языка
Sys.setlocale("LC_ALL","Russian_Russia") #установка поддержки русского языка

#установка рабочей директории
setwd("H:\\Предметы\\Предметы_осень\\Специальные_ главы_математики\\")

#загрузка таблицы данных в виде data frame
my_data<-read.csv("human_data.tsv", header=T, sep="\t", check.names = FALSE)
my_data
```

```
#выбор данных по мужчинам и женщинам из города Москва
my_data<-my_data[my_data$город_проживания=="Москва", ]

rownames(my_data)<-paste0(my_data$номер, "_", my_data$пол)

Expl<-my_data[, c(2, 3)]

my_data<-my_data[, -c(1, 2, 3)]

Expl
my_data

#нормировка данных ранжированием

my_data<-decostand(my_data,  method = "range")

#рассечет дистанций
d<-vegdist(my_data, method = "euclidean")

#анализ распределения дистанций

d<-as.matrix(d)

n_male<-which(Expl$пол %in% "м") #выделение номеров строк, характеризующих мужчин

n_females<-which(Expl$пол %in% "ж") #выделение номеров строк, характеризующих женщин

#выделение дистанций, характеризующих различие внутри группы - мужчины
d_male<-d[n_male, n_male]
d_male<-as.dist(d_male)

#выделение дистанций, характеризующих различие внутри группы - жен шины
d_females<-d[n_females, n_females]
d_females<-as.dist(d_females)

#выделение дистанций межгрупповых дистанций
d_between<-d[n_male, n_females]
d_between<-as.dist(d_between)

#сравнительный анализ распределений дистанций

#формирование классов для разбиении на интервалы
(nrow(my_data)^2)/2-nrow(my_data) #размер выборки

k<-30 #количество интервалов, (смотрим в лекции 1)

dx<-(max(as.dist(d))-min(as.dist(d)))/k #ширина одного интервала

my_breaks <- seq(min(as.dist(d)), max(as.dist(d)), by = dx) #задание диапазона литералов в ручную
```

```
#построение гистограмм распределения для дистанций внутри мужнин, женщин и межгрупповых дистанций
par(mfrow = c(3, 1))

hist(d_male,
    breaks = my_breaks,
    main = "гистограмма распределения дистанции для мужчин",
    xlab = "Значения",
    ylab = "Плотность вероятности",
    freq = FALSE,
    col = "lightgreen")
lines(density(d_male), col = "red", lwd = 4)

hist(d_females,
    breaks = my_breaks,
    main = "гистограмма распределения дистанции для женщин",
    xlab = "Значения",
    ylab = "Плотность вероятности",
    freq = FALSE,
    col = "lightgreen")
lines(density(d_females), col = "red", lwd = 4)

hist(d_between,
    breaks = my_breaks,
    main = "гистограмма распределения межгрупповых дистанций",
    xlab = "Значения",
    ylab = "Плотность вероятности",
    freq = FALSE,
    col = "lightgreen")
lines(density(d_between), col = "red", lwd = 4)

#////////////////////////////////////////////////////////////////////////////////
#////////////////////////////////////////////////////////////////////////////////
#выбор наиболее оптимального типа распределения мужчин с помощью функции распределения

val<-as.vector(d_male)
length(val)

#расчет среднего и стандартного отклонения

m_value<-mean(val)#наблюдаемое среднее
sd_value<-sd(val)#наблюдаемое стандартное отклонение

m_male<-mean(val)#наблюдаемое среднее
sd_male<-sd(val)#наблюдаемое стандартное отклонение

#формирования значений эмпирической функции кумулятивного распределения

#получение таблицы встречаемости различных значений из ряда наблюдений
x_value<-table(val)
```

```
#создание data frame - таблицы данных со списком разных значений x - x_value и их встречаемости - count в ряде наблюдений
df_com<-data.frame(x_value=as.numeric(names(x_value)), count=as.vector(x_value))

#упорядочивание таблицы данных по увеличению значений x_value
df_com<-df_com[order(df_com$x_value, decreasing=F),]

#добавление в таблицу данных кумулятивной суммы по ряду знамений count
df_com<-cbind(df_com, cumulative_summ<-cumsum(df_com$count))

#добавление в таблицу данных расчетных значений наблюдаемой - кумулятивной функции распределения для элементов ряда наблюдения x
df_com<-cbind(df_com, F_observ=df_com$cumulative_summ/sum(df_com$count))

#вектор ожидаемых значений функции распределения(в предположении равномерного распределения)
F_expected_uniform<-rep(0, nrow(df_com))
for(i in 1:nrow(df_com)){
  F_expected_uniform[i]<-punif(df_com$x_value[i], min=min(df_com$x_value), max=max(df_com$x_value))
 }

#вектор ожидаемой значений функции распределения (в предположении экспоненциального распределения) попадания элемента в интервал
F_expected_exp<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  F_expected_exp[i]<-pexp(df_com$x_value[i], rate=1/m_value)
 }

#вектор ожидаемых значений функции распределения (в предположении нормального распределения) попадания элемента в интервал
F_expected_normal<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  F_expected_normal[i]<-pnorm(df_com$x_value[i], mean=m_value, sd=sd_value)
 }

#вектор ожидаемых значений функции распределения (в предположении логнормального распределения) попадания элемента в интервал
mu_log<-log(m_value^2/(sd_value^2+m_value^2)^0.5) #расчет логсреднего
sd_log<-(log(sd_value^2/m_value^2+1))^0.5        #расчет логстандартного поклонения

F_expected_lognormal<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  F_expected_lognormal[i]<-plnorm(df_com$x_value[i], meanlog=mu_log, sdlog=sd_log)
 }

#получение итоговой таблицы с характеристики распределения и значениями функций распределения
df_com<-cbind(df_com, F_expected_uniform, F_expected_exp, F_expected_normal, F_expected_lognormal)
df_com
```

```
#расчет SSD

#для равномерного распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_uniform[i])^2
 }

SSD_uniform<-sum(SSum) #сумма квадратов для равномерного распределения

#для экспоненциального распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_exp[i])^2
 }

SSD_exp<-sum(SSum) #сумма квадратов для равномерного распределения

#для нормального распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_normal[i])^2
 }

SSD_normal<-sum(SSum) #сумма квадратов для нормального распределения

#для логнормального распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_lognormal[i])^2
 }

SSD_lognormal<-sum(SSum) #сумма квадратов для логнормального распределения

#анализ результата оценок SSD

Distribution<-c("uniform", "exp", "normal", "lognormal")
SSD<-c(SSD_uniform, SSD_exp, SSD_normal, SSD_lognormal)

df_ssd<-data.frame(Distribution, SSD)
df_ssd #вывод результата на экран

#построение ломаной линии наблюдаемой кумулятивной функции вероятности и графиков ожидаемых функций вероятностей
par(mfrow = c(1, 1))

plot(df_com$x_value, df_com$F_observ,
    main = "Кумулятивной функции вероятности для ряда x",
    xlab = "Значения",
    ylab = "Плотность вероятности",
          pch=19,
          cex=1.2,
```

```
        col="red"
   )
lines(df_com$x_value, df_com$F_observ, col = "red", lwd = 2)

#добавление кривой графика ожидаемой функции вероятности для равномерного распределения
curve(punif(x, min = min(df_com$x_value), max = max(df_com$x_value)),
     col = "blue",
     lwd = 3,
     add = TRUE)

#добавление кривой графика ожидаемой функции вероятности для экспоненциального распределения
curve(pexp(x, rate=1/m_value),
     col = "goldenrod1",
     lwd = 3,
     add = TRUE)

#добавление кривой графика ожидаемой функции вероятности для нормального распределения
curve(pnorm(x, mean = m_value, sd = sd_value),
     col = "green",
     lwd = 3,
     add = TRUE)

#добавление кривой графика ожидаемой функции вероятности для логнормального распределения
curve(plnorm(x, meanlog = mu_log, sdlog = sd_log),
     col = "black",
     lwd = 3,
     add = TRUE)

#////////////////////////////////////////////////////////////////////////////////////////////////


#////////////////////////////////////////////////////////////////////////////////////
#////////////////////////////////////////////////////////////////////////////////////
#выбор наиболее оптимального типа распределения для дистанций внутри группы мужчин с помощью функции распределения

val<-as.vector(d_females)
length(val)

#расчет среднего и стандартного отклонения

m_value<-mean(val)#наблюдаемое среднее
sd_value<-sd(val)#наблюдаемое стандартное отклонение

m_female<-mean(val)#наблюдаемое среднее
sd_female<-sd(val)#наблюдаемое стандартное отклонение
```

```
#формирования значений эмпирической функции кумулятивного распределения

#получение таблицы встречаемости различных значений из ряда наблюдений
x_value<-table(val)

#создание data frame - таблицы данных со списком разных значений x - x_value и их встречаемости - count в ряде наблюдений
df_com<-data.frame(x_value=as.numeric(names(x_value)), count=as.vector(x_value))

#упорядочивание таблицы данных по увеличению значений x_value
df_com<-df_com[order(df_com$x_value, decreasing=F),]

#добавление в таблицу данных кумулятивной суммы по ряду знамений count
df_com<-cbind(df_com, cumulative_summ<-cumsum(df_com$count))

#добавление в таблицу данных расчетных значений наблюдаемой - кумулятивной функции распределения для элементов ряда наблюдения x
df_com<-cbind(df_com, F_observ=df_com$cumulative_summ/sum(df_com$count))

#вектор ожидаемых значений функции распределения(в предположении равномерного распределения)
F_expected_uniform<-rep(0, nrow(df_com))
for(i in 1:nrow(df_com)){
  F_expected_uniform[i]<-punif(df_com$x_value[i], min=min(df_com$x_value), max=max(df_com$x_value))
 }

#вектор ожидаемой значений функции распределения (в предположении экспоненциального распределения) попадания элемента в интервал
F_expected_exp<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  F_expected_exp[i]<-pexp(df_com$x_value[i], rate=1/m_value)
 }

#вектор ожидаемых значений функции распределения (в предположении нормального распределения) попадания элемента в интервал
F_expected_normal<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  F_expected_normal[i]<-pnorm(df_com$x_value[i], mean=m_value, sd=sd_value)
 }

#вектор ожидаемых значений функции распределения (в предположении логнормального распределения) попадания элемента в интервал
mu_log<-log(m_value^2/(sd_value^2+m_value^2)^0.5) #расчет логсреднего
sd_log<-(log(sd_value^2/m_value^2+1))^0.5        #расчет логстандартного поклонения

F_expected_lognormal<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  F_expected_lognormal[i]<-plnorm(df_com$x_value[i], meanlog=mu_log, sdlog=sd_log)
 }
```

```
#получение итоговой таблицы с характеристики распределения и значениями функций распределения
df_com<-cbind(df_com, F_expected_uniform, F_expected_exp, F_expected_normal, F_expected_lognormal)
df_com

#расчет SSD

#для равномерного распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_uniform[i])^2
 }

SSD_uniform<-sum(SSum) #сумма квадратов для равномерного распределения

#для экспоненциального распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_exp[i])^2
 }

SSD_exp<-sum(SSum) #сумма квадратов для равномерного распределения

#для нормального распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_normal[i])^2
 }

SSD_normal<-sum(SSum) #сумма квадратов для нормального распределения

#для логнормального распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_lognormal[i])^2
 }

SSD_lognormal<-sum(SSum) #сумма квадратов для логнормального распределения

#анализ результата оценок SSD

Distribution<-c("uniform", "exp", "normal", "lognormal")
SSD<-c(SSD_uniform, SSD_exp, SSD_normal, SSD_lognormal)

df_ssd<-data.frame(Distribution, SSD)
df_ssd #вывод результата на экран

#построение ломаной линии наблюдаемой кумулятивной функции вероятности и графиков ожидаемых функций вероятностей
par(mfrow = c(1, 1))
```

```
plot(df_com$x_value, df_com$F_observ,
    main = "Кумулятивной функции вероятности для ряда x",
    xlab = "Значения",
    ylab = "Плотность вероятности",
          pch=19,
          cex=1.2,
          col="red"
   )
lines(df_com$x_value, df_com$F_observ, col = "red", lwd = 2)

#добавление кривой графика ожидаемой функции вероятности для равномерного распределения
curve(punif(x, min = min(df_com$x_value), max = max(df_com$x_value)),
     col = "blue",
     lwd = 3,
     add = TRUE)

#добавление кривой графика ожидаемой функции вероятности для экспоненциального распределения
curve(pexp(x, rate=1/m_value),
     col = "goldenrod1",
     lwd = 3,
     add = TRUE)

#добавление кривой графика ожидаемой функции вероятности для нормального распределения
curve(pnorm(x, mean = m_value, sd = sd_value),
     col = "green",
     lwd = 3,
     add = TRUE)

#добавление кривой графика ожидаемой функции вероятности для логнормального распределения
curve(plnorm(x, meanlog = mu_log, sdlog = sd_log),
     col = "black",
     lwd = 3,
     add = TRUE)

#////////////////////////////////////////////////////////////////////////////////////////////


#////////////////////////////////////////////////////////////////////////////////////////////
#////////////////////////////////////////////////////////////////////////////////////////////
#выбор наиболее оптимального типа распределения для межгрупповых дистанций с помощью функции распределения

val<-as.vector(d_between)
length(val)

#расчет среднего и стандартного отклонения

m_value<-mean(val)#наблюдаемое среднее
```

```
sd_value<-sd(val)#наблюдаемое стандартное отклонение

m_between<-mean(val)#наблюдаемое среднее
sd_between<-sd(val)#наблюдаемое стандартное отклонение

#формирования значений эмпирической функции кумулятивного распределения

#получение таблицы встречаемости различных значений из ряда наблюдений
x_value<-table(val)

#создание data frame - таблицы данных со списком разных значений x - x_value и их встречаемости - count в ряде наблюдений
df_com<-data.frame(x_value=as.numeric(names(x_value)), count=as.vector(x_value))

#упорядочивание таблицы данных по увеличению значений x_value
df_com<-df_com[order(df_com$x_value, decreasing=F),]

#добавление в таблицу данных кумулятивной суммы по ряду знамений count
df_com<-cbind(df_com, cumulative_summ<-cumsum(df_com$count))

#добавление в таблицу данных расчетных значений наблюдаемой - кумулятивной функции распределения для элементов ряда наблюдения x
df_com<-cbind(df_com, F_observ=df_com$cumulative_summ/sum(df_com$count))

#вектор ожидаемых значений функции распределения(в предположении равномерного распределения)
F_expected_uniform<-rep(0, nrow(df_com))
for(i in 1:nrow(df_com)){
  F_expected_uniform[i]<-punif(df_com$x_value[i], min=min(df_com$x_value), max=max(df_com$x_value))
 }

#вектор ожидаемой значений функции распределения (в предположении экспоненциального распределения) попадания элемента в интервал
F_expected_exp<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  F_expected_exp[i]<-pexp(df_com$x_value[i], rate=1/m_value)
 }

#вектор ожидаемых значений функции распределения (в предположении нормального распределения) попадания элемента в интервал
F_expected_normal<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  F_expected_normal[i]<-pnorm(df_com$x_value[i], mean=m_value, sd=sd_value)
 }

#вектор ожидаемых значений функции распределения (в предположении логнормального распределения) попадания элемента в интервал
mu_log<-log(m_value^2/(sd_value^2+m_value^2)^0.5) #расчет логсреднего
sd_log<-(log(sd_value^2/m_value^2+1))^0.5        #расчет логстандартного поклонения

F_expected_lognormal<-rep(0, nrow(df_com))
```

```
for(i in 1:(nrow(df_com))){
  F_expected_lognormal[i]<-plnorm(df_com$x_value[i], meanlog=mu_log, sdlog=sd_log)
 }

#получение итоговой таблицы с характеристики распределения и значениями функций распределения
df_com<-cbind(df_com, F_expected_uniform, F_expected_exp, F_expected_normal, F_expected_lognormal)
df_com

#расчет SSD

#для равномерного распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_uniform[i])^2
 }

SSD_uniform<-sum(SSum) #сумма квадратов для равномерного распределения

#для экспоненциального распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_exp[i])^2
 }

SSD_exp<-sum(SSum) #сумма квадратов для равномерного распределения

#для нормального распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_normal[i])^2
 }

SSD_normal<-sum(SSum) #сумма квадратов для нормального распределения

#для логнормального распределения
SSum<-rep(0, nrow(df_com))
for(i in 1:(nrow(df_com))){
  SSum[i]<-(df_com$F_observ[i] - df_com$F_expected_lognormal[i])^2
 }

SSD_lognormal<-sum(SSum) #сумма квадратов для логнормального распределения

#анализ результата оценок SSD

Distribution<-c("uniform", "exp", "normal", "lognormal")
SSD<-c(SSD_uniform, SSD_exp, SSD_normal, SSD_lognormal)

df_ssd<-data.frame(Distribution, SSD)
df_ssd #вывод результата на экран
```

```
#построение ломаной линии наблюдаемой кумулятивной функции вероятности и графиков ожидаемых функций вероятностей
par(mfrow = c(1, 1))

plot(df_com$x_value, df_com$F_observ,
    main = "Кумулятивной функции вероятности для ряда x",
    xlab = "Значения",
    ylab = "Плотность вероятности",
          pch=19,
          cex=1.2,
          col="red"
   )
lines(df_com$x_value, df_com$F_observ, col = "red", lwd = 2)

#добавление кривой графика ожидаемой функции вероятности для равномерного распределения
curve(punif(x, min = min(df_com$x_value), max = max(df_com$x_value)),
     col = "blue",
     lwd = 3,
     add = TRUE)

#добавление кривой графика ожидаемой функции вероятности для экспоненциального распределения
curve(pexp(x, rate=1/m_value),
     col = "goldenrod1",
     lwd = 3,
     add = TRUE)

#добавление кривой графика ожидаемой функции вероятности для нормального распределения
curve(pnorm(x, mean = m_value, sd = sd_value),
     col = "green",
     lwd = 3,
     add = TRUE)

#добавление кривой графика ожидаемой функции вероятности для логнормального распределения
curve(plnorm(x, meanlog = mu_log, sdlog = sd_log),
     col = "black",
     lwd = 3,
     add = TRUE)
```

Задание по теме 7:

Файл human_data.tsv содержит таблицу, строки которой представляют собой вектора многомерных статических данных, характеризующих ряды показателей людей из различных населённых пунктов.

Для анализа необходимо выбрать строки. Характеризующие женящих из городов Красноярск и Владивосток – две группы в многомерном массиве данных.

Для выбранных групп необходимо провести нормировку данных Min-Max – ранжированием путем от 0 до 1 и логарифмированием (в шкале десятичного логарифма log10(x+1)), поучив, таким образом два массива данных с разными способами нормирования.

Для двух нормированных массивов данных необходимо провести процедуру метрического и не метрического многомерного шкалирования на основе Евклидовых дистанций. Для каждого

способа нормирования и методов многомерного шкалирования необходимо построить диаграмму рассеяния объектов в пространстве двух координат (с визуализации принадлежности к городу проживания) многомерного шкалирования, построить гистограммы распределений (с функциями плотности распределения) наблюдаемых и ожидаемых дистанций, построить графики зависимостей между наблюдаемыми и ожидаемыми дистанциями. Для каждого способа нормирования необходимо рассчитать значения Stress функции в относительной и абсолютной шкале максимальное - минимальное значение относительных различий между наблюдаемыми и ожидаемыми дистанциями, 95% интервал значений для относительных различий.

Ответ:

```
#подключение библиотеки vegan
library(vegan)

#установка рабочей директории
setwd("E:\\Предметы\\Предметы_осень\\Специальные_главы_математики\\")


#установка поддержки русского языка
Sys.setlocale("LC_ALL","Russian_Russia") #установка поддержки русского языка

#загрузка таблицы данных в виде data frame
my_data<-read.csv("human_data.tsv", header=T, sep="\t", check.names = FALSE)
my_data

#выбор данных по мужчинам и женщинам из города Москва
my_data<-my_data[my_data$город_проживания=="Москва", ]

rownames(my_data)<-paste0(my_data$номер, "_", my_data$пол)

Expl<-my_data[, c(2, 3)]

my_data<-my_data[, -c(1, 2, 3)]

Expl
my_data

#нормировка данных ранжированием

my_data<-decostand(my_data,  method = "range")

#рассечет дистанций
d<-vegdist(my_data, method = "euclidean")

#////////////////////////////////////
#////////////////////////////////////
#метрическое многомерное шкалирование

fit <- cmdscale(d, eig=TRUE, k=2) #процедура метрического многомерного шкалирования
x <- fit$points[,1] #выделение x координаты образцов
y <- fit$points[,2] #выделение y координаты образцов
```

```
#визуализация результатов в виде диаграммы рассеяния с отображением точек, распределенных по фактору пола
plot(x, y, xlab="MDS1", ylab="MDS2", main="Metric MDS", typ="n")
points(x[Expl$пол=="м"], y[Expl$пол=="м"], pch=0, lwd=3.5, cex = 1.7, col="blue")
points(x[Expl$пол=="ж"], y[Expl$пол=="ж"], pch=0, lwd=3.5, cex = 1.7, col="red")
text(x, y, labels = row.names(my_data), pos=1, cex=.7) #добавление на диаграмму рассеяния текстовых идентификаторов объектов

#оценка качества работы MDS

d_observ<-d #матрица наблюдаемых дистанций между объектами

#таблица данных координат объектов в пространстве MDS
xy<-data.frame(x, y)
rownames(xy)<-rownames(my_data)

d_expected<-vegdist(xy, method = "euclidean") #матрица ожидаемых дистанций между объектами

#сравнительный анализ распределения дистанций

#формирование классов для разбиении на интервалы
(nrow(my_data)^2)/2-nrow(my_data) #размер выборки

k<-30 #количество интервалов, (смотрим в лекции 1)

d_max<-max(c(max(as.dist(d_observ)), max(as.dist(d_expected)))) #максимальное значение из дух матриц дистанций

d_min<-min(c(min(as.dist(d_observ)), min(as.dist(d_expected)))) #минимальное значение из дух матриц дистанций

dx<-(d_max-d_min)/k #ширина одного интервала

my_breaks <- seq(d_min, d_max, by = dx) #задание диапазона литералов в ручную

#построение гистограмм распределений наблюдаемых и ожидаемых дистанций
par(mfrow = c(2, 1))

hist(d_observ,
    breaks = my_breaks,
    main = "гистограмма распределения наблюдаемых дистанции",
    xlab = "Значения",
    ylab = "Плотность вероятности",
    freq = FALSE,
    col = "lightgreen")
lines(density(d_observ), col = "red", lwd = 4)

hist(d_expected,
    breaks = my_breaks,
    main = "гистограмма распределения ожидаемых дистанции",
    xlab = "Значения",
```

```
    ylab = "Плотность вероятности",
    freq = FALSE,
    col = "lightgreen")
lines(density(d_expected), col = "red", lwd = 4)


#графическая оценка зависимости между наблюдаемыми и ожидаемыми дистанциями
par(mfrow = c(1, 1))
plot(d_observ, d_expected, xlab="d_observ", ylab="d_expected", main="зависимость между наблюдаемыми ожидаемыми дистанциями", typ="n")
points(d_observ, d_expected, pch=20, lwd=0.25, cex = 1.7, col="green")

#оценка абсолютного значения Stress функции

dsq<-(d_observ-d_expected)^2 #матрица разностей квадратов

Stress<-sum(dsq)^0.5 #значение функции Stress
Stress

#оценка относительного значения Stress функции

d_v_observ<-as.vector(d_observ)  #вектор наблюдаемых дистанций
d_v_expected<-as.vector(d_expected) #вектор ожидаемых дистанций

d_relativ<-rep(0, length(d_v_observ)) #вектор относительных различий между наблюдаемыми и ожидаемыми дистанциями

#вычисление относительных различий
for(i in 1:length(d_v_observ))
 {
  dmax<-max(c(d_v_observ[i], d_v_expected[i]))
  dmin<-min(c(d_v_observ[i], d_v_expected[i]))
  d_relativ[i]<-(dmax-dmin)/dmax
 }

Stress_relativ<-mean(d_relativ) #среднее значение относительных различий - относительное значение Stress

Stress_relativ

max(d_relativ) #максимальное значение относительных различий
min(d_relativ) #минимальное значение относительных различий
quantile(d_relativ, probs = 0.025) #нижняя граница 95% интервала различий
quantile(d_relativ, probs = 0.975) #верхняя граница 95% интервала различий


#////////////////////////////////////
#////////////////////////////////////
#не метрическое многомерное шкалирование

fit <- metaMDS(my_data, distance = "euclidean") #процедура не метрического многомерного шкалирования
```

```
x <- fit$points[,1] #выделение x координаты образцов
y <- fit$points[,2] #выделение y координаты образцов

#визуализация результатов в виде диаграммы рассеяния с отображением точек, распределенных по фактору пола
plot(x, y, xlab="NMDS1", ylab="NMDS2", main="Nonmetric Metric MDS", typ="n")
points(x[Expl$пол=="м"], y[Expl$пол=="м"], pch=0, lwd=3.5, cex = 1.7, col="blue")
points(x[Expl$пол=="ж"], y[Expl$пол=="ж"], pch=0, lwd=3.5, cex = 1.7, col="red")
text(x, y, labels = row.names(my_data), pos=1, cex=.7) #добавление на диаграмму рассеяния текстовых идентификаторов объектов

#оценка качества работы MDS

d_observ<-d #матрица наблюдаемых дистанций между объектами

#таблица данных координат объектов в пространстве MDS
xy<-data.frame(x, y)
rownames(xy)<-rownames(my_data)

d_expected<-vegdist(xy, method = "euclidean") #матрица ожидаемых дистанций между объектами

#сравнительный анализ распределения дистанций

#формирование классов для разбиении на интервалы
(nrow(my_data)^2)/2-nrow(my_data) #размер выборки

k<-30 #количество интервалов, (смотрим в лекции 1)

d_max<-max(c(max(as.dist(d_observ)), max(as.dist(d_expected)))) #максимальное значение из дух матриц дистанций

d_min<-min(c(min(as.dist(d_observ)), min(as.dist(d_expected)))) #минимальное значение из дух матриц дистанций

dx<-(d_max-d_min)/k #ширина одного интервала

my_breaks <- seq(d_min, d_max, by = dx) #задание диапазона литералов в ручную

#построение гистограмм распределений наблюдаемых и ожидаемых дистанций
par(mfrow = c(2, 1))

hist(d_observ,
    breaks = my_breaks,
    main = "гистограмма распределения наблюдаемых дистанции",
    xlab = "Значения",
    ylab = "Плотность вероятности",
    freq = FALSE,
    col = "lightgreen")
lines(density(d_observ), col = "red", lwd = 4)

hist(d_expected,
```

```
    breaks = my_breaks,
    main = "гистограмма распределения ожидаемых дистанции",
    xlab = "Значения",
    ylab = "Плотность вероятности",
    freq = FALSE,
    col = "lightgreen")
lines(density(d_expected), col = "red", lwd = 4)


#графическая оценка зависимости между наблюдаемыми и ожидаемыми дистанциями
par(mfrow = c(1, 1))
plot(d_observ, d_expected, xlab="d_observ", ylab="d_expected", main="зависимость между наблюдаемыми ожидаемыми дистанциями", typ="n")
points(d_observ, d_expected, pch=20, lwd=0.25, cex = 1.7, col="green")

#оценка абсолютного значения Stress функции

fit #вывод на экран значения Stress функции

#оценка относительного значения Stress функции

d_v_observ<-as.vector(d_observ)  #вектор наблюдаемых дистанций
d_v_expected<-as.vector(d_expected) #вектор ожидаемых дистанций

d_relativ<-rep(0, length(d_v_observ)) #вектор относительных различий между наблюдаемыми и ожидаемыми дистанциями

#вычисление относительных различий
for(i in 1:length(d_v_observ))
 {
  dmax<-max(c(d_v_observ[i], d_v_expected[i]))
  dmin<-min(c(d_v_observ[i], d_v_expected[i]))
  d_relativ[i]<-(dmax-dmin)/dmax
 }

Stress_relativ<-mean(d_relativ) #среднее значение относительных различий - относительное значение Stress

Stress_relativ

max(d_relativ) #максимальное значение относительных различий
min(d_relativ) #минимальное значение относительных различий
quantile(d_relativ, probs = 0.025) #нижняя граница 95% интервала различий
quantile(d_relativ, probs = 0.975) #верхняя граница 95% интервала различий
```

Задание по теме 8:

Файл gen_data_1.tsv содержит таблицу встречаемости списка их 10 генов в 200 научных публикациях.

Используя средства языка программирования R и пакет «igraph» необходимо оставить матрицу смежности и визуализировать на ее основе ненаправленный граф межгенных связей. При визуализации необходимо использовать два способа: 1) группировка вершин по количеству связей; 2) кольцевая схема построения графа. При визуализации необходимо

раскрасить: вершины для генов с 1 по 3 в красны цвет; вершины для генов с 4 по 7 в зеленый цвет; вершины для генов с 8 по 10 в синий цвет.
Используя средства языка программирования R и пакет «igraph» необходимо оставить таблицу смежных вершин и визуализировать на ее основе ненаправленный граф межгенных связей. При визуализации необходимо использовать два способа: 1) группировка вершин по количеству связей; 2) кольцевая схема построения графа. При визуализации необходимо раскрасить: вершины для генов с 1 по 3 в красны цвет; вершины для генов с 4 по 7 в зеленый цвет; вершины для генов с 8 по 10 в синий цвет. Для всех вариантов визуализаций использовать динамическую толщину ребер графа (текущая толщина линии, деланная на 2).

Ответ:

```
#установка  пакета igraph
install.packages("igraph")


#подключение библиотеки
library(igraph)

#установка рабочей директории
setwd("H:\\Предметы\\Предметы_осень_2025\\Специальные_
главы_математики\\Занятие_10")


#установка поддержки русского языка
Sys.setlocale("LC_ALL","Russian_Russia") #установка поддержки русского языка

#загрузка таблицы данных в виде data frame
my_data<-read.csv("gen_data_1.tsv", header=T, sep="\t", check.names = FALSE)
rownames(my_data)<-my_data[,1]
my_data<-my_data[,-1]

#способ постарения графа из матрицы смежности
#создание матрицы инцидентов - матрицы смежности
DI<-matrix(nrow=nrow(my_data), ncol=nrow(my_data)) #Incidence matrix

rownames(DI)<-rownames(my_data)
colnames(DI)<-rownames(my_data)

for(i in 1:nrow(my_data))
 for(j in 1:nrow(my_data))
  {
   dfv<-my_data[c(i,j),]
   s<-colSums(dfv)
   DI[i,j]<-length(s[s==2])
   if(i==j) DI[i,j]<-0
  }

for(i in 1:nrow(my_data))
 for(j in 1:nrow(my_data))
  {
   if(DI[i,j]>=3) DI[i,j]<-1 else DI[i,j]<-0
  }
```

```
#построение графа
network <- graph_from_adjacency_matrix(DI , mode='undirected', diag=F)
#mode='undirected' - будет создан неориентированный граф, и max(A(i,j), A(j,i)) даст количество ребер.
#diag=F - не учитывать диагональный элементы матрицы для построения петель

#визуализация графа
plot(network)
#группировка вершин по количеству связей

plot(network, layout=layout_in_circle)
#layout=layout_in_circle - кольцевая схема построения графа

#визуализация с параметрами
plot(network,
    edge.width = 5, #толщина ребра
    ertex.size = 20, #размер указателя вершины
    vertex.color = "lightblue", #цвет указателя вершины
    vertex.label.color = "black", #цвет контура указателя вершины
    edge.color = "gray50", #цвет ребра
    )

#динамическое определение параметров визуализации
nrow(DI)

my_col<-c(rep("red", 5), rep("green", 5))

plot(network,
    edge.width = 2, #толщина ребра
    ertex.size = 20, #размер указателя вершины
    vertex.color = my_col, #цвет указателя вершины
    vertex.label.color = "black", #цвет контура указателя вершины
    edge.color = "gray50", #цвет ребра
   )


#////////////////////////////////////////////////
#способ постарения графа из таблицы смежных вершин
#создание таблицы смежных вершин
links_table<-data.frame(source=character(), target=character(), width=numeric()) #таблица смежных вершин

for(i in 1:(nrow(my_data)-1))
 for(j in (i+1):(nrow(my_data)))
  {
   dfv<-my_data[c(i,j),]
   s<-colSums(dfv)
   if(length(s[s==2])>=3)
    {
        links_tv<-data.frame(source=rownames(dfv)[1], target=rownames(dfv)[2], width=length(s[s==2]))
        links_table<-rbind(links_table, links_tv)
```

```
        }
  }

#построение графа
network <- graph_from_data_frame(links_table, directed=F)
#directed=F - создавать иле нет направленный граф.


#визуализация графа
plot(network)
#группировка вершин по количеству связей

plot(network, layout=layout_in_circle)
#layout=layout_in_circle - кольцевая схема построения графа

#визуализация с параметрами
plot(network,
     edge.width = 2, #толщина ребра
     ertex.size = 20, #размер указателя вершины
     vertex.color = "lightblue", #цвет указателя вершины
     vertex.label.color = "black", #цвет контура указателя вершины
     edge.color = "gray50", #цвет ребра
    )

#динамическое определение параметров визуализации

my_col<-c(rep("red", 5), rep("green", 5))

plot(network,
     edge.width = links_table$width/2, #толщина ребра
     ertex.size = 20, #размер указателя вершины
     vertex.color = my_col, #цвет указателя вершины
     vertex.label.color = "black", #цвет контура указателя вершины
     edge.color = "gray50", #цвет ребра
    )
```

Критерий оценивания самодеятельной работы – результаты по каждой работе оформляться по указанным требованиям (смотрите в описании задания) и загружаться на образовательный портал ИГУ (https://educa.isu.ru/). Преподаватель оценивает задания, если все решено верно, студен получает зачет по задания, если имеются недочеты или ошибки, задание отправляться на доработку с указанием допущенных ошибок. Отчёт по переработанному заданию загружается на образовательный портал для повторного оценивания.

3.Оценочные средства для промежуточной аттестации

Промежуточная аттестация проходит в форме экзамена (3 семестр), к которому допускаются студенты, выполнившие в полном объеме аудиторную нагрузку, самостоятельную работу. Студенты, имеющие задолженность, должны выполнить все обязательные виды деятельности.

Фонд оценочных средств для промежуточной аттестации включает:

- тестовые задания для экзамена.

Назначение оценочных средств: выявить сформированность компетенций ОПК-2, ОПК-3 (см. п. III).

Тестовое задание включает два варианта по 20 вопросов по всем темам курса. К тесту допускаются студенты, задавшие все домашние заданий и получившие по каждому заданию зачет.

Критерий оценивания тестового экзаменационного задания

№	Тип задания	Критерии оценки	Результат оценивания
1	Задание закрытого типа на установление соответствия	Считается верным, если правильно установлены все соответствия (позиции одного столбца верно соотнесены с позициями другого столбца)	Полное совпадение с верным ответом – 1 балл Все остальные случаи – 0 баллов
2	Задание закрытого типа на установление последовательности	Считается верным, если правильно указана вся последовательность цифр	Полное совпадение с верным ответом – 1 балл Все остальные случаи – 0 баллов
3	Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора	Считается верным, если правильно указана цифра (буква) правильного ответа и приведены корректные аргументы, используемые при выборе ответа	Полное совпадение с верным ответом – 1 балл Все остальные случаи – 0 баллов
4	Задание комбинированного типа с выбором нескольких верных ответов из четырех предложенных и обоснованием выбора	Считается верным, если правильно указаны цифры (буквы) правильного ответа и приведены корректные аргументы, используемые при выборе ответа	Полное совпадение с верным ответом – 1 балл Все остальные случаи – 0 баллов
5	Задание открытого типа с развернутым ответом	Считается верным, если ответ совпадает с эталонным ответом по содержанию и полноте	Полное соответствие эталонному ответу – 1 балл Все остальные случаи – 0 баллов

Система получения баллов за тестирование

Оценка	критерий
отлично	18 и более баллов
хорошо	16 – 17 баллов
удовлетворительно	15 – 13 баллов
неудовлетворительно	12 баллов и менее

3.1 Оценочные материалы для промежуточной аттестации (экзамена)

Тестирование (Вариант 1).

Индекс и содержание формируемой компетенции	Индикаторы компетенций	Тестовые задания для промежуточной аттестации
ОПК-2 Способен использовать специализированные знания фундаментальных разделов математики, физики, химии и биологии для проведения исследований в области биоинженерии, биоинформатики и смежных дисциплин (модулей)	*ИДК ОПК-2.1* Демонстрирует специализированные знания в области фундаментальных разделов математики, физики, химии, биологии и перспективы междисциплинарных исследований *ИДК ОПК-2.2* Умеет использовать навыки проведения исследований в области биоинженерии, биоинформатики с учетом специализированных фундаментальных знаний *ИДК ОПК-2.3* Владеет методами химии, физики и математического моделирования для проведения исследований в области биоинженерии, биоинформатики	**Задание комбинированного типа с выбором одного или нескольких верных ответов из четырех предложенных с аргументацией выбора** ***Вопрос 1.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой из следующих способов является наиболее рекомендуемым для установки пакета dplyr из CRAN в R? а) source("http://cran.r-project.org/package=dplyr") б) install.packages("dplyr") в) library("dplyr") г) update.packages("dplyr") Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: б) Обоснование: • а) source() используется для выполнения R-скриптов, а не для установки пакетов. • б) install.packages("dplyr") - это стандартный и рекомендуемый способ установки пакетов из CRAN. Он скачивает и устанавливает пакет dplyr и все необходимые зависимости. • в) library("dplyr") - загружает установленный пакет в текущую сессию R, а не устанавливает его. Пакет должен быть установлен до использования library(). • г) update.packages("dplyr") - обновляет уже установленный пакет dplyr до последней версии. ***Вопрос 2.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой фрагмент кода с циклом while в R корректно выводит числа от 1 до 5 (включительно)? а) i <- 1; while (i <= 5) { print(i); i <- i + 1 } б) i <- 1; while (i < 5) { print(i); i <- i + 1 } в) i <- 1; while (i <= 5) { print(i) } г) i <- 1; while (TRUE) { print(i); i <- i + 1; if (i > 5) break } Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: а) и г) Обоснование:
ОПК-3 Способен проводить	*ИДК ОПК-3.1* Проводит	

экспериментальную работу с организмами и клетками, использовать физико-химические методы исследования макромолекул, математические методы обработки результатов биологических исследований	экспериментальную работу с организмами и клетками с использованием физико-химических методов исследования макромолекул *ИДК ОПК-3.2* Демонстрирует практические навыки математических методов обработки результатов экспериментальных исследований *ИДК ОПК-3.3* Владеет опытом применения методов для исследования макромолекул, обработки результатов биологических исследований, прогнозирования перспектив и социальных последствий своей профессиональной деятельности.	• а) Это верный код. Цикл while (i <= 5) будет выполняться, пока i меньше или равно 5. Внутри цикла print(i) выводит текущее значение i, а i <- i + 1 увеличивает i на 1. • б) Этот код выводит числа только до 4, поскольку условие i < 5 означает, что цикл остановится, когда i достигнет значения 5. • в) Этот код создает бесконечный цикл, так как значение i никогда не изменяется, и условие i <= 5 всегда будет истинным. • г) Это тоже верный код. Здесь используется бесконечный цикл while (TRUE), но внутри цикла есть проверка if (i > 5) break. Когда i становится больше 5, оператор break прерывает выполнение цикла. ***Вопрос 3.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Вопрос: Что делает оператор break внутри цикла в R? а) Прерывает текущую итерацию цикла и переходит к следующей. б) Завершает выполнение всего цикла. в) Возвращает значение из цикла. г) Переходит к началу текущей итерации цикла. Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: б) Обоснование: • а) Это описание оператора next (см. следующий вопрос), а не break. • б) Это верное описание оператора break. break полностью завершает выполнение цикла, независимо от того, сколько итераций еще осталось. • в) Оператор break не возвращает значение. • г) Это также описание оператора next ***Вопрос 4.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Что произойдет, если вы попытаетесь создать вектор в R, объединив числовые и символьные значения (например, c(1, "a", 2, "b"))? а) Будет создана ошибка, так как вектор не может содержать элементы разных типов. б) Будет создан числовой вектор, в котором символьные значения будут автоматически преобразованы в числа (NA). в) Будет создан символьный вектор, в котором все числовые значения будут преобразованы в символьные строки. г) Будет создан список, содержащий как числовые, так и символьные значения. Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: в) Обоснование: • а) Ошибка не возникнет. R автоматически преобразует элементы к общему типу.

		• б) Преобразование в NA не произойдет. • в) Это верное утверждение. Векторы в R могут содержать только элементы одного типа. Когда вы пытаетесь объединить числовые и символьные значения, R приводит все элементы к наиболее общему типу - в данном случае, символьному (character). • г) Создастся символьный вектор, а не список. ***Вопрос 5.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой из следующих синтаксисов корректно определяет функцию в R, которая принимает два аргумента x и y, и возвращает их сумму? а) sum_function(x, y) { return(x + y) } б) sum_function <- function(x, y) { return(x + y) } в) function sum_function(x, y) { return x + y } г) sum_function = x, y -> { return(x + y) } Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: б) Обоснование: • а) Отсутствует оператор присваивания <- или =. Это создает функцию, но она никуда не сохраняется. • б) Это верный синтаксис. sum_function <- function(x, y) { return(x + y) } правильно определяет функцию с именем sum_function, принимающую аргументы x и y, и возвращающую их сумму (последнее выражение в блоке кода функции автоматически возвращается). • в) function sum_function(x, y) - Неправильный порядок слов и отсутствие оператора присваивания. return x + y также не использует скобки, что не является распространенной практикой. • г) sum_function = x, y -> { return(x + y) } - Это синтаксис, похожий на лямбда-функции в других языках программирования (например, Python или JavaScript), но он не является корректным синтаксисом для определения функций в R. ***Вопрос 6.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какие из перечисленных функций в R используются для вычисления основных показателей описательной статистики для числовых данных? (Выберите все верные ответы) а) mean() б) median() в) mode() г) sd()

		Выберите все верные ответы и обоснуйте свой выбор: Правильный ответ: а), б), г) Обоснование: • а) mean() - Это верная функция. Она вычисляет среднее арифметическое значение вектора числовых данных. • б) median() - Это верная функция. Она вычисляет медиану вектора числовых данных. • в) mode() - Эта функция в R определяет тип данных переменной (например, "numeric", "character", "logical"), а не моду (наиболее часто встречающееся значение). • г) sd() - Это верная функция. Она вычисляет стандартное отклонение вектора числовых данных. ***Вопрос 7.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какая функция в R используется для создания гистограммы? а) plot() б) hist() в) boxplot() г) density() Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: б) Обоснование: • а) plot() - plot() - это универсальная функция для построения графиков, но она не предназначена специально для гистограмм. Её можно использовать, но потребуются дополнительные параметры. • б) hist() - Это верный ответ. hist(x) создаст гистограмму для числового вектора x. • в) boxplot() - boxplot() используется для создания ящиков с усами (диаграмм размаха), а не гистограмм. • г) density() - density() оценивает плотность вероятности, но сама по себе не строит гистограмму. Обычно используется с plot() для визуализации оценки плотности. ***Вопрос 8.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какая функция в R используется для вычисления матрицы расстояний между строками в таблице данных? а) dist() б) distance() в) cor() г) scale() Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: а)

		Обоснование: • а) Это верный ответ. dist() - функция в R, предназначенная для вычисления матрицы расстояний. • б) distance() - Такой функции в базовом R нет. Она может быть доступна в каких-либо пакетах, но не является стандартной. • в) cor() - Вычисляет матрицу корреляций, а не расстояний. • г) scale() - Стандартизирует данные (вычитает среднее и делит на стандартное отклонение), но не вычисляет расстояния. ***Вопрос 9.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какие типы расстояний можно вычислить с помощью функции dist() в R? (Выберите все верные ответы) а) Евклидово расстояние (Euclidean distance) б) Манхэттенское расстояние (Manhattan distance) в) Косинусное расстояние (Cosine distance) г) Расстояние Чебышева (Chebyshev distance) Выберите все верные ответы и обоснуйте свой выбор: Правильный ответ: а), б), г) Обоснование: • а) Это верный ответ. Евклидово расстояние является стандартной метрикой, используемой по умолчанию функцией dist(). • б) Это верный ответ. Манхэттенское расстояние (также известное как расстояние городских кварталов) можно вычислить с помощью dist(method = "manhattan"). • в) Косинусное расстояние не поддерживается непосредственно функцией dist(). Для вычисления косинусного расстояния потребуются другие функции или пакеты (например, lsa или написание собственной функции). • г) Это верный ответ. Расстояние Чебышева можно вычислить с помощью dist(method = "maximum"). Метод "maximum" соответствует расстоянию Чебышева. ***Вопрос 10.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* В колл-центр поступают звонки. Известно, что в среднем поступает 6 звонков в течение 30-минутного интервала. Предполагается, что число звонков подчиняется распределению Пуассона. Какие из следующих утверждений о данной ситуации являются ВЕРНЫМИ? А) Вероятность того, что в течение следующих 30 минут поступит ровно 4 звонка, составляет приблизительно 0.1339 (или 13.39%). В) Среднее количество звонков, поступающих в течение ЧАСА, равно 3.

		C) Вероятность того, что в течение 15 минут поступит больше двух звонков, можно рассчитать, суммируя вероятности P(X = 3) + P(X = 4) + P(X = 5) + … до бесконечности, где среднее значение для 15-минутного интервала равно 1.5. D) Дисперсия количества звонков, поступающих в течение 30 минут, равна 36. Правильный ответ: A), C) Обоснование: • Почему вариант A верен: ○ Аргументация: По условию, λ = 6 (среднее число звонков за 30 минут). Формула Пуассона: P(X = k) = (e^(-λ) * λ^k) / k!. Для k = 4, P(X = 4) = (e^(-6) * 6^4) / 4! Выполняя расчеты, получаем, что P(X = 4) ≈ 0.1339. То есть вероятность получить ровно 4 звонка за 30 минут примерно равна 13.39%. • Почему вариант B неверен: ○ Аргументация: В условии сказано, что λ = 6 за 30 минут. Один час содержит два 30-минутных интервала. Таким образом, среднее число звонков в час должно быть 6 * 2 = 12, а не 3. • Почему вариант C верен: ○ Аргументация: В условии дано, что в среднем 6 звонков поступают в течении 30 минут. Значит, в течении 15 минут, в среднем, поступит 6 / 2 = 3 звонка. Утверждение о том, что среднее равно 1.5 - неверно! Вероятность того, что поступит БОЛЬШЕ двух звонков в течении 15 минут, вычисляется как сумма вероятностей всех событий, начиная с трех звонков и до бесконечности. То есть P(X > 2) = P(X = 3) + P(X = 4) + P(X = 5) + … Но чтобы это вычислить, нужно правильно вычислить среднее число звонков в течении 15 минут, и оно равно 3. • Почему вариант D неверен: ○ Аргументация: Для распределения Пуассона среднее значение РАВНО дисперсии. В задаче дано, что среднее значение количества звонков за 30 минут равно 6. Следовательно, дисперсия также должна быть равна 6, а не 36. ***Вопрос 11.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Каково основное назначение метрического многомерного шкалирования (MDS)? а) Уменьшение размерности данных, сохраняя при этом наиболее важные переменные. б) Визуализация схожести (или несхожести) между объектами в низкоразмерном пространстве. в) Кластеризация объектов на основе их сходства. г) Прогнозирование значений одной переменной на основе значений других переменных. Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: б) Обоснование: • а) Уменьшение размерности данных - Это функция многих методов, включая PCA, но не является *основной*

целью MDS. MDS может уменьшать размерность, но в первую очередь фокусируется на визуализации схожести.

- б) Это верный ответ. MDS стремится представить объекты в низкоразмерном пространстве (обычно 2D или 3D) таким образом, чтобы расстояния между объектами в этом пространстве максимально соответствовали их исходным мерам несхожести (dissimilarity).
- в) Кластеризация объектов - Это функция кластерного анализа, хотя результаты MDS могут быть использованы для кластеризации, это не основное назначение MDS.
- г) Прогнозирование значений одной переменной - Это задача регрессии, а не MDS.

Задание закрытого типа на установление соответствия

Вопрос12.

Прочитайте вопрос и установите соответствие.

Установите соответствие между задачей и наиболее подходящим типом циклического оператора в R для ее решения.

Задачи:

1. Вычислить сумму элементов вектора.
2. Читать строки из файла до тех пор, пока файл не закончится.
3. Повторять попытки подключения к серверу, пока соединение не будет установлено (максимум 10 попыток).
4. Выполнять итерацию, пока не будет введено корректное значение (проверка ввода).

Операторы (могут быть использованы несколько раз):

A. for B. while C. repeat

Правильный ответ:

- 1 - A
- 2 - B
- 3 - B
- 4 - C

Вопрос 13.

Прочитайте вопрос и установите соответствие.

Сопоставьте каждый тип дистанции (колонка А) с ее наиболее точным описанием (колонка В). Запишите букву, соответствующую описанию, в колонку "Соответствие".

Колонка А: Тип дистанции	**Колонка В: Описание**	**Соответствие**
1. Евклидово расстояние	А. Инвариантна к масштабированию и вращению векторов; игнорирует величину, фокусируясь на ориентации в пространстве.	
2. Манхэттенское	В. Эквивалентна прямой линии между точками; чувствительна к	

расстояние	единицам измерения, сильнее выделяет большие разности.	
3. Косинусное сходство (Distance)	C. Минимизирует влияние выбросов; использует только абсолютные разности, двигаясь "шагами" вдоль осей координат.	
4. Расстояние Махаланобиса	D. Учитывает ковариационную структуру данных; может быть использована для выявления выбросов, но требует обратимости матрицы ковариаций.	

Правильный ответ:

Колонка A: Тип дистанции	**Колонка B: Описание**	**Соответствие**
1. Евклидово расстояние	A. Инвариантна к масштабированию и вращению векторов; игнорирует величину, фокусируясь на ориентации в пространстве.	B
2. Манхэттенское расстояние	B. Эквивалентна прямой линии между точками; чувствительна к единицам измерения, сильнее выделяет большие разности.	C
3. Косинусное сходство (Distance)	C. Минимизирует влияние выбросов; использует только абсолютные разности, двигаясь "шагами" вдоль осей координат.	A
4. Расстояние Махаланобиса	D. Учитывает ковариационную структуру данных; может быть использована для выявления выбросов, но требует обратимости матрицы ковариаций.	D

Вопрос 14.

Прочитайте вопрос и установите соответствие.

Сопоставьте каждую графическую функцию R (колонка A) с ее наиболее точным описанием (колонка B). Запишите букву, соответствующую описанию, в колонку "Соответствие".

Колонка A: Функция R	**Колонка B: Описание**	**Соответствие**
1. plot()	A. Строит гистограмму частот для числового вектора; позволяет настраивать количество столбцов и их границы.	
2. hist()	B. Рисует ящики с усами (boxplots) для сравнения распределений одного или нескольких наборов данных; отображает медиану, квартили и выбросы.	
3. boxplot()	C. Создает диаграмму рассеяния (scatterplot) для двух переменных; является универсальной функцией для создания различных типов графиков в зависимости от типа входных данных.	
4. barplot()	D. Отображает столбчатую диаграмму для представления категориальных данных; высота столбцов соответствует частоте или	

	пропорции каждой категории.	

Правильный ответ:

Колонка A: Функция R	Колонка B: Описание	Соответствие
1. plot()	A. Строит гистограмму частот для числового вектора; позволяет настраивать количество столбцов и их границы.	C
2. hist()	B. Рисует ящики с усами (boxplots) для сравнения распределений одного или нескольких наборов данных; отображает медиану, квартили и выбросы.	A
3. boxplot()	C. Создает диаграмму рассеяния (scatterplot) для двух переменных; является универсальной функцией для создания различных типов графиков в зависимости от типа входных данных.	B
4. barplot()	D. Отображает столбчатую диаграмму для представления категориальных данных; высота столбцов соответствует частоте или пропорции каждой категории.	D

Вопрос 15.

Прочитайте вопрос и установите соответствие.

Сопоставьте каждый тип распределения случайной величины (колонка A) с его наиболее точным описанием или характерной особенностью (колонка B). Запишите букву, соответствующую описанию, в колонку "Соответствие".

Колонка A: Тип распределения	Колонка B: Описание / Характеристика	Соответствие
1. Нормальное распределение	A. Моделирует количество успехов в фиксированном числе независимых испытаний Бернулли.	
2. Равномерное распределение	B. Все значения в заданном интервале равновероятны; часто используется как пример "случайного" выбора.	
3. Биномиальное распределение	C. Широко известно своей "колоколообразной" формой; характеризуется средним значением и стандартным отклонением.	
4. Распределение Пуассона	D. Моделирует количество событий, происходящих в фиксированный период времени или месте, при условии что эти события происходят с известной средней интенсивностью.	

Правильный ответ:

Колонка A: Тип распределения	Колонка B: Описание / Характеристика	Соответствие
1. Нормальное распределение	A. Моделирует количество успехов в фиксированном числе независимых испытаний Бернулли.	C
2. Равномерное распределение	B. Все значения в заданном интервале равновероятны; часто используется как пример "случайного" выбора.	B
3. Биномиальное распределение	C. Широко известно своей "колоколообразной" формой; характеризуется средним значением и стандартным отклонением.	A
4. Распределение Пуассона	D. Моделирует количество событий, происходящих в фиксированный период времени или месте, при условии что эти события происходят с известной средней интенсивностью.	D

Задание закрытого типа на установление последовательности

Вопрос 16.

Прочитайте вопрос и установите последовательность.

Расположите шаги вычисления матрицы евклидовых расстояний между образцами (строками) в data.frame df с использованием функций пакета vegan в R в правильном порядке.

Шаг	Описание	Порядок
A. vegdist(df, method = "euclidean")	Вычисление матрицы расстояний с использованием евклидовой метрики на основе данных в df.	
B. as.matrix(dist_matrix)	Преобразование объекта класса "dist" (или "vegdist") в матрицу.	
C. library(vegan)	Загрузка пакета vegan.	
D. df <- data.frame(Ваши данные)	Создание (или загрузка) data.frame с данными. (Предполагается, что Ваши данные уже существуют или готовы к созданию).	
E. dist_matrix <- vegdist(df, method = "euclidean")	Присваивание полученной матрицы расстояний переменной dist_matrix.	

Правильный ответ:

Шаг	Описание	Порядок
A. vegdist(df, method = "euclidean")	Вычисление матрицы расстояний с использованием евклидовой метрики на основе данных в df.	4
B. as.matrix(dist_matrix)	Преобразование объекта класса "dist" (или "vegdist")	5

	в матрицу.	
C. library(vegan)	Загрузка пакета vegan.	2
D. df <- data.frame(Ваши данные)	Создание (или загрузка) data.frame с данными. (Предполагается, что Ваши данные уже существуют или готовы к созданию).	1
E. dist_matrix <- vegdist(df, method = "euclidean")	Присваивание полученной матрицы расстояний переменной dist_matrix.	3

Вопрос 17.

Прочитайте вопрос и установите последовательность.

Расположите шаги выполнения метрического многомерного шкалирования (метрического MDS) в R в правильном порядке. Предположим, что у вас есть матрица расстояний distance_matrix, полученная, например, с помощью функции dist() или vegdist().

Шаг	Описание	Порядок
A. mds_result <- cmdscale(distance_matrix, k = 2)	Применение функции cmdscale() для выполнения метрического MDS. Установите k = 2 для получения двумерной конфигурации.	
B. plot(mds_result[, 1], mds_result[, 2], ...)	Визуализация результатов MDS. mds_result[,1] и mds_result[,2] содержат координаты в двух измерениях.	
C. coords <- as.data.frame(mds_result)	Преобразование результатов MDS в data.frame для удобства работы.	
D. distance_matrix <- dist(data, method = "euclidean")	Вычисление матрицы расстояний из исходных данных (если у вас есть только исходные данные, а не матрица расстояний).	
E. data <- read.csv("your_data.csv")	Чтение исходных данных из CSV-файла (если у вас есть только исходные данные, а не матрица расстояний).	

Правильный ответ:

Шаг	Описание	Порядок
A. mds_result <- cmdscale(distance_matrix, k = 2)	Применение функции cmdscale() для выполнения метрического MDS. Установите k = 2 для получения двумерной конфигурации.	3
B. plot(mds_result[, 1], mds_result[, 2], ...)	Визуализация результатов MDS. mds_result[,1] и mds_result[,2] содержат координаты в двух измерениях.	5

C. coords <- as.data.frame(mds_result)	Преобразование результатов MDS в data.frame для удобства работы.	4
D. distance_matrix <- dist(data, method = "euclidean")	Вычисление матрицы расстояний из исходных данных (если у вас есть только исходные данные, а не матрица расстояний).	2
E. data <- read.csv("your_data.csv")	Чтение исходных данных из CSV-файла (если у вас есть только исходные данные, а не матрица расстояний).	1

Вопрос 18.

Прочитайте вопрос и установите последовательность.

Расположите шаги вычисления доверительного интервала для среднего значения выборки с использованием t-распределения в R в правильном порядке.

Шаг	Описание	Порядок
A. t_value <- qt(1 - alpha/2, df = n - 1)	Вычисление критического значения t-статистики для заданного уровня значимости alpha и степеней свободы df = n - 1.	
B. lower_bound <- sample_mean - t_value * (sample_sd / sqrt(n))	Вычисление нижней границы доверительного интервала.	
C. sample_mean <- mean(data)	Вычисление среднего значения выборки.	
D. alpha <- 0.05	Определение уровня значимости (например, 0.05 для 95% доверительного интервала).	
E. data <- c(Ваши данные)	Создание или загрузка данных выборки. (Предполагается, что Ваши данные представляют собой числовой вектор или могут быть легко преобразованы в него).	
F. sample_sd <- sd(data)	Вычисление стандартного отклонения выборки.	
G. n <- length(data)	Определение размера выборки.	
H. upper_bound <- sample_mean + t_value * (sample_sd / sqrt(n))	Вычисление верхней границы доверительного интервала.	
I. cat("Доверительный интервал:", lower_bound, upper_bound)	Вывод результатов: нижней и верхней границ доверительного интервала.	

Правильный ответ:

Шаг	Описание	Порядок
A. t_value <- qt(1 - alpha/2, df = n - 1)	Вычисление критического значения t-статистики для заданного уровня значимости alpha и степеней	5

	свободы df = n - 1.	
B. lower_bound <- sample_mean - t_value * (sample_sd / sqrt(n))	Вычисление нижней границы доверительного интервала.	6
C. sample_mean <- mean(data)	Вычисление среднего значения выборки.	2
D. alpha <- 0.05	Определение уровня значимости (например, 0.05 для 95% доверительного интервала).	1
E. data <- c(Ваши данные)	Создание или загрузка данных выборки. (Предполагается, что Ваши данные представляют собой числовой вектор или могут быть легко преобразованы в него).	0
F. sample_sd <- sd(data)	Вычисление стандартного отклонения выборки.	3
G. n <- length(data)	Определение размера выборки.	4
H. upper_bound <- sample_mean + t_value * (sample_sd / sqrt(n))	Вычисление верхней границы доверительного интервала.	7
I. cat("Доверительный интервал:", lower_bound, upper_bound)	Вывод результатов: нижней и верхней границ доверительного интервала.	8

Вопрос 19.

Прочитайте вопрос и установите последовательность.

Укажите правильную последовательность шагов для построения гистограммы с наложением эмпирической функции плотности вероятности (PDF) на основе набора числовых данных.

Исходные данные: Предположим, у вас есть набор данных, представляющий собой выборку значений некоторой случайной величины.

Шаги (перемешаны):

A. Нормировка гистограммы: Разделите частоту каждого столбца гистограммы на общее количество наблюдений (размер выборки). Это преобразует гистограмму в гистограмму относительных частот.

B. Построение графика эмпирической PDF: Используйте оценку плотности ядра (Kernel Density Estimation - KDE) для аппроксимации эмпирической функции плотности вероятности на основе данных. Наложите график PDF на гистограмму.

C. Определение интервалов (бинов): Разбейте диапазон значений данных на несколько неперекрывающихся интервалов (бинов). Количество интервалов должно быть выбрано разумно, чтобы не потерять детали распределения (слишком мало интервалов) и не создать шум (слишком много интервалов).

D. Подсчет частот: Определите, сколько значений данных попадает в каждый интервал. Это и есть частота для каждого интервала.

E. Построение гистограммы: Создайте столбчатую диаграмму, где горизонтальная ось представляет интервалы, а высота каждого столбца пропорциональна частоте значений в этом интервале (или относительной частоте после нормировки).

		Правильный ответ: C → D → E → A → B **Задание открытого типа с развернутым ответом** ***Вопрос 20.*** *Прочитайте вопрос и запишите развернутый обоснованный ответ.* Опишите циклический оператор for в языке программирования? Ответ: Циклический оператор for в R – это основной инструмент для повторения блока кода заданное количество раз. Он позволяет эффективно обрабатывать массивы, списки и другие структуры данных, выполняя одни и те же операции над каждым элементом. Синтаксис for цикла в R достаточно прост и понятен: for (переменная in последовательность) { действия }. Здесь переменная принимает значение каждого элемента из последовательности на каждой итерации цикла, а действия – это блок кода, который будет выполнен. ***Вопрос 21.*** *Прочитайте вопрос и запишите развернутый обоснованный ответ.* Определение понятия Евклидово расстояние? Ответ: Евклидово расстояние (также известное как "расстояние по прямой") – это наиболее интуитивно понятная и часто используемая мера расстояния между двумя точками в Евклидовом пространстве. Оно представляет собой длину отрезка прямой, соединяющего эти две точки. Определение: Формально, евклидово расстояние между двумя точками $P = (p_1, p_2, \ldots, p_n)$ и $Q = (q_1, q_2, \ldots, q_n)$ в n-мерном Евклидовом пространстве определяется как: $d(P, Q) = \sqrt{((q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_n - p_n)^2)}$ ***Вопрос 22.*** *Прочитайте вопрос и запишите развернутый обоснованный ответ.* Дайте определение понятя марщрут в математичеком графе? Ответ: В теории графов, маршрут (или путь) — это последовательность вершин в графе, соединенных ребрами. Определение: Маршрутом в графе G = (V, E) называется последовательность вершин $v_1, v_2, \ldots, v_k$, где для каждого i от 1 до k-1, $(v_i, v_{i+1}) \in E$ (т.е. существует ребро между вершинами v_i и v_{i+1}). ***Вопрос 23.*** *Прочитайте вопрос и запишите развернутый обоснованный ответ.* Дайте описание гистограмме как способу визуализации распределена элементов в выборках статистичских

		данных? Ответ: Гистограмма распределения – это графическое представление распределения частот числовых данных. Она представляет собой столбчатую диаграмму, где: • **Горизонтальная ось (ось X):** Разделена на интервалы (или "бины," "классы") равной или переменной ширины, охватывающие диапазон значений данных. • **Вертикальная ось (ось Y):** Показывает частоту (количество) значений, попадающих в каждый интервал. Обычно используется частота (количество наблюдений), но также можно использовать относительную частоту (процент наблюдений) или плотность вероятности (для нормализации по площади). • **Столбцы:** Высота каждого столбца пропорциональна частоте (или относительной частоте/плотности) значений в соответствующем интервале. Столбцы гистограммы примыкают друг к другу (если только в данных нет разрывов).

Тестирование (Вариант 2).

Индекс и содержание формируемой компетенции	**Индикаторы компетенций**	**Тестовые задания для промежуточной аттестации**
ОПК-2 Способен использовать специализированные знания фундаментальных разделов математики, физики, химии и биологии для проведения исследований в области биоинженерии, биоинформатики и смежных дисциплин (модулей)	*ИДК ОПК-2.1* Демонстрирует специализированные знания в области фундаментальных разделов математики, физики, химии, биологии и перспективы междисциплинарных исследований *ИДК ОПК-2.2* Умеет использовать навыки проведения исследований в области биоинженерии, биоинформатики с	**Задание комбинированного типа с выбором одного или нескольких верных ответов из четырех предложенных и аргументацией выбора** ***Вопрос 1.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Что произойдет, если при использовании install.packages() в R, пакет, который вы пытаетесь установить, имеет неустановленные зависимости? а) Установка пакета завершится с ошибкой, и R предложит установить зависимости вручную. б) install.packages() автоматически попытается установить все необходимые зависимости из CRAN. в) install.packages() проигнорирует неустановленные зависимости и установит только запрошенный пакет. г) R запросит подтверждение на установку зависимостей перед продолжением установки основного пакета. Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: б) Обоснование: • а) Хотя ошибки могут возникнуть при проблемах с доступом к репозиториям, install.packages() автоматически разрешает и устанавливает зависимости, если они доступны в репозитории (например, CRAN). • б) Это верное описание поведения install.packages(). По умолчанию, функция автоматически загружает и

	учетом специализированных фундаментальных знаний *ИДК ОПК-2.3* Владеет методами химии, физики и математического моделирования для проведения исследований в области биоинженерии, биоинформатики	устанавливает все необходимые зависимости для выбранного пакета. • в) install.packages() не игнорирует зависимости. Пакет может работать неправильно или вообще не работать, если его зависимости не установлены. • г) install.packages() по умолчанию не запрашивает подтверждение на установку зависимостей, а делает это автоматически. ***Вопрос 2.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой из следующих вариантов кода корректно использует цикл for в R для вывода квадратов чисел от 1 до 5 (включительно)? а) for (i in 1:5) { print(i^2) } б) for (i = 1, i <= 5, i++) { print(i^2) } в) for (i in range(1, 6)) { print(i^2) } г) for i = 1 to 5 { print(i^2) } Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: а) Обоснование: • а) Это верный код. Цикл for (i in 1:5) корректно итерируется по последовательности чисел от 1 до 5, и print(i^2) выводит квадрат каждого числа. • б) Синтаксис for (i = 1, i <= 5, i++) не является корректным синтаксисом цикла for в R. Это больше похоже на синтаксис C-подобных языков. • в) Функция range() не существует в базовом R для создания последовательности чисел. В R для этого используется оператор :. • г) Синтаксис for i = 1 to 5 не является корректным синтаксисом цикла for в R. Это синтаксис, используемый в некоторых других языках программирования, но не в R. ***Вопрос 3.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Что делает оператор next внутри цикла в R? а) Завершает выполнение всего цикла. б) Прерывает текущую итерацию цикла и переходит к следующей. в) Возвращает значение из цикла. г) Выполняет следующую итерацию цикла, пропуская оставшуюся часть текущей итерации. Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: б) и г) Обоснование: • а) Это описание оператора break. • б) и г) Оба варианта по сути говорят об одном и том же, просто разными словами. Оператор next пропускает оставшуюся часть кода в текущей итерации цикла и сразу переходит к следующей итерации. Поэтому можно сказать, что он "прерывает текущую итерацию и переходит к следующей".
	ОПК-3 Способен проводить экспериментальную работу с организмами и клетками, использовать физико-химические методы исследования макромолекул, математические методы обработки результатов биологических исследований *ИДК ОПК-3.2* Демонстрирует практические навыки математических методов обработки результатов экспериментальных	

	исследований *ИДК ОПК-3.3* Владеет опытом применения методов для исследования макромолекул, обработки результатов биологических исследований, прогнозирования перспектив и социальных последствий своей профессиональной деятельности.	• в) Оператор next не возвращает значение. ***Вопрос 4.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какой из следующих способов корректно создает числовой вектор в R, содержащий элементы 1, 2, 3, 4 и 5? а) vector <- c(1, 2, 3, 4, 5) б) vector = list(1, 2, 3, 4, 5) в) vector <- 1:5 г) vector = seq(1, 5) Выберите все верные ответы и обоснуйте свой выбор: Правильный ответ: а), в), г) Обоснование: • а) vector <- c(1, 2, 3, 4, 5) - Это верный способ. Функция c() используется для объединения элементов в вектор. • б) vector = list(1, 2, 3, 4, 5) - Это неверный способ. list() создает список, а не вектор. Списки могут содержать элементы разных типов, в то время как векторы обычно содержат элементы одного типа. • в) vector <- 1:5 - Это верный способ. Оператор : создает последовательность чисел от 1 до 5 с шагом 1. • г) vector = seq(1, 5) - Это верный способ. Функция seq() создает последовательность чисел от 1 до 5 (по умолчанию с шагом 1). ***Вопрос 5.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Как определить функцию в R, которая принимает аргумент x и необязательный аргумент y со значением по умолчанию равным 10? а) my_function <- function(x, y = 10) { ... } б) my_function <- function(x, optional y = 10) { ... } в) my_function <- function(x, y = NULL) { if(is.null(y)) y <- 10; ... } г) my_function <- function(x = 10, y) { ... } Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: а) Обоснование: • а) Это верный способ. my_function <- function(x, y = 10) { ... } задает значение по умолчанию для аргумента y равным 10. Если при вызове функции аргумент y не будет указан, он автоматически примет значение 10. • б) optional y = 10 - Некорректный синтаксис. Слово optional не используется в R для определения аргументов по умолчанию. • в) y = NULL и проверка на is.null(y) - Это рабочий способ, но он более громоздкий, чем использование аргумента по умолчанию напрямую. y = 10 при определении функции - более лаконичный и предпочтительный вариант. • г) x = 10, y - Нельзя, чтобы аргумент без значения по умолчанию шел после аргумента со значением по умолчанию. Порядок имеет значение. В R сначала должны идти обязательные аргументы, а затем - аргументы со значениями по умолчанию. ***Вопрос 6.***

		Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа. Как в R можно обработать пропущенные значения (NA) при вычислении среднего значения с помощью функции mean()? а) Функция mean() автоматически игнорирует NA значения. б) Необходимо удалить NA значения из данных перед вычислением среднего значения. в) Нужно установить аргумент na.rm = TRUE в функции mean(). г) Нужно заменить NA значения нулями перед вычислением среднего значения. Выберите один верный ответ и обоснуйте свой выбор: Правильный ответ: в) Обоснование: • а) Функция mean() по умолчанию возвращает NA, если в данных есть пропущенные значения. • б) Удаление NA значений может быть необходимо в некоторых случаях, но не является единственным или всегда лучшим решением. Удаление данных может привести к потере информации. • в) Это верный способ. Установка аргумента na.rm = TRUE указывает функции mean() игнорировать NA значения при вычислении среднего значения. • г) Замена NA нулями может исказить результаты, особенно если NA представляют собой отсутствие данных, а не реальные нулевые значения. ***Вопрос 7.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какая функция в R используется для создания диаграммы размаха (боксплота)? а) hist() б) plot() в) boxplot() г) pie() Выберите один верный ответ и обоснуйте свой выбор: Ваш ответ: в) Обоснование: • а) hist() - Используется для создания гистограмм. • б) plot() - Можно использовать, но требует дополнительных параметров для создания боксплота. • в) Это верный ответ. Функция boxplot() создает диаграмму размаха. • г) pie() - Используется для создания круговых диаграмм. ***Вопрос 8.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какие утверждения о параметре lambda функции rpois() в R являются верными? A) lambda должен быть целым числом. B) lambda должен быть неотрицательным числом. C) lambda определяет среднее значение и дисперсию распределения Пуассона. D) Если lambda равно нулю, функция rpois() всегда возвращает вектор, состоящий только из нулей. Ответ: B, C, D Аргументация:

		○ B: lambda должен быть неотрицательным, поскольку это среднее значение числа событий, и отрицательным оно быть не может. ○ C: lambda является параметром, определяющим среднее значение и дисперсию распределения Пуассона (они равны lambda). ○ D: Если lambda равно 0, ожидаемое количество событий равно нулю. Значит каждое сгенерированное число будет равно нулю, что и вернет функция. ○ A: lambda не обязательно должно быть целым числом, оно может быть и дробным (вещественным) числом. ***Вопрос 9.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* В каких случаях рекомендуется нормализовать данные перед вычислением расстояний (например, евклидова расстояния)? (Выберите все верные ответы) а) Когда переменные имеют разные единицы измерения. б) Когда переменные имеют сильно различающиеся масштабы. в) Когда все переменные имеют одинаковые единицы измерения и масштабы. г) Нормализация всегда необходима перед вычислением расстояний. Выберите все верные ответы и обоснуйте свой выбор: Правильный ответ: а), б) Обоснование: • а) Это верный ответ. Если переменные измерены в разных единицах (например, метры и килограммы), нормализация необходима, чтобы одна переменная не доминировала при вычислении расстояния. • б) Это верный ответ. Если переменные имеют сильно различающиеся масштабы (например, возраст от 0 до 100 и доход от 0 до 1,000,000), переменная с большим масштабом будет оказывать непропорционально большое влияние на вычисление расстояния. • в) Когда переменные имеют одинаковые единицы измерения и масштабы, нормализация не всегда необходима, но может быть полезна в некоторых случаях. • г) Нормализация не всегда необходима. Если переменные имеют сопоставимые масштабы и единицы измерения, можно вычислять расстояния без нормализации. Однако, нормализация может улучшить результаты в некоторых случаях, даже если масштабы похожи. ***Вопрос 10.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какие из следующих утверждений о манхэттенском расстоянии в контексте анализа многомерных данных являются верными? A) Манхэттенское расстояние всегда меньше или равно евклидову расстоянию между двумя точками. B) Манхэттенское расстояние менее чувствительно к выбросам, чем евклидово расстояние. C) Манхэттенское расстояние инвариантно к поворотам системы координат. D) Манхэттенское расстояние вычисляется как сумма абсолютных разностей координат между двумя точками. Правильный ответ: D)

		Обоснование: • Выбор D: Манхэттенское расстояние, по определению, вычисляется как сумма абсолютных разностей координат между двумя точками. Это прямо следует из формулы: d(x, y) = |x1 - y1| + |x2 - y2| + … + |xn - yn|. • Отклонение A: Манхэттенское расстояние, как правило, больше Евклидова, но, если точки лежат на одной прямой, параллельной оси координат, расстояния будут равны. • Отклонение B: Манхэттенсткое расстояние более чувствительно к выбросам, чем Евклидово. Евклидово расстояние возводит разность координат в квадрат, таким образом, расстояние более сильно подвержено сильным отклонениям. • Отклонение C: Манхэттенское расстояние зависит от системы координат и не инвариантно к поворотам. Поворот системы координат изменит разность между значениями координат, соответственно, изменит манхэттенское расстояние. ***Вопрос 11.*** *Прочитайте вопрос, выберите правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа.* Какие типы многомерного шкалирования существуют? (Выберите все верные ответы) а) Метрическое MDS (Metric MDS) б) Неметрическое MDS (Non-metric MDS) в) Линейное MDS (Linear MDS) г) Полиномиальное MDS (Polynomial MDS) Выберите все верные ответы и обоснуйте свой выбор: Правильный ответ: а), б) Обоснование: • а) Это верный ответ. Метрическое MDS (Metric MDS) предполагает, что исходные меры несхожести являются метрическими (т.е. удовлетворяют аксиомам метрики) и стремится сохранить точные расстояния в низкоразмерном пространстве. • б) Это верный ответ. Неметрическое MDS (Non-metric MDS) предполагает, что важен только ранг (порядок) мер несхожести, а не их точные значения. Он стремится сохранить порядок расстояний в низкоразмерном пространстве. • в) Линейное MDS (Linear MDS) - Такого общепринятого типа MDS не существует. MDS может быть реализован с использованием линейных алгебраических методов, но сама классификация не является "линейным MDS". • г) Полиномиальное MDS (Polynomial MDS) - Такого общепринятого типа MDS не существует. **Задание закрытого типа на установление соответствия** ***Вопрос 12.*** *Прочитайте вопрос и установите соответствие.* Установите соответствие между циклическим оператором в R и его описанием. Операторы:

		• for • while • repeat • break Описания: A. Выполняет блок кода до тех пор, пока условие истинно. Проверяет условие *перед* каждой итерацией. B. Немедленно прекращает выполнение цикла и переходит к следующей строке кода после цикла. C. Выполняет блок кода заданное количество раз. D. Выполняет блок кода бесконечно, пока не будет прерван оператором break. Правильный ответ: • 1 - C • 2 - A • 3 - D • 4 - B ***Вопрос 13.*** *Прочитайте вопрос и установите соответствие.* Установите соответствие между функцией для генерации случайных чисел в R (Список А) и типом данных, которые она генерирует (Список Б). Например, если функция генерирует только целые числа, ей соответствует вариант "целые числа", если вещественные - то "вещественные числа" и т.д. Запишите букву, соответствующую типу данных, рядом с каждой функцией. Список А: Функции • rnorm() • sample() • rbinom() • round() • runif() Список Б: Типы генерируемых данных A) Вещественные числа (с плавающей точкой) с нормальным распределением. B) Случайные элементы из заданного вектора (тип данных зависит от элементов вектора) C) Целые числа, представляющие количество успехов в биномиальном распределении. D) Вещественные числа (с плавающей точкой) с равномерным распределением. E) Округленные значения (тип данных зависит от аргументов функции и исходных данных). Задание: Укажите соответствие: • rnorm(): • sample(): • rbinom():

		• round(): • runif(): Правильный ответ: • rnorm(): A • sample(): B • rbinom(): C • round(): E • runif(): D ***Вопрос 14.*** *Прочитайте вопрос и установите соответствие.* Сопоставьте описание типа расстояния, используемого в многомерной статистике (Список А), с соответствующей функцией или аргументом функции vegdist в пакете vegan языка R (Список В). Укажите букву, соответствующую правильной функции/аргументу, рядом с каждым описанием. • Расстояние, учитывающее только наличие/отсутствие видов (бинарные данные) и игнорирующее информацию об их обилии. • Расстояние, чувствительное к различиям в видовом составе и обилии, пропорциональное общему количеству видов, которые не встречаются одновременно в двух сравниваемых сообществах. • Расстояние, менее чувствительное к различиям в видовом составе и обилии, особенно в случае доминирования нескольких видов (редкие виды имеют меньшее влияние) • Расстояние, представляющее собой корень суммы квадратов разностей между значениями соответствующих переменных. Список В: Функции/Аргументы vegdist A) "bray" B) "euclidean" C) "jaccard" (или любой другой бинарный индекс, например, "binary") D) "manhattan" Правильный ответ: • 1: C • 2: D • 3: A • 4: B ***Вопрос 15.*** *Прочитайте вопрос и установите соответствие.* Сопоставьте описание действия или типа графика (Список А) с соответствующей базовой графической функцией в R (Список В). Укажите букву, обозначающую правильную функцию, рядом с каждым описанием.

Список A: Описание действия/типа графика

- Создание базового графика рассеяния (scatter plot) для визуализации взаимосвязи между двумя переменными.
- Добавление линии регрессии на уже существующий график.
- Создание гистограммы для отображения распределения одной переменной.
- Создание графика "ящик с усами" (boxplot) для сравнения распределений одной или нескольких переменных.

Список B: Графические функции
A) hist()
B) plot(x, y)
C) boxplot()
D) abline(lm_object)
Задание:
Укажите соответствие:
Правильный ответ:

- 1: B
- 2: D
- 3: A
- 4: B

Задание закрытого типа на установление последовательности

Вопрос 16.
Прочитайте вопрос и установите последовательность.
Расположите следующие этапы вычисления доверительного интервала для среднего значения, используя нормальное распределение, в правильной последовательности. В таблице напротив каждого шага укажите его порядковый номер. (Т.е. в столбце "Порядок" укажите число от 1 до 5)

Шаг	Порядок
A) Рассчитайте стандартную ошибку среднего (SE).	
B) Определите уровень значимости (alpha) и соответствующий z-критический коэффициент.	
C) Вычислите нижнюю и верхнюю границы доверительного интервала.	
D) Определите набор данных (вектор числовых значений).	
E) Рассчитайте среднее значение выборки.	
F) Рассчитайте стандартное отклонение S	

Правильный ответ:

Шаг	Порядок

A) Рассчитайте стандартную ошибку среднего (SE).	5
B) Определите уровень значимости (alpha) и соответствующий z-критический коэффициент.	4
C) Вычислите нижнюю и верхнюю границы доверительного интервала.	6
D) Определите набор данных (вектор числовых значений).	1
E) Рассчитайте среднее значение выборки.	2
F) Рассчитайте стандартное отклонение S	3

Вопрос 17.

Прочитайте вопрос и установите последовательность.

Предположим, у вас есть матрица диссимилярностей dist_matrix, представляющая различия между объектами. Какова правильная последовательность шагов для применения метрического многомерного шкалирования (Metric Multidimensional Scaling - MDS) с использованием функции cmdscale() в R для визуализации данных в двумерном пространстве?

a) Создать матрицу диссимилярностей -> Применить cmdscale(dist_matrix, k = 2) -> Визуализировать точки с помощью plot()

b) Визуализировать точки с помощью plot() -> Создать матрицу диссимилярностей -> Применить cmdscale(dist_matrix, k = 2)

c) Применить cmdscale(dist_matrix, k = 2) -> Создать матрицу диссимилярностей -> Визуализировать точки с помощью plot()

d) Применить cmdscale(dist_matrix, k = 2) -> Визуализировать точки с помощью plot() -> Создать матрицу диссимилярностей

Правильный ответ: a)

Вопрос 18.

Прочитайте вопрос и установите последовательность.

Расположите следующие шаги в правильной последовательности для вычисления матрицы расстояний Брея-Кертиса между образцами в датафрейме community_data с использованием пакета vegan в R.

A) Убедитесь, что датафрейм содержит только числовые значения (данные о численности видов/обилие).

B) Вызовите функцию vegdist() из пакета vegan, указав датафрейм community_data и метод "bray".

C) Установите и загрузите пакет vegan, если он еще не установлен.

D) Сохраните результат, возвращенный функцией vegdist(), в переменную, например, bray_curtis_dist.

E) Проверьте структуру и первые несколько строк датафрейма community_data, чтобы убедиться в правильности формата данных. В ответе укажите только последовательность букв (например, "A, B, C, D")

Правильный ответ:

C, E, A, B, D

Вопрос 19.

Прочитайте вопрос и установите последовательность.
Расположите следующие шаги в правильной последовательности для построения графика боксплот в R. В таблице напротив каждого шага укажите его порядковый номер.

Шаг	Порядок
A) Добавьте заголовок к графику с помощью функции title().	
B) Импортируйте данные в R (например, из CSV файла).	
C) Выберите переменные для построения боксплота.	
D) Используйте функцию boxplot() для создания графика боксплот.	
E) Установите и загрузите необходимые пакеты (например, ggplot2, если будете использовать его).	

Правильный ответ:

Шаг	Порядок
A) Добавьте заголовок к графику с помощью функции title().	5
B) Импортируйте данные в R (например, из CSV файла).	2
C) Выберите переменные для построения боксплота.	3
D) Используйте функцию boxplot() для создания графика боксплот.	4
E) Установите и загрузите необходимые пакеты (например, ggplot2, если будете использовать его).	1

Задание открытого типа с развернутым ответом

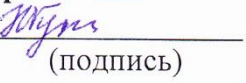
Вопрос 20.
Прочитайте вопрос и запишите развернутый обоснованный ответ.
Опишите циклический оператор while в языке программирования R.
Ответ:
Циклический оператор while в языке программирования R представляет собой фундаментальную конструкцию для организации повторяющихся вычислений, пока определенное условие остается истинным. Он позволяет выполнять блок кода многократно, что незаменимо в ситуациях, когда количество итераций заранее неизвестно и зависит от динамически изменяющихся данных или состояний программы. Синтаксис оператора while: сначала указывается ключевое слово while, за которым в круглых скобках следует логическое выражение, выступающее в качестве условия продолжения цикла. В фигурных скобках заключен блок кода, который будет выполняться на каждой итерации цикла, пока условие истинно.

Вопрос 21.
Прочитайте вопрос и запишите развернутый обоснованный ответ.
Определения понятия расстояние Брея-Кертиса?

		Ответ: Расстояние Брея-Кертиса (Bray-Curtis Dissimilarity) - это мера различия между двумя образцами (обычно экологическими сообществами), основанная на количественных данных. Оно показывает, насколько отличаются относительные значения признаков (например, видов) в двух образцах. Главное отличие от евклидового расстояния в том, что величина учитывает только разницу между двумя выборками, не принимая во внимание обилие видов, общих для обеих выборок. Формула: BC = Σ |xi - yi| / Σ (xi + yi) где: • xi - значение признака i (например, количество особей вида i) в образце X • yi - значение признака i в образце Y • Σ - сумма по всем признакам ***Вопрос 22.*** *Прочитайте вопрос и запишите развернутый обоснованный ответ.* Дате математическое определение понятия графу? Ответ: Граф представляет собой структуру, состоящую из множества объектов (узлов или вершин), между которыми существуют связи (ребра или дуги). Определение. Простым графом G(V, E) называется совокупность двух множеств – непустого множества V и множества E неупорядоченных пар различных элементов множества V . Множество V называется множеством вершин, множество E называется множеством ребер. ***Вопрос 23.*** *Прочитайте вопрос и запишите развернутый обоснованный ответ.* Дате описание способу визуализации Боксплоту (ящики с усами) как инструменту для визуализации распределений элементов в статистических выборках? Ответ: Определение: Боксплот (box plot), или "ящик с усами" – это стандартизированный способ графического отображения распределения данных на основе пяти ключевых значений: • Минимум (Minimum): Наименьшее значение в данных, не являющееся выбросом. • Первый квартиль (Q1): 25-й процентиль данных. 25% данных лежат ниже этого значения. • Медиана (Q2/Median): 50-й процентиль данных. Разделяет данные на две равные части.

		• Третий квартиль (Q3): 75-й процентиль данных. 75% данных лежат ниже этого значения. • Максимум (Maximum): Наибольшее значение в данных, не являющееся выбросом. Элементы боксплота: • Ящик (Box): Ограничен первым (Q1) и третьим (Q3) квартилями. Длина ящика представляет межквартильный размах (IQR = Q3 - Q1), который содержит центральные 50% данных. • Медиана (линия внутри ящика): Отмечает медиану (Q2) распределения. • Усы (Whiskers): Линии, идущие от ящика до наименьшего и наибольшего значений.

Разработчик:


доцент Букин Ю.С.

(подпись)

.