Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и фундаментальной информатики
Базовая кафедра вычислительных и информационных технологий

УТВЕРЖДАЮ
Заведующий кафедрой
_____ / В.В. Шайдуров

«17» июня 2019 г.

БАКАЛАВРСКАЯ РАБОТА

Направление 02.03.01 Математика и компьютерные науки

СТАТИСТИЧЕСКИЙ АНАЛИЗ ДАННЫХ ДЛЯ ИССЛЕДОВАНИЯ СТРУКТУРНОЙ ОРГАНИЗАЦИИ И ПРОСТРАНСТВЕННОЙ ДИНАМИКИ СООБЩЕСТВ ЗООБЕНТОСА В РЕКАХ БАССЕЙНА ЕНИСЕЯ

Научный руководитель
кандидат физико-математических наук,
доцент _____ / Е.Д. Карепова
17.06.19

Выпускник _____ / М.П. Лепьявко
17.06.19

Красноярск 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность выпускной квалификационной работы связано с ролью зообентоса в водных системах. Зообентосом называют беспозвоночных животных, обитающие в водоёмах на поверхности грунта и в его толще. Он представляет собой звено в трофической цепи, способствует естественному самоочищению вод, становятся активными минерализаторами органических веществ и биофильтрами воды и является биоиндикатором экологического состояния разнотипных водных объектов.

Задачей выпускной квалификационной работы является применение методов анализа данных для исследования структурированности и пространственной динамики сообщества зообентоса рек бассейна реки Енисея. Также в задачу работы входит упрощение анализа данных путем автоматизации процесса получения результатов методов анализа, используя скрипты на языке python и готовые реализации статистических методов.

Целью является обоснование и дополнение имеющихся сведений о количественном распространении зообентоса по реке Кан.

Анализ данных включает в себя:

- Преобразование данных;
- Использование описательных статистик;
- Использование проверки статистических гипотез;
- Корреляционный анализ;
- Регрессионный анализ;
- Кластерный анализ.

Первоначально имеем данные в xlsx формате семейств зообентоса. Из них данные просуммировали до пяти крупных таксономических групп (поденки, веснянки, ручейники, двукрылые, прочие).

Далее эти данные преобразуются в csv (comma-separated values – значения, разделенные запятыми). И над этими данными выполняются следующие преобразования:

- Логарифмирование данных;
- Получение долей зообентоса.

Для первоначальных данных используются описательные статистики для наглядного представления о содержании зообентоса на станции и реке в целом. Также используются гистограммы относительных частот для графического представления возможного распределения. Учитывая возможное распределение происходит проверка статистических гипотез.

Для определения силы взаимосвязи между биотическими факторами (количество и биомасса зообентоса) используется корреляционный анализ.

Регрессионный анализ используется для определения силы влияния абиотических факторов (температура и содержание кислорода) на биотические факторы (количество и биомасса зообентоса).

Кластерный анализ применяется для нахождения пространственной организации по реке и нахождению иных структур в данных. В качестве методов использовались алгоритм k-means++ и иерархическая агломеративная

кластеризация. Оценкой качества был взят критерий силуэт (агломеративная кластеризаци) и сумма внутрекластерных расстояний (k-means++ кластеризация).

Для визуализации процесса кластеризации используются дендрограммы (агломеративная кластеризация) и представление попадания в кластеры элементов выборки для разного количества кластеров (агломеративная и k-means ++ кластеризация).

Результаты анализа представляются в табличной форме и в графическом представлении с соответсвующей раскраской.

Процессс получения результатов методов автоматизирован и выполняется для зообентоса рек бассейна реки Енисея.

1 Используемые методы и критерии

1.1 Статистические методы

Представленные статистические методы и критерии были взяты из книг Крупкиной Т.В [1, 2] и Кобзаря А.И. [3] и Шапиро С.С. [7].

1.1.1 Описательные статистики

Пусть $X = \{x_1, \dots, x_n\}, Y = \{y_1, \dots, y_n\}$ выборки случайной величины объемом n.

Определение 1.1.1. Выборочным средним назовём величину

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i \,. \tag{1.1.1}$$

Здесь и далее среднее значение выборки будем обозначать как $\overline{X}$.

Определение 1.1.2. Выборочной ковариацией между X и Y называется величина

$$K_{XY} = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{X}\right)\left(y_i - \overline{Y}\right) = \overline{XY} - \overline{X}\,\overline{Y}, \tag{1.1.2}$$

где $\overline{XY} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i$.

Определение 1.1.3. Выборочной дисперсией называется величина

$$S_X^2 = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{X}\right)^2 . \tag{1.1.3}$$

Определение 1.1.4. Выборочным среднеквадратичным отклонением называется величина

$$S_x = \sqrt{S_X^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{X}\right)^2}. \tag{1.1.4}$$

Определение 1.1.5. Поставим в соответствие выборке X упорядоченную последовательность

$$x_1^* \le x_2^* \le \dots \le x_n^* , \tag{1.1.5}$$

где $x_1^* = \min\left(x_1, \dots, x_n\right)$;

x_i^* - i-ое по величине значение из $\{x_1, \dots, x_n\}$;

$x_1^* = \max\left(x_1, \dots, x_n\right)$.

Последовательность (1.1.5) назовём вариационным рядом.

Определение 1.1.6. Выборочной квантилью порядка q, $0 < q < 1$ называется значение x_q, равное члену вариационного ряда (1.1.5) с номером $[nq] + 1$, где $[nq]$ – целая часть числа nq.

Часто используемые квантили:

- 0.25-квантиль называется первым (нижним) квартилем;
- 0.5-квантиль называется медианой (вторым) квартилем;
- 0.75-квантиль называется третьим (верхним) квартилем.

1.1.2 Линейный коэффициент корреляции

Для определения линейного коэффициента корреляции воспользуемся введёнными понятиями: выборочная ковариация (1.1.2), выборочное среднеквадратичное отклонение (1.1.4).

Определение 1.2.1. Линейным коэффициентом корреляции называется величина

$$r_{XY} = \frac{K_{XY}}{S_x S_Y}, \quad r_{XY} \in [-1,1]. \tag{1.2.1}$$

Данный коэффициент описывает силу линейной корреляции между случайными величинами X и Y. Чем ближе r_{XY} по модулю к 1, тем сильнее линейная зависимость. И соответственно, чем ближе к нулю, тем слабее эта связь.

Будем делить на слабую, среднюю, высокую и очень высокую. Данное разделение представлено в таблице 1.

Таблица 1 – Определение силы корреляционной

Значение $\lvert r_{XY}\rvert$	Связь
$0 \le \lvert r_{XY}\rvert \le 0.25$	Слабая
$0.25 < \lvert r_{XY}\rvert \le 0.5$	Средняя
$0.5 < \lvert r_{XY}\rvert \le 0.75$	Высокая
$0.75 < \lvert r_{XY}\rvert \le 1$	Очень высокая

Если коэффициент корреляции с отрицательным знаком, то будем называть эту связь отрицательная, в противном случае положительная.

1.1.3 Ранговый коэффициент корреляции Спирмена

Предположим, что имеется n пар наблюдений из непрерывного распределения. Проранжируем X и Y отдельно от меньшего к большему (если есть одинаковые наблюдения, то им присваивается средний ранг) и получим новые пары $(u_i, v_i), i = 1, \dots, n$. Ранговый коэффициент корреляции r_S является

мерой корреляции между рангами, рассчитанной с использованием рангов вместо фактических наблюдений.

Коэффициент корреляции Спирмена вычисляется по следующей формуле:

$$r_s = \frac{n\sum_{i=1}^{n} u_i v_i - \left(\sum_{i=1}^{n} u_i\right)\left(\sum_{i=1}^{n} v_i\right)}{\sqrt{\left[n\sum_{i=1}^{n} u_i^2 - \left(\sum_{i=1}^{n} u_i\right)^2\right]\left[n\sum_{i=1}^{n} v_i^2 - \left(\sum_{i=1}^{n} v_i\right)^2\right]}} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}, \quad (1.3.1)$$

где $d_i = u_i - v_i$.

Проверка гипотезы о значимости коэффициента корреляции Спирмена включает в себя:

- Нулевая гипотеза H_0: $\rho_s = 0$ (корреляции между рангами);
- Альтернативная гипотеза H_α: $\rho_s > 0, \quad \rho_s < 0, \quad \rho_s \neq 0$;
- Статистика: $r_s \sim N\left(0, \frac{1}{\sqrt{n-1}}\right)$;
- Область отклонения гипотезы: $r_s \geq r_{s,\alpha}, \quad r_s < -r_{s,\alpha}, \quad |r_s| \geq r_{s,\alpha/2}$;

где $r_{s,\alpha}$ - критическое значение для критерия коэффициента ранговой корреляции Спирмена;

α - уровень значимости.

1.1.4 Метод наименьших квадратов

Пусть Y – случайная величина, $X_i \ \ i = \overline{1,k}$ – контролируемые (неслучайные) переменные. При этом значения величины Y зависят не только от значений X_i но и от других факторов, которые, возможно, и не поддаются контролю.

Модель (функциональная зависимость) известна из предварительных соображений с точностью до параметров:

$$Y = f\left(X_1,...,X_k,a_1,...,a_s\right) + \varepsilon, \quad (1.4.1)$$

где $a_i \ i = \overline{1,s}$ –параметры модели;

ε – вектор ошибок.

Набор данных имеет вид

$$\begin{aligned} Y_1 &= f\left(X_{11},...,X_{1k},a_1,...,a_s\right) + \varepsilon_1, \\ &... \\ Y_n &= f\left(X_{n1},...,X_{nk},a_1,...,a_s\right) + \varepsilon_n, \end{aligned} \quad (1.4.2)$$

где X_{ij}- значение j-й переменной при i-ом измерении.

Будем считать, что $E\varepsilon = 0$ и ошибки некоррелированные: $K_\varepsilon = E(\varepsilon\varepsilon^T) = \sigma^2 E_n$, где E_n- единичная матрица размером $n \times n$.

Метод наименьших квадратов оценивает параметры модели так, чтобы минимизировать сумму квадратов ошибок, то есть

$$R = \sum_{i=1}^{n} \left(Y_i - f\left(X_{i1}, ..., X_{ik}, a_1, ..., a_s \right) \right)^2, \tag{1.4.2}$$

$$\frac{\partial R}{\partial a_j} = 0 \quad j = \overline{1, s}. \tag{1.4.3}$$

Полученную линию $Y = f(X_1, ..., X_k, a_1, ..., a_s) + \varepsilon$ будем называть линией регрессии Y по X.

1.1.5 Линейная регрессия

Будем рассматривать линейную по параметрам модель вида

$$Y = a_0 + a_1 X_1 + a_2 X_2 + ... + a_k X_k + \varepsilon. \tag{1.5.1}$$

Данную модель будем называть моделью линейной регрессии или линейной регрессией.

В общем случае исходные данные имеют вид:

$$\begin{aligned} Y_1 &= a_0 + X_{11} a_1 + ... + X_{1k} a_k + \varepsilon_1, \\ &... \\ Y_n &= a_0 + X_{n1} a_1 + ... + X_{nk} a_k + \varepsilon_n. \end{aligned} \tag{1.5.2}$$

Или в векторном виде

$$Y = X \cdot a + \varepsilon, \tag{1.5.3}$$

где

$$Y = \begin{pmatrix} Y_1 \\ \cdots \\ Y_n \end{pmatrix}, \quad X = \begin{pmatrix} X_{11} & \cdots & X_{1k} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nk} \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \cdots \\ \varepsilon_n \end{pmatrix}. \tag{1.5.4}$$

Как и в методе наименьших квадратов будем предполагать $E(\varepsilon) = 0, \quad E(\varepsilon\varepsilon^T) = \sigma^2 E_n$, где E_n- единичная матрица размером nxn.

Воспользуемся методом наименьших квадратов для линейной регрессии. Получим следующее:

$$R = \sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{k} X_{ij}a_j - a_0\right)^2 \to \min, \tag{1.5.5}$$

$$\frac{\partial R}{\partial a_l} = -2\sum_{i=1}^{n} X_{il}\left(Y_i - \sum_{j=1}^{k} X_{ij}a_j - a_0\right) = 0, \quad l = \overline{1,k}. \tag{1.5.6}$$

$$\frac{\partial R}{\partial a_0} = -2\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{k} X_{ij}a_j - a_0\right) = 0 \tag{1.5.7}$$

Из поставленной задачи (1.5.5), (1.5.6) и (1.5.7) находятся коэффициенты a_j.

1.1.6 Выборочный коэффициент детерминации

Пусть $Y = (Y_1, \dots, Y_n)$ – случайная величина, $X = (X_1, \dots, X_n)$ – контролируемые (неслучайные) переменные и имеем модель линейной регрессии $\hat{Y} = f(X)$ и $\hat{Y}_i = f(X_i)\ i = \overline{1,n}$.

Определение 1.6.1. Выборочный коэффициент детерминации модели зависимости случайной величины Y от признаков X назовём величину

$$R^2 = 1 - \frac{S_Y^2}{S_Y^2} = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - Y_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \overline{Y_i}\right)^2}, \quad R^2 \in [0,1]. \tag{1.6.1}$$

Проверка значимости коэффицинта детерминации при линейной регрессии с нормально распределенными остатками включает в себя:

- Нулевая гипотеза: $r^2 = 0$;
- Альтернативная гипотеза: $r^2 \neq 0$;
- Ститистика: $F = \frac{R^2/(k-1)}{(1-R^2)/(n-k)} \sim F(k-1, n-k)$;
- Область отклонения гипотезы: $|F| \geq F_{\alpha/2}$;

где $F_{\alpha/2}$- критическое значение коэффициента детерминации,

n – размер выборки;

k – количетсво факторов.

Нормальность регрессионных остатков проверяется с помощью критерия Шапиро-Уилка (1.8.1).

Данный коэффициент принимает значения из интервала $[0,1]$, чем ближе значение к 1, тем ближе модель к зависимости случайной величины Y от не случайных переменных X и соответственно, чем ближе к нулю, тем слабее зависимость случайной величины Y от неслучайных переменных X.

1.1.7 Скорректированный коэффициент детерминации

Для того, чтобы была возможность сравнивать модели с разным числом факторов так, чтобы число регрессоров не влияло на статистику R^2 используется скорректированный коэффициент детерминации, в котором используются несмещенные оценки дисперсий.

Пусть $Y = (Y_1, \dots, Y_n)$ – случайная величина, $X = (X_1, \dots, X_n)$ – контролируемые (неслучайные) переменные, имеем модель линейной регрессии $\hat{Y} = f(X)$, $\hat{Y}_i = f(X_i)\ i = \overline{1,n}$ и R^2 определённый раннее выборочный коэффициент детерминации ().

Определение 1.7.1 Скорректированным коэффициентом детерминации назовём величину

$$R^2_{adj} = 1 - \left(1 - R^2\right)\frac{(n-1)}{(n-k)} \le R^2, \quad R^2_{adj} \in [0,1], \tag{1.7.1}$$

где n – количество наблюдений;

k – количество факторов;

R^2 – ранее введённый коэффициент детерминации (1.6.1).

Скорректированный коэффициент детерминации штрафует за дополнительно включенные факторы. Так как при увеличении числа факторов коэффициент детерминации R^2 будет увеличиваться за счёт дополнительных факторов и скорректированный коэффициент детерминации уменьшает влияние увеличения числа факторов, но не уменьшает показатель силы влияния факторов.

1.1.8 Критерий Шапиро-Уилка

Пусть имеем выборку $X = \{X_1, \dots, X_n\}$ предположительно распределённую из нормального распределения и имеем порядковую статистику $\hat{X} = \{\hat{X}_1, \dots, \hat{X}_n\}$, полученный из данной выборки.

Проверка нормальности распределения проверяется следующим образом:

- Нулевая гипотеза H_0: выборка X нормально распределена;
- Альтернативная гипотеза H_0: выборка X не нормально распределена;
- Статистика:

$$W = \frac{\left(\sum_{i=1}^{n} a_i X_i\right)^2}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}, \tag{1.8.1}$$

где $a = (a_1, \dots, a_n) = \frac{\mathrm{m}V^{-1}}{\sqrt{\mathrm{m}V^{-1}V^{-1}m^T}}$;

$m = (m_1, \dots, m_n)$ – вектор значений нормального распределения, элементы которого упорядочены по не убыванию;

V – соответствующая ковариационная матрица, соответствующая вектору m.

Пусть x_i, $i=1,\cdots,n$ - выборка из нормального распределения со средним значением 0 и дисперсией равным 1, тогда вектор m и матрица V обладает следующими свойствами:

- $E(x_i)=m_i$, $i=1,\cdots,n$;
- $Cov(x_i,x_j)=v_{i,j}$, $i,j=1,\cdots,n$.

Если значение полученной статистики меньше W критического значения статистики $W(\alpha)$ (α – уровень значимости статистики), то гипотеза о нормальности распределений H_0 отвергается.

1.2 Кластеризация

1.2.1 Агломеративная кластеризация

Агломеративная (восходящая) кластеризация – метод кластеризации, основанный на объединении объектов в более крупные кластеры, на основании заданного расстояния между кластерами [4].

Первоначально для данного алгоритма кластеризации нужно задать метрику $\rho(x,x')$ – расстояние между элементами выборки. Будем использовать евклидову метрику:

$$\rho(x,x')=\sqrt{\sum_{i=1}^{k}(x_i-x_i')^2} \qquad (2.1.1)$$

Теперь выберем метод, по которому будут объединяться два кластерам между собой. Будем использовать метод Уорда с межкластерным расстояние равным:

$$R(W,S)=\frac{|S||W|}{|S|+|W|}\rho^2\left(\sum_{w\in W}\frac{w}{|W|},\sum_{s\in S}\frac{s}{|S|}\right), \qquad (2.1.2)$$

где $|S|,|W|$ – мощности множеств S и W.

Алгоритм агломеративной кластеризации:

а) Инициализировать множество кластеров $C_1=\{\{c_1\},\{c_2\},\cdots,\{c_n\}\}$ (каждый элемент выборки является кластером);

б) Для всех $t=2,\ldots,n$:

1) Найти в C_{t-1} два ближайших кластера:
$(U,V)=\arg\min_{U\neq V}R(U,V)$;
$R_t=(U,V)$;

2) Убрать кластеры U и V, добавить новый кластер $W=U\cup V$:
$C_t=C_{t-1}\cup\{W\}\setminus\{U,V\}$;

3) Для всех S $\in C_t$:
Вычислить расстояние $R(W,S)$.

Сложность алгоритма (n – размер выборки):

- по времени: $O(n^3)$;
- по памяти: $O(n^2)$.

1.2.2 K-means кластеризация

Алгоритм k-means кластеризации:

а) Произвольный образом выбрать k центров кластеров $C = \{c_1, \ldots, c_k\}$;

б) Для каждого $i \in \{1, \ldots, k\}$ установить кластер C_i такой, что все точки $x \in C_i \subset X$ ближе к центру кластера c_i, чем до других центров кластеров c_j, $j \neq i$;

в) Для каждого $i \in \{1, \ldots, k\}$ установить новый центр кластера C_i:

$$c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x;$$

г) Повторять шаги 2 и 3 пока множество центроид кластеров C не перестанет меняться;

Сложность алгоритма по времени:

- Худшее время: $O(n^{k+\frac{2}{p}})$;
- Среднее время: $O(nki)$;
- По памяти: $O(n^2)$;

где n – размер выборки;
k – количество кластеров;
p – количество признаков (факторов);
i – количество итераций.

1.2.3 K-means++ кластеризация

Поскольку в алгоритме k-means на этапе 1 не говориться о том, как должны выбираться первоначальные приближения центров кластеров, то улучшенный алгоритм k-means++ пользуется этим для увеличения скорости сходимости алгоритма, предложенный Д. Артуром и Васильвитским. С [5] .

Алгоритм k-means++ кластеризации:

а) взять центр c_1 случайным образом из X;

б) выбрать новый центр c_i, выбирая x $\in X$ с вероятностью $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$;

в) повторять шаг b пока не будем иметь k центроид;

г) далее выполняется алгоритм кластеризации k-means.

$D(x)$ – представляет собой функцию, возвращающую самое короткое расстояние от уже выбранных центроид до точки $x \in X$.

1.3 Критерии кластеризации

1.3.1 Сумма квадратов расстояний от точек до центроидов кластеров

Данный критерий используется для определения количества кластеров. Этот метод является эвристическим.

Суммой квадратов расстояний от точек до центроидов кластеров является величина

$$I(C)=\sum_{k=1}^{K}\sum_{x_i\in C_k}\|x_i-\mu_k\|^2, \qquad (3.1.1)$$

где $\mathrm{C}=\{C_1,\dots,C_K\}$ – полученные множества кластеров мощности K;

$\mu_k=\frac{1}{|C_k|}\sum_{x_i\in C_k}x_i,\quad k=\overline{1,K}$ - центроиды кластеров.

Эвристика данного критерия такова: выбирается число кластеров K такое, начиная с которого, функционал $I(C)$ падает “не так уж быстро” или более формально:

$$D(k)=\frac{|I(C_k)-I(C_{k+1})|}{|I(C_{k-1})-I(C_k)|}\to\min_k, \qquad (3.1.2)$$

где C_K - множество кластеров мощности K, полученные в процессе кластеризации.

1.3.2 Критерий силуэт

Данный критерий предложен П.Дж. Роусью [6] в 2007 году

Пусть в процессе кластеризации получено множество кластеров $C=\{C_1,\dots,C_K\}$. Кластеры имеют мощности $|C_1|=m_1,\dots,|C_K|=m_k$ соответственно.

Силуэтом объекта называется величина

$$s_j^i=\frac{b_j^i-a_j^i}{\max(b_j^i,a_j^i)}\quad i=\overline{1,K},\quad j=\overline{1,m_i},\quad s_j^i\in[-1,1], \qquad (3.2.1)$$

где $a_j^i=\frac{1}{m_i-1}\sum_{\substack{d=1\\d\neq j}}^{m_i}|x_d^i-x_j^i|$ - среднее расстояние от данного объекта до объектов из того же кластера (среднее внутри кластерное расстояние);

$b_j^i=\min_{C_k\neq C_i}\frac{1}{m_k}\sum_{d=1}^{m_k}|x_d^k-x_j^i|$ - среднее расстояние от данного объекта до объектов из ближайшего кластера (отличного от того, в котором лежит сам объект).

Силуэтом называется средняя величина силуэта объектов данной выборки:

$$s = \frac{1}{m_1 + \cdots + m_K} \sum_{i=1}^{K} \sum_{j=1}^{m_i} s_j^i, \quad s \in [-1,1], \tag{3.2.2}$$

Чем ближе силуэт выборки к -1, тем кластеры менее плотные. Если значение силуэта ближе к нулю, то, скорее всего, отсутствуют кластеры или они наложены друг на друга. Если значение ближе к 1, то кластеры более плотные (отделены друг от друга).

1.4 Преобразование данных

1.4.1 Стандартизация данных

Стандартизация используется для приведения данных к безразмерным величинам. Это полезно тем, что мы отходим от размерностей данных и можем сравнивать обезразмеренные данные. Стандартизация данных представляет из себя преобразования данных типа.

Определение 4.1.1. Стандартизацией данных назовём преобразование вида:

$$X_i^{'} = \frac{X_i - \overline{X}}{S_x}, \; i = 1, \cdots, n. \tag{4.1.1}$$

Стандартизированные данные имеют следующие свойства:

а) $\overline{X_i} = 0$;

б) $S_{X_i} = 1$.

1.4.2 Логарифмирование данных

Определение 4.2. Логарифмированием данных назовём преобразование вида:

$$X_i^{'} = \ln(X_i + 1), \quad i = 1, \cdots, n. \tag{4.2.1}$$

Под X_i может пониматься и вектор значений. Тогда предобразование для одного элемента будет иметь вид:

$$X^{'} = (x_1^{'}, x_2^{'}, \cdots, x_n^{'}) = \ln(X) = (\ln(x_1 + 1), \ln(x_2 + 1), \cdots, \ln(x_n + 1)). \tag{4.2.2}$$

1.5 Используемые программные пакеты

Используемый язык программирования: python.

В библиотеке NumPy используется готовая реализация многомерных массивов и математические функции, оптимизированные для многомерных массивов. Библиотека pandas как надстройка над библиотекой NumPy используется для чтения и сохранения данных в csv формате, для получения описательных статистик и для удобного обращения с данными.

Библиотека SciPy используется для проверка статистических гипотез и получения дендрограммы в агломеративной кластеризации. В библиотеке scikit-learn (sklearn) содержатся алгоритмы k-means и агломеративной кластеризаций, а также реализация линейной регрессии и метрики проверки качества кластеризации.

Остальные библиотеки matplotlib и xlsxwriter используются для удобного графического представления результатов в виде графиков и таблиц.

Версии используемых библиотек и языка программирования:

- Python 3.7.2;
- xlsxwriter 1.1.5;
- Matplotlib 3.0.2;
- NumPy 1.16.1;
- Sklearn 0.20.2;
- SciPy 1.2.1;
- Pandas 0.24.1.

Для автоматизации процесса анализа рек были созданы следующие методы и классы, упрощающие получения результатов анализа:

- класс для создания файлов, куда сохраняются результаты анализа;
- скрипты, получающие описательные статистики для первоначальных данных с дальнейшим сохранением результатов в графическом (гистограммы относительных частот) и табличном видах в xlsx формате;
- методы, вычисляющие коэффициенты корреляции Спирмена для всех численных атрибутов, входящих в данные, с проверкой на достоверность коэффициентов корреляции с уровнем значимости равным 0,05. Также эти методы создают и сохраняют значимые коэффициенты корреляции в виде корреляционного графа с заданным минимальным пороговым значением. Кроме того, методы сохраняют результаты корреляционного анализа в графическом (корреляционные графы) и табличном (xlsx файлы) видах, причем таблицы раскрашиваются в соответствии со значениями достоверности коэффициентов корреляции;
- методы, выполняющие линейную регрессию на каждом биотическом факторе с абиотическими факторами по отдельности и вычисляющие коэффициент детерминации с проверкой на достоверности при уровне значимости равным 0,05. Далее методы сохраняют результаты линейной регрессии в виде графиков самой линейной регрессии и в табличном виде с раскраской по значимости линейной регрессии от определенного абиотического фактора;
- методы, выполняющие преобразование данных и агломеративную кластеризацию по методу Варда над преобразованными данными с сохранением результатов в виде дендрограмм и таблиц в xlsx формате с раскраской по принадлежности к каждому кластеру;
- класс, объединяющий все вышеперечисленные методы и класс технологии анализа.

Код программ представлен в приложении В.

2 Анализ зообентоса рек бассейна реки Енисея

2.1 Методика анализа данных

Процесс анализа данных состоит из шести этапов:

- Получение данных в xlsx формате и преобразование их в csv формат;
- Преобразование данных;
- Использование описательных статистик и проверка статистических гипотез;
- Корреляционный анализ;
- Регрессионный анализ;
- Кластерный анализ.

Первый этап заключается в получении данных в xlsx формате и переводе их в csv (comma-separated values) формат.

В xlsx файле имеются данные по семействам зообентоса реки бассейна Енисея. Всего представлено 39 семейств: Ephemeridae, Potamanthidae, Ephemerellidae, Heptageniidae, Baetidae, Leptophlebiidae, Caenidae, Taeniopterycidae, Perlodidae, Chloroperlidae, Perlidae, Pteronarcyidae, Arctopsychidae, Hydropsychidae, Stenopsychidae, Psychomyiidae, Polycentropodidae, Limnephilidae, Brachycentridae, Goeridae, Rhyacophilidae, Lepidostomatidae, Sericostomatidae, Glossosomatidae, Leptoceridae, Ceratopogonidae, Limoniidae, Tipulidae, Athericidae, Tanypodinae, Orthocladiinae, Chironomini, Tanytarsini, Lymnaeidae, Valvatidae, Planorbidae, Bivalvia, Lumbriculidae, Tubificidae.

Семейства описываются двумя биотическими факторами: количество зообентоса и их биомасса.

Каждая проба имеет уникальный идентификатор – номер пробы, имеющий формат <номер_станции>.<номер_пробы_на_станции>. Проба содержит название станции, описание грунта, температуру, содержание кислорода в месте взятии пробы, количество особей и биомасса каждого семейства зообентоса.

Для некоторых видов анализа биотические факторы суммируются до пяти крупных таксономических групп: поденки, веснянки, ручейники, двукрылые и прочие. Таким образом, анализ проводится как на первоначальных данных, так и на сгруппированных по принадлежности к таксономической группе.

Первый этап завершается добавлением столбца описания местности в месте взятии пробы. Включает в себя объединение данных описаний грунта и названия станции. Далее данные сохраняются в csv формате.

Второй этап заключается в преобразовании первоначальных данных по пробам с помощью различных методов: стандартизации, логарифмирования данных и получение долей каждой таксономической группы зообентоса в общем количестве или биомассе зообентоса в пробе.

Стандартизация используется с целью обезразмеривания данных. Однако, от анализа стандартизированных данных отказались, поскольку в анализе всегда присутствуют однородные по смыслу данные (только численность, или только

биомасса) и она не даёт новых результатов по сравнению с использованием первоначальных данных.

Логарифмирование данных используется с целью приближения распределения данных к нормальному.

В дальнейшем над первоначальными и логарифмированными данными будут использоваться описательные статистики и статистики для проверки гипотез, корреляционный, регрессионный и кластерный анализы. Следует отметить, что подсчет доли показателя (численность, биомасса) каждой таксономической группы по отношению к суммарному показателю в пробе вносит зависимость данных внутри строки. Поэтому оправданным будет использование этих данных только для кластерного анализа с целью выявления доминантных таксономических групп в пробе.

Описательные статистики применяются в анализе для представления структурированности данных. Они в себя включают:

- Определения главных представителей зообентоса в пробе, станции и реке;
- Определение возможных выбросов в данных;
- Определение возможного распределения с помощью гистограммы относительных частот;
- Использование критерия Шапиро-Уилка для определения нормальности распределения зообентоса.

С учётом возможного распределения проверятся те или иные статистики для проверки гипотез.

Корреляционный анализ включает в себя использование линейного коэффициента корреляции для нахождения силы статистической зависимости биотических факторов (численность и биомасса). Однако, из-за ненормальности распределения зообентоса или сильной асимметричности используется ранговый коэффициент корреляции Спирмена. Значимость полученного коэффициента корреляции проверяется с помощью статистики с уровнем значимости равному 0.05.

Регрессионный анализ используется для определения силы влияния абиотических факторов (температура и содержание кислорода) на биотические факторы (численность и биомасса зообентоса). Для этого строится линейная регрессия с помощью метода наименьших квадратов. Для полученной линии регрессии рассчитываются коэффициент детерминации, как критерий силы влияния, и выполняется проверка статистической значимости коэфиициента детерминации с помощью статистики с уровнем значимости равным 0.05.

Для определения структурной организации и пространственной динамики используются методы кластеризации k-means++ и агломеративная кластеризация. Качеством кластеризации выступает критерий силуэт (k-means++ и агломертивная кластеризация) и сумма внутрекластерных расстояний (k-means++ кластеризация).

Основным для k-mean++ критерием была сумма внутекластерных расстояний. Однако на практике график сумм внутрекластерных расстояний не

имеет выраженного падения при определенном количестве кластеров, что может быть следствием отсутствия выраженных кластеров, зашумленности данных или недостаточного количества данных. Поэтому от k-means++ кластеризации отказались по причине трудности определения количества кластеров.

Агломеративная кластеризация удобна в анализе тем, что виден процесс получения кластеров, который представляется в виде дендрограммы. Помимо критерия силуэта используется подход определения количества путем нахождения такого количества кластеров k, которому будет соответствовать большое межкластерное расстояние.

2.2 Анализ данных на примере реки Кан

2.2.1 Общий обзор данных

Первоначально данные по семейства представляют собой таблицу 27 строк на 43. Строки представляют собой набор признаков отдельной пробы грунта, взятой на выделенных биологами станциях р. Кан. Биотические признаки пересчитаны на $м^2$ площади пробы. В столбцах записаны данные проб. Название столбцов и их описание представлены в таблице. Краткое описание атрибутов данных:

- Название станции – представляется строкой;
- Грунт в месте взятии пробы – представляется строкой;
- Номер пробы – натуральное число;
- Температура (температура воды в месте взятии пробы) – цельсии (°C);
- Содержание кислорода (содержание кислорода в месте взятии пробы) – мг/л;
- Количество особей зообентоса определенного семейства – штуки.

Для анализа данных по семействам воспользуемся количеством зообентоса. Названием станции и грунтом в месте взятии пробы воспользуемся при получения результатов – эти данные будут запиываться для проб зообентоса для удобного определения возможного разделения на кластеры.

Биотические данные, просуммированные до пяти крупных таксономических групп, представляют из себя такую же таблицу, но с меньшим числом признаков. 39 столбцов семейств были сокращены до 5 столбцов крупных таксономических групп с тем же измерением (в штуках) количества особей зообентоса.

Следует отметить, что целеполагание анализа при использовании данных по семействам и по крупным таксономическим группам разное. Данные по семействам сильно чувствительны при детальном исследовании изменения структуры зообентоса вдоль реки – например, семейства характерные для верховья практически отсутствуют в низовье. Использование крупных таксономических групп сглаживает эту изменчивость, на первый план выдвигаются другие особенности пробы, связанные с природными особенностями зоны и его экологическими и антропогенными характеристиками.

2.2.2 Описательные статистики

Для данных по семействам описательные статистики не используются из-за большой разреженности данных численного количества зообентоса. Однако, у данных просуммированных до пяти крупных таксономических групп такой проблемы нет. Поэтому воспользуемся этими данными для общего представления о зообентосе на реке Кан.

Описательные статистики представлены в таблице 2.

Таблица 2 – Описательные статистики крупных таксономических групп зообентоса

	T, °C	O2,мг/л	Двукрылые	Поденки	Веснянки	Ручейники	Прочие
Количество	27	27	27	27	27	27	27
Среднее значение	15,44	10,16	273,78	370,96	62,22	366,22	122,07
Стандартное отклонение	1,95	0,75	346,2	241,43	94,9	428,8	172,39
Минимальное значение	12,1	8,93	16	64	0	32	0
25%	14	9,66	80	200	0	96	16
50%	16,5	10,09	160	304	16	240	32
75%	16,6	10,6	304	472	104	488	128
Максимальное значение	18,6	11,5	1440	1072	336	2160	560

Из среднего значения и стандартного отклонения можно сделать вывод что в основном вклад в общую численность зообентоса несут Двукрылые, Поденки и Ручейники. Однако максимальные значения у них относительно большие, а все квантили лежат близко к среднему значению.

Если считать распределение этих величин нормальным, то такая ситуация может быть при аномально больших выбросах в данных, что и служит причиной смещения среднего значения и среднего отклонения в большую сторону.

2.2.3 Распределения данных

Представим данные в виде гистограммы относительных частот. Они будут являться оценкой функции плотности распределения данных и по ним уже можно предположить возможное распределение данных. Распределения будем рассматривать для Двукрылых, Поденок, Веснянок, Ручейников и прочих. Для семейств зообентоса гистограммы относительных частот представлены на рисунках 1-5.

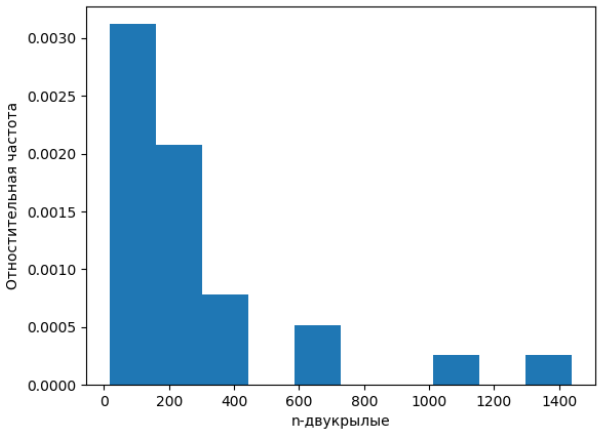

Рисунок 1 – Гистограмма относительных частот для Двукрылых

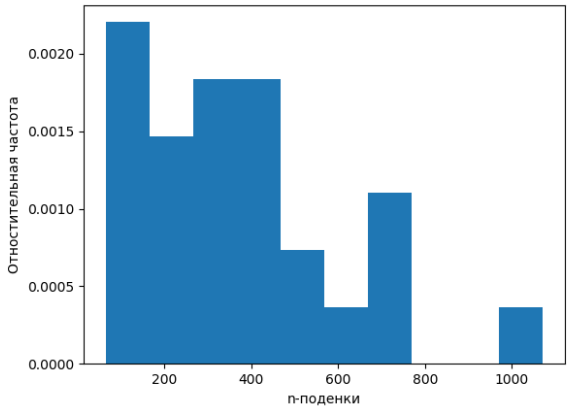

Рисунок 2 – Гистограмма относительных частот для Поденок

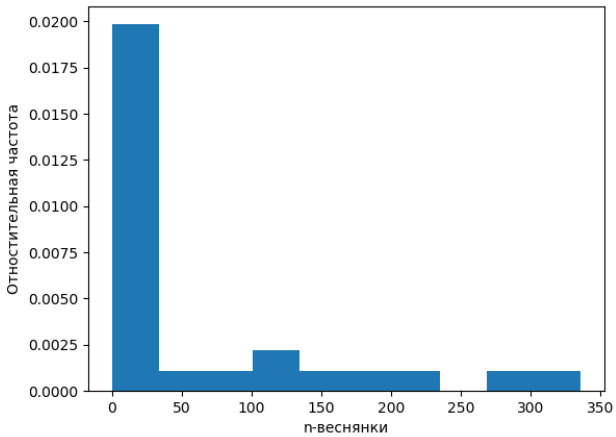

Рисунок 3 – Гистограмма относительных частот для Веснянок

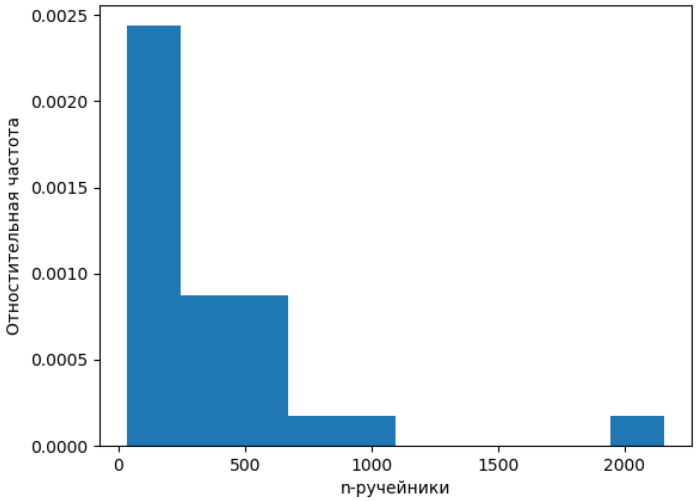

Рисунок 4 – Гистограмма относительных частот для Ручейников

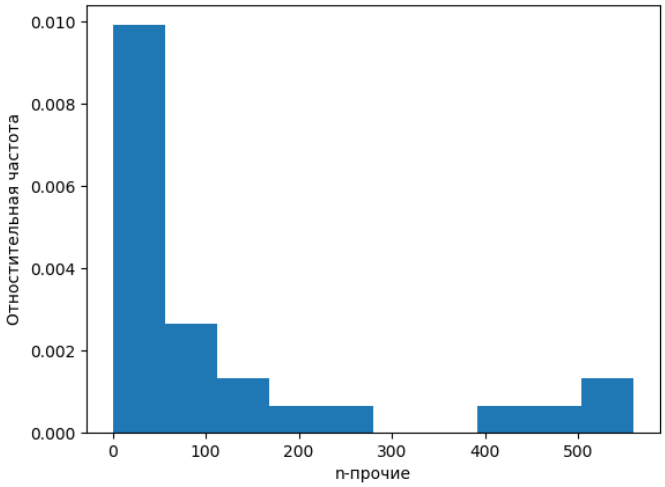

Рисунок 5 – Гистограмма относительных частот для Прочих

Как видно из рис. 1-5, предположение о нормальности распределения данных невозможно. Это следует из явной асимметричности распределения данных и из того, что данные имеют только дискретный характер. Следовательно, некоторые сильные статистические критерии невозможно применить на данных выборках.

Распределения для кислорода и температуры имеют другой вид и представлены на рис. 6-7.

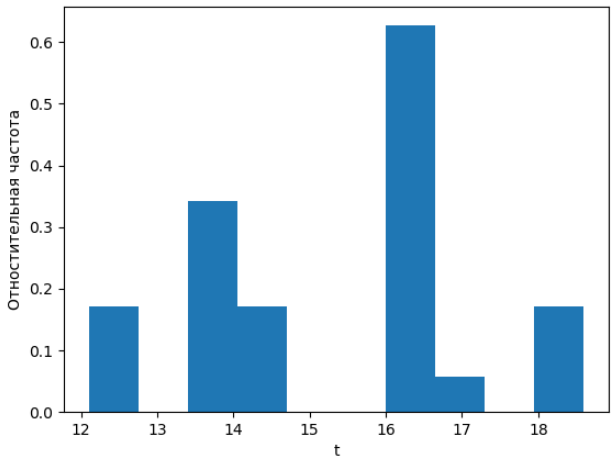

Рисунок 6 – Гистограмма относительных частот для температуры

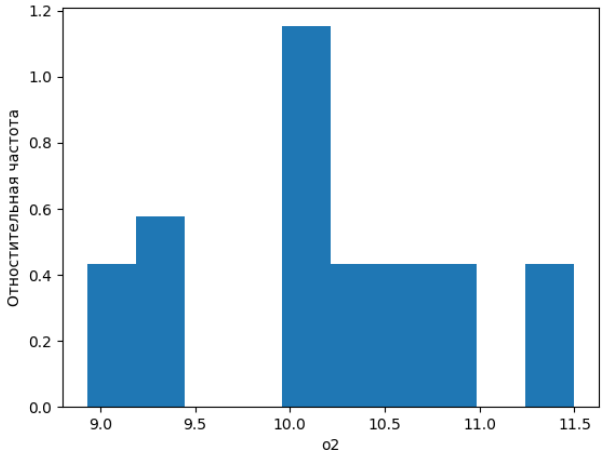

Рисунок 7 – Гистограмма относительных частот для Кислорода

Проверим гипотезу нормальности распределений температуры и кислорода критерием Шапиров-Уилка с уровнем значимости $\alpha = 0.05$. Результаты вычисления значения критерия представлены в табл. 3.

Таблица 3 – Значение критериев Шапиро-Уилка и p-значение для кислорода и температуры

Фактор	Значение критерия W	p-значение
Кислород	0.9433	0.1467
Температура	0.8985	0.0124

Как видно значение критерия W для температуры соответствует p-значению равному 0.0124. Из чего следует что, гипотезу о нормальности распределения температуры можно отклонить.

Для кислорода получаем значение W равным 0.9433, что соответствует p-значению равному 0.1467. Из чего следует, что гипотезу о нормальности распределения кислорода не отклоняем.

2.2.4 Корреляция

Вычислим линейный коэффициент корреляции на численности зообентоса по реке Кан (табл. 4).

Таблица 4 –Таблица линейных коэффициентов корреляции для численности таксономических групп зообентоса

Линейный коэффициент корреляции					
	Двукрылые	Поденки	Веснянки	Ручейники	Прочие
Двукрылые	1	0,203	0,645	0,2575	-0,14
Поденки	0,203	1	0,3574	0,7044	-0,0944
Веснянки	0,645	0,3574	1	0,2165	-0,1903
Ручейники	0,2575	0,7044	0,2165	1	-0,1606
Прочие	-0,14	-0,0944	-0,1903	-0,1606	1

Как можно заметить, присутствует высокая корреляция между Двукрылыми и Веснянками и Поденками и Ручейниками. Также видно среднюю корреляцию между Поденками и Веснянками и Двукрылыми и Ручейниками.

Поскольку распределения зообентоса не нормальны, то мы не можем применить t-критерий Стьюдента для проверки значимости коэффициентов корреляций. Поэтому откажемся от линейного коэффициента корреляции по причине невозможности дальнейшего применения статистик для проверки гипотез.

Вычислим ранговый коэффициент корреляции Спирмена с проверкой на значимость (табл. 5-6).

В данных таблицах представлен фрагмент из xlsx файла. В нём используется раскраска для лучшего определения значимых коэффициентов. Зелёным цветом обозначаются коэффициенты, у которых вычисленное p-значение для статистики меньше 0.05. Если p-значение больше или равно 0,05, то ячейка красится красным цветом.

Таблица 5 – Ранговый коэффициент корреляции для численности таксономических групп зообентоса

Коэффициент корреляции	n-двукрылые	n-поденки	n-веснянки	n-ручейники	n-прочие
n-двукрылые	1	0,333	0,433	0,251	-0,005
n-поденки	0,333	1	0,391	0,654	-0,172
n-веснянки	0,433	0,391	1	0,474	-0,264
n-ручейники	0,251	0,654	0,474	1	-0,366
n-прочие	-0,005	-0,172	-0,264	-0,366	1

Таблица 6 – Р-значение статистик для проверки гипотезы рангового коэффициента корреляции численности зообентоса на значимость

Р-значение	n-двукрылые	n-поденки	n-веснянки	n-ручейники	n-прочие
n-двукрылые	0	0,089	0,024	0,206	0,98
n-поденки	0,089	0	0,044	0	0,39
n-веснянки	0,024	0,044	0	0,012	0,184
n-ручейники	0,206	0	0,012	0	0,06
n-прочие	0,98	0,39	0,184	0,06	0

Можно заметить, что Поденки и Ручейники имеют высокую положительную корреляцию, Двукрылые и Веснянки, Поденки и Веснянки и Веснянки и ручейники имеют среднюю положительную корреляцию.

Также приведём корреляционный граф, для достоверных данных. Граф представлен на рис. 8.

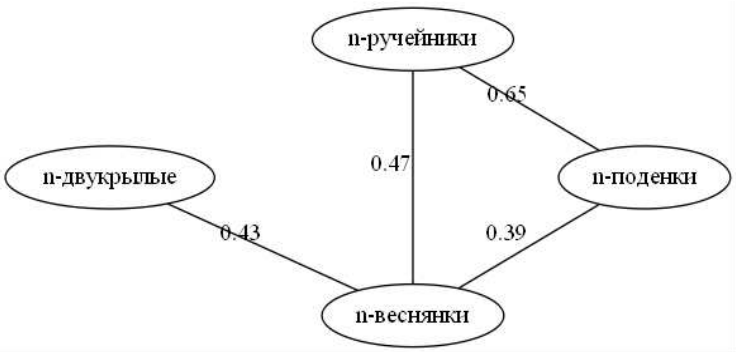

Рисунок 8 – Корреляционный граф со значимыми ранговыми коэффициентами корреляции численности зообентоса

Аналогичный корреляционный анализ проведен для биомассы зообентоса. Полученные результаты представлены в табл. 7-8 и рис. 9.

В данных таблицах можно заметить высокую положительную корреляцию массы Поденок и Прочих и Веснянок и Ручейников и высокую отрицательную корреляцию Поденок и Веснянок.

Таблица 7 – Ранговый коэффициент корреляции для биомасса таксономических групп зообентоса

Коэффициент корреляции	b-двукрылые	b-поденки	b-веснянки	b-ручейники	b-прочие
b-двукрылые	1	-0,229	0,345	0,331	0,242
b-поденки	-0,229	1	-0,568	-0,332	0,504
b-веснянки	0,345	-0,568	1	0,719	-0,095
b-ручейники	0,331	-0,332	0,719	1	-0,134
b-прочие	0,242	0,504	-0,095	-0,134	1

Таблица 8 – P-значение статистик для проверки гипотезы рангового коэффициента корреляции биомассы зообентоса на значимость

P-значение	b-двукрылые	b-поденки	b-веснянки	b-ручейники	b-прочие
b-двукрылые	0	0,25	0,078	0,091	0,225
b-поденки	0,25	0	0,002	0,09	0,007
b-веснянки	0,078	0,002	0	0	0,639
b-ручейники	0,091	0,09	0	0	0,504
b-прочие	0,225	0,007	0,639	0,504	0

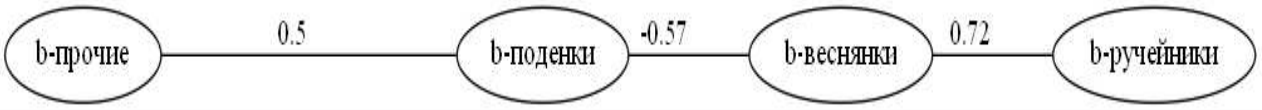

Рисунок 9 – Корреляционный граф со значимыми ранговыми коэффициентами корреляции численности зообентоса

2.2.5 Линейная регрессия

Рассмотрим количество зообентоса от температуры, от содержания кислорода и от температуры и содержания кислорода вместе. В качестве оценки силы линейной зависимости берутся два критерия: коэффициент детерминации R^2 и скорректированный коэффициент детерминации R^2_{adj}. Для проверки значимости коэффициента детерминации применяется статистика Фишера и критерий Шапиро-Уилка для проверки на нормальность остатков регрессии.

Полученные коэффициенты детерминации с проверкой на значимость представлены в табл. 9. Зелёным цветом показаны результаты линейной регрессии, на которых p-значение меньше, чем 0.05, или равно 0,05 для коэффициента детерминации и больше чем 0.05 для критерия Шапиро-Уилка. В ином случае обозначается красным цветом. В дальнейшем используются аналогичные обозначения по цвету для статистик линейной регрессии.

Таблица 9 – Коэффициенты детерминации для линейной регрессии от температуры со статистиками Фишера и Шапиро-Уилка

Температура					
Зообентос	R^2	F value	F p-value	Shapiro	Shapiro p-value
n-двукрылые	0.31	11.21	0.003	0.911	0.329
n-поденки	0.082	2.239	0.147	0.897	0.001
n-веснянки	0.23	7.462	0.011	0.896	0.426
n-ручейники	0.045	1.165	0.291	0.639	0.308
n-прочие	0.267	9.114	0.006	0.875	0.446
b-двукрылые	0.181	5.536	0.027	0.788	0.391
b-поденки	0.006	0.149	0.703	0.748	0.497
b-веснянки	0.116	3.292	0.082	0.669	0.023
b-ручейники	0.065	1.745	0.198	0.818	0.446
b-прочие	0	0.004	0.949	0.671	0.478

Из представленных данных можно заметить, что 31, 23 и 26 процентов дисперсии количества Двукрылых, Веснянок и Прочих и 18 процентов дисперсии биомассы объясняются температурой. Остальные значения относительно малы и не значимы.

Дальше рассмотрим влияние содержания кислорода на количество и биомассу зообентоса. Результаты линейной регрессии представлены в таб. 10.

Таблица 10 – Коэффициент детерминации для линейной регрессии от кислорода со статистиками Фишера и Шапиро-Уилка

Кислород					
Зообентос	R^2	F value	F p-value	Shapiro	Shapiro p-value
n-двукрылые	0.036	0.938	0.342	0.704	0.434
n-поденки	0.123	3.517	0.072	0.925	0.053
n-веснянки	0.022	0.571	0.457	0.784	0.303
n-ручейники	0.221	7.09	0.013	0.816	0.457
n-прочие	0.089	2.43	0.132	0.77	0.478
b-двукрылые	0.007	0.188	0.668	0.578	0.385
b-поденки	0.207	6.516	0.017	0.857	0.456
b-веснянки	0.005	0.125	0.726	0.552	0.416
b-ручейники	0.072	1.928	0.177	0.861	0.385
b-прочие	0.11	3.079	0.092	0.785	0.453

В случае кислорода можно увидеть лишь 22 процента объяснённой дисперсии для численности Ручейников и 20 процентов объяснённой дисперсии. Остальные значения относительно малы и не значимы.

Значения коэффициентов детерминации для соответствующих совместных линейных моделей для температуры и кислорода приведены в табл. 11.

Таблица 11 – Коэффициент детерминации и скорректированный коэффициент детерминации для линейной регрессии от температуры и кислорода со статистикой Фишера и Шапиро-Уилка

Температура + Кислород						
Зообентос	R^2	R^2_{adj}	F value	F p-value	Shapiro	Shapiro p-value
n-двукрылые	0.339	0.283	6.142	0.007	0.929	0.064
n-поденки	0.199	0.132	2.973	0.07	0.937	0.103
n-веснянки	0.247	0.185	3.943	0.033	0.899	0.387
n-ручейники	0.259	0.197	4.187	0.028	0.768	0.37
n-прочие	0.367	0.315	6.966	0.004	0.926	0.056
b-двукрылые	0.186	0.119	2.749	0.084	0.795	0.47
b-поденки	0.215	0.15	3.296	0.054	0.858	0.363
b-веснянки	0.12	0.046	1.633	0.216	0.668	0.348
b-ручейники	0.132	0.06	1.826	0.183	0.849	0.335
b-прочие	0.11	0.035	1.478	0.248	0.785	0.431

В представленных данных можно заметить повышение процента объясненной дисперсии R^2 для линейных моделей по сравнению с предыдущими моделями. Скорректированный R^2_{adj} занижает значение коэффициента для оценки точного влияния при увеличении количества факторов. Но все же они не слишком малы, чтобы их не учитывать.

Представим в виде таблицы изменение коэффициента детерминации при добавлении новых факторов (табл. 12).

Таблица 12 – Таблица коэффициентов детерминации для линейных регрессии от температуры, кислорода, температуры и кислорода вместе

Зообентос	$R^2\ (T)$	$R^2\ (O_2)$	$R^2\ (T + O_2)$	$R^2_{adj}(T + O_2)$
n-двукрылые	0,31	0,036	0,339	0,283
n-поденки	0,082	0,123	0,199	0,132
n-веснянки	0,23	0,022	0,247	0,185
n-ручейники	0,045	0,221	0,259	0,197
n-прочие	0,267	0,089	0,367	0,315
b-двукрылые	0,181	0,007	0,186	0,119
b-поденки	0,006	0,207	0,215	0,15
b-веснянки	0,116	0,005	0,12	0,046
b-ручейники	0,065	0,072	0,132	0,06
b-прочие	0	0,11	0,11	0,035

Для всех исследуемых таксономических групп значение коэффициента детерминации R^2 в модели, использующей совместно оба абиотических фактора (кислород и температуру) увеличивается по сравнению с моделями простейших зависимостей от каждого из абиотических факторов по отдельности. При этом значение скорректированного коэффициента регрессии R^2_{adj} относительно велико у количества Двукрылых и Прочих, что свидетельствует о значимом увеличении достоверности совместной модели.

Количество Поденок, Веснянок и Ручейников также лучше описываются совместным влиянием температуры и кислорода, хотя сила совместного влияния абиотических факторов в этих случаях меньше, чем у Двукрылых и Прочих.

В целом можно сделать вывод, что все исследованные модели все ещё плохо описывают количество зообентоса от внешних факторов. Это может быть следствием нелинейной зависимости между зообентосом и внешними факторами, которые не учтены в анализе.

2.2.6 Агломеративная кластеризация по методу Варда

Агломеративная кластеризация была проведена на трёх видах данных:
- На первоначальных данных;
- На логарифмированных данных;
- На долях зообентоса пробе.

Температура и содержание кислорода в месте взятия пробы не используется в кластеризации. Также кластеризация проводилась не только по численности зообентоса, но и по его биомассе. Далее приводятся результаты кластеризации не только по крупным таксономическим группами, но и по численности семейств с использование таких же преобразований данных.

Рассмотрим результаты кластеризаций. Сперва посмотрим результаты над численностью и биомассой без преобразований данных и над семействами зообентоса. Дендрограммы кластеризации представлены на рис. 10-12.

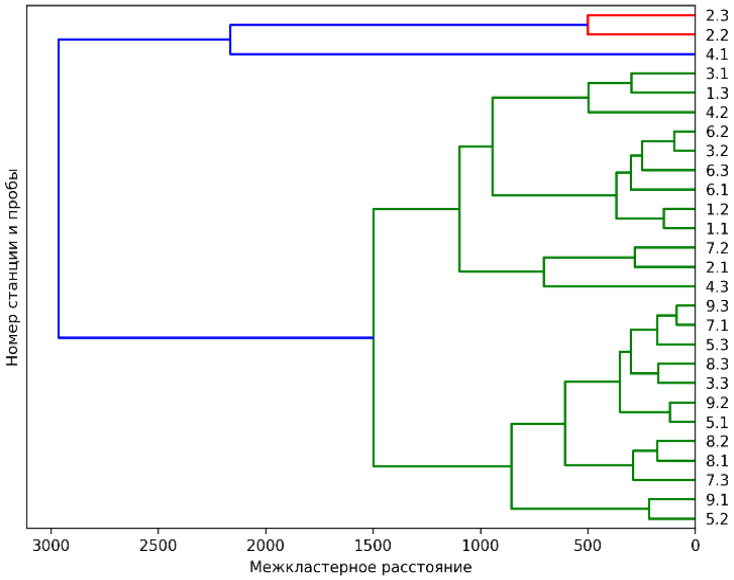

Рисунок 10 – Дендрограма агломеративной кластеризации по методу Варда на численности зообентоса крупных таксономических групп

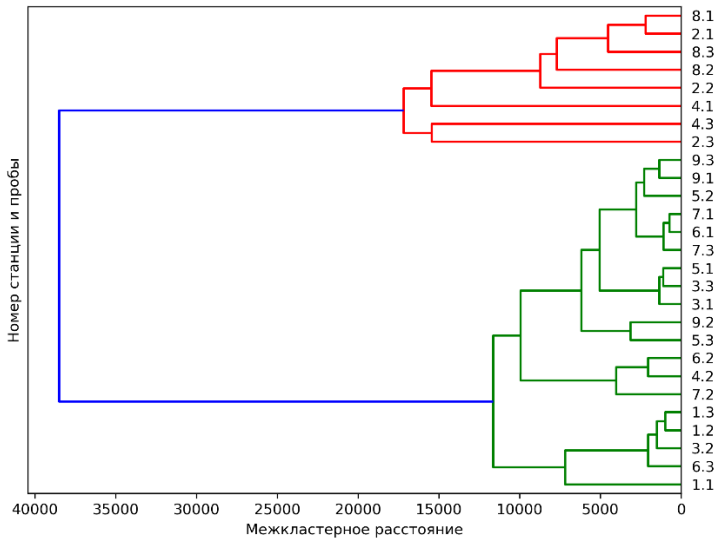

Рисунок 11 – Дендрограма агломеративной кластеризации по методу Варда на биомассы зообентоса крупных таксономических групп

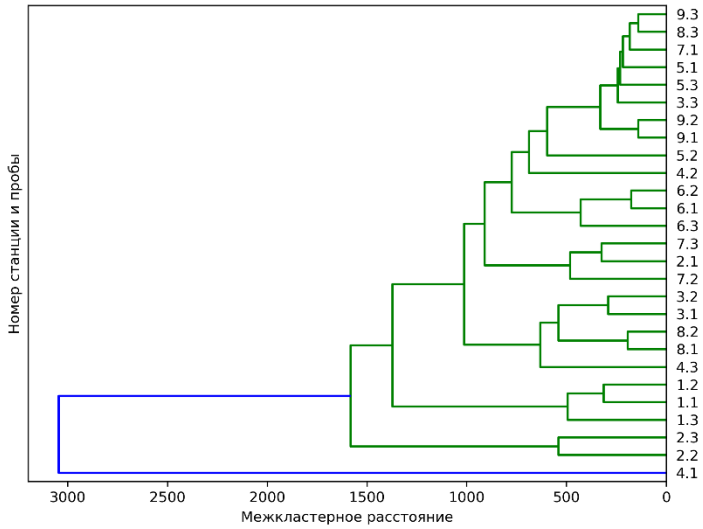

Рисунок 12 – Дендрограма агломеративной кластеризации по методу Варда на численности семейств зообентоса.

Индексы справа представлены парой чисел, разделенных точкой n.m, где n – определяет станцию, с которой брали пробу и m – номер пробы со станции n. Эти же обозначения для проб использованы далее.

На рис. 10-12 слабое разделение на кластеры, поэтому их рассматривать не будем. На рисунке 11 явно видно разделение на два кластера.

Представим результаты кластеризаций в виде таблицы (табл. 13).

Из трёх представленных результатов только в случае биомассы крупных таксономических групп есть явное разделение на кластеры. Это можно заметить на дендрограмме в виде большого межкластерного расстояния. Большое межкластерное расстояние говорит о том, что пары кластеров сильно отличаются друг от друга.

Применим критерий силуэт для определения силы разделения на кластеры. Результат представлен на рис. 13.

Как видно на рис. 13, один кластер выделяется хорошо. Это можно заметить по не сильному отличию значений силуэта для каждого элемента кластера. Второй кластер выделяется хуже (обозначен зелёным цветом). Причем, один элемент оказывается ближе ко первому кластеру, что может быть результатом плохой кластеризации или явного отклонения (выброса в данных).

По кластеризации биомассы можно сделать вывод, что есть один явно выделенный кластер со слабо отличающимися элементами (это следствие малой отличности значения силуэта). Второй кластер, представляет из себя не плотную структуры. Это следствие большого различия значений силуэта для каждого элемента кластера.

Однако данное разделение объясняется сильным доминированием биомассы ручейников. Такой вывод можно сделать из относительно большого значения биомассы ручейников и отсутствия в выделенном кластере проб 1.2, 1.3, 3.2, 6.3,у которых выполняется доминирование биомассы ручейников, но их значение меньше, чем у тех, что попали в кластер.

Таблица 13 – Результаты агломеративной кластеризации по методу Варда на численности по трём кластерам, биомассе по трём кластерам и семействам по двум кластерам.

Индексы	Численность (3 кластера)	Биомасса (2 кластера)	Семейства (2 кластера)
1,1	1	1	1
1,2	1	1	1
1,3	1	1	1
2,1	1	2	1
2,2	2	2	1
2,3	2	2	1
3,1	1	1	1
3,2	1	1	1
3,3	1	1	1
4,1	3	2	2
4,2	1	1	1
4,3	1	2	1
5,1	1	1	1
5,2	1	1	1
5,3	1	1	1
6,1	1	1	1
6,2	1	1	1
6,3	1	1	1
7,1	1	1	1
7,2	1	1	1
7,3	1	1	1
8,1	1	2	1
8,2	1	2	1
8,3	1	2	1
9,1	1	1	1
9,2	1	1	1
9,3	1	1	1

Рассмотрим теперь кластеризацию логарифмированных данных. Их также представим в виде дендрограмм (рис. 14-16).

На дендрограммах кластеризации на логарифмированной численности и биомассе можно заметить, что кластеризация на двух кластерах полностью совпадает. Причем, при трёх и более кластерах различие в разбиении на кластеры присутствует.

На всех трёх дендрограммах видно чёткое деление на два кластера. Однако на кластеризации биомассы можно заметить ещё и чёткое деление на три кластера. Теперь применим критерий силуэт для определения чёткости кластеризации и для определения точного количества кластеров в кластеризации биомассы.

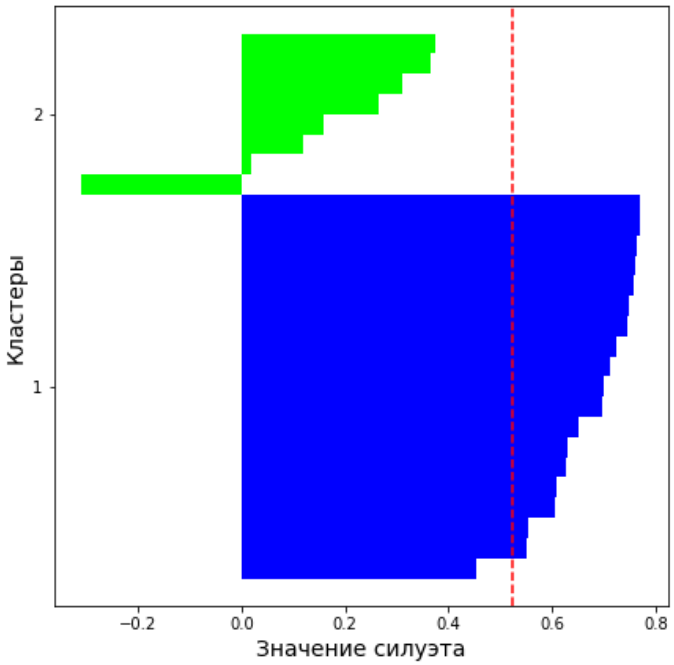

Рисунок 13 – Силуэт для каждого элемента выборки агломеративной кластеризации по методу Варда на логарифмированной численности крупных таксономических групп зообентоса.

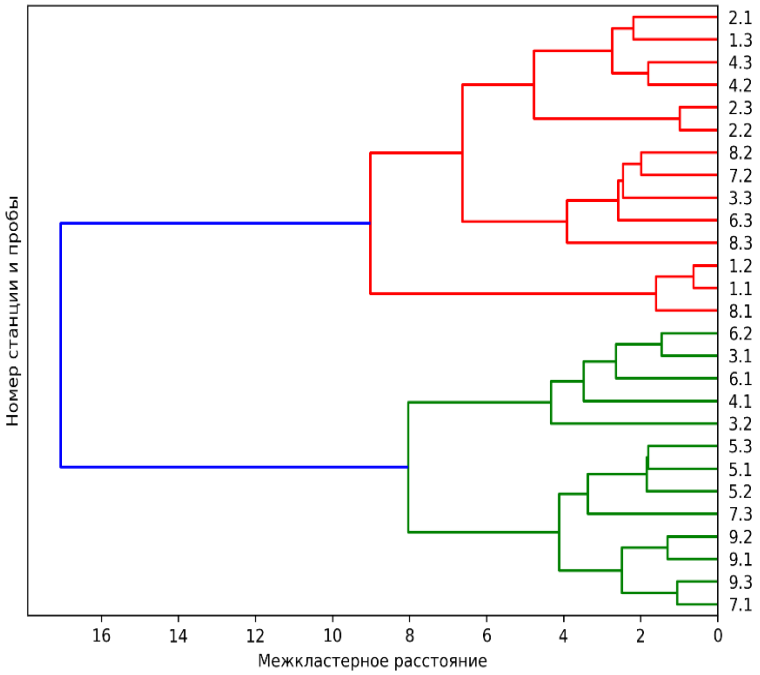

Рисунок 14 – Дендрограмма агломеративной кластеризации по методу Варда на логарифмированной численности крупных таксономических групп зообентоса.

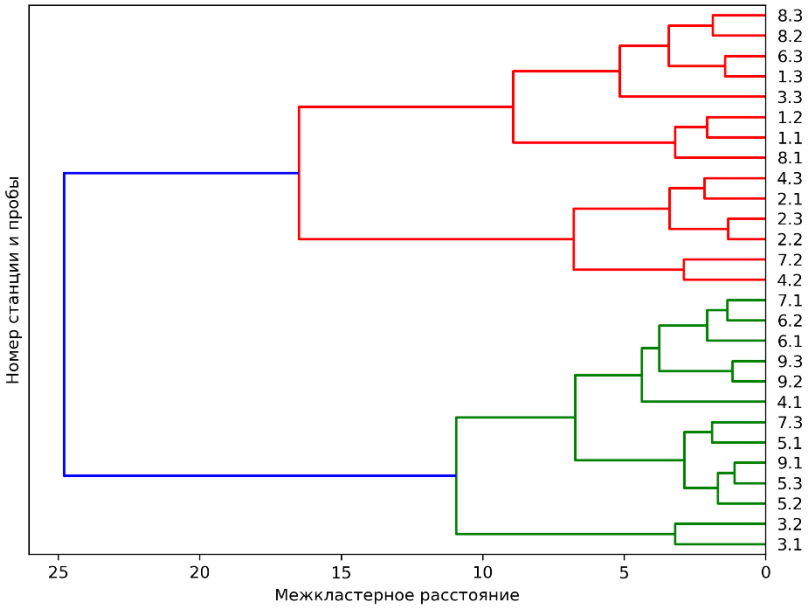

Рисунок 15 – Дендрограмма агломеративной кластеризации по методу Варда на логарифмированной биомассе крупных таксономических групп зообентоса.

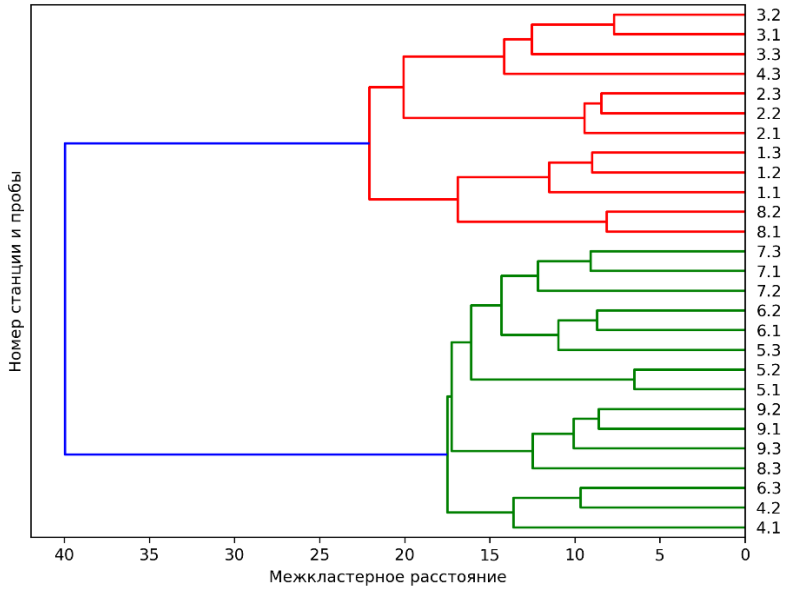

Рисунок 16 – Дендрограмма агломеративной кластеризации по методу Варда на логарифмированной численности семейств зообентоса.

На рис. 17 представлен силуэт каждого элемента выборки для двух кластеров агломеративной кластеризации по методу Варда над логарифмированной численностью таксономических групп. На них можно заметить два не плотных кластера. Это следствие большого отличия значений между силуэтами выборки. Можно заметить, что алгоритмы кластеризуют данные по принципу вхождения/отсутствия веснянок в первоначальных данных. В одном кластере выделяются пробы у которых присутствуют веснянки (зелёный кластер в рисунке 17). Во втором кластере кластере полностью отсутствуют веснянки и в некоторых пробах отсутствуют прочие (проба 8.1 и 3.2). Для биомассы в случае трёх кластеров выполняются такое же условие.

Для кластеризации над логарифмированной биомассой используем силуэт для определения точного количества кластеров. Результаты представлены на рис. 18 и 19.

Из графиков силуэтов можно сделать вывод, что на трёх кластерах средний силуэт выше, чем на двух кластерах. Будем продолжать вычислять силуэт для увеличивающегося количества кластеров, пока силуэт не начнёт падать. Это будет признаком ухудшения кластеризации. Результаты кластеризации представлены в виде таблицы (табл. 14).

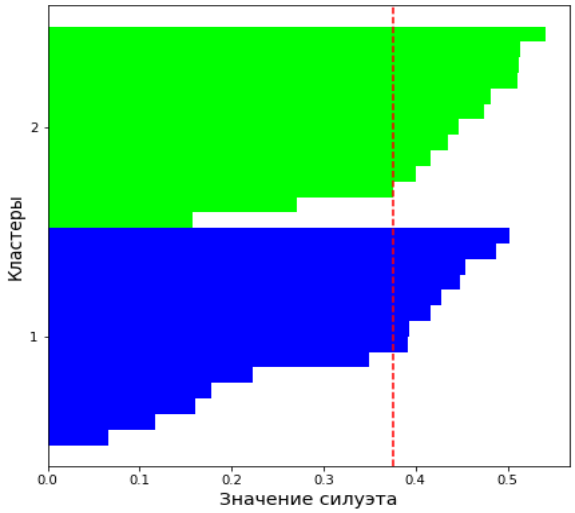

Рисунок 17 – Силуэт для каждого элемента выборки агломеративной кластеризации по методу Варда по логарифмированной численности таксономических групп зообентоса

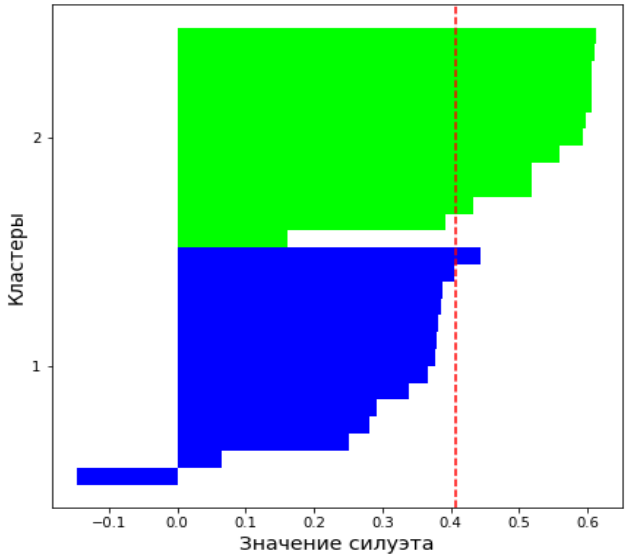

Рисунок 18 – Силуэт для каждого элемента выборки агломеративной кластеризации по методу Варда по логарифмированной численностиаксономических групп зообентоса для двух кластеров

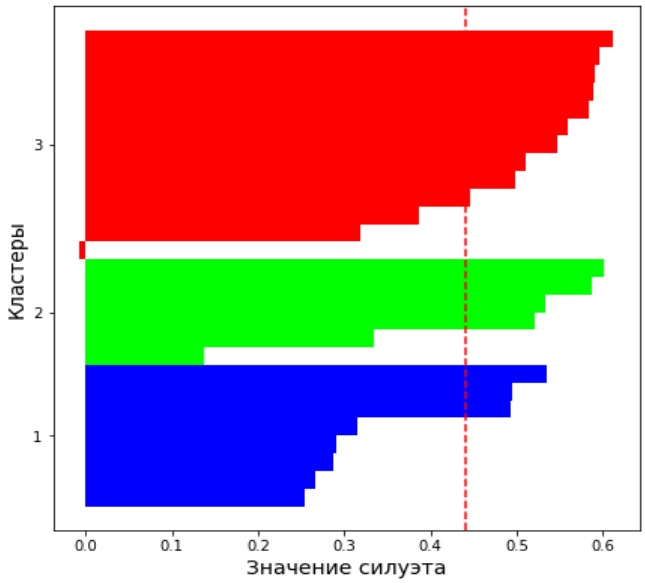

Рисунок 19 – Силуэт для каждого элемента выборки агломеративной кластеризации по методу Варда по логарифмированной биомассе таксономических групп зообентоса для трёх кластеров

Таблица 14. Таблица средних значений силуэтов для агломеративной кластеризации по логарифмированной биомассе

Количество кластеров	Среднее значение силуэта
2	0,408
3	0,440
4	0,440
5	0,451
6	0,429

Как можно заметить, ухудшение кластеризации происходит на шести кластерах. Поэтому при использовании коэффициента силуэт, как меру качества кластеризации, будем считать оптимальным по количеству 5 кластеров. Покажем силуэт выборки в виде графика, как делали ранее. Результат представлен на рис. 20.

Данная кластеризация не даёт никаких результатов в смысле принципа кластеризации данных. Например, в пятый кластер (выделен жёлтым цветом на рис. 20) входят пробы, у которых отсутствуют веснянки. Однако такие же пробы входят и в четвертый кластер. Если сравнивать пробы из эти кластеров, то явных отличий в биомассе зообентоса не выявлено.

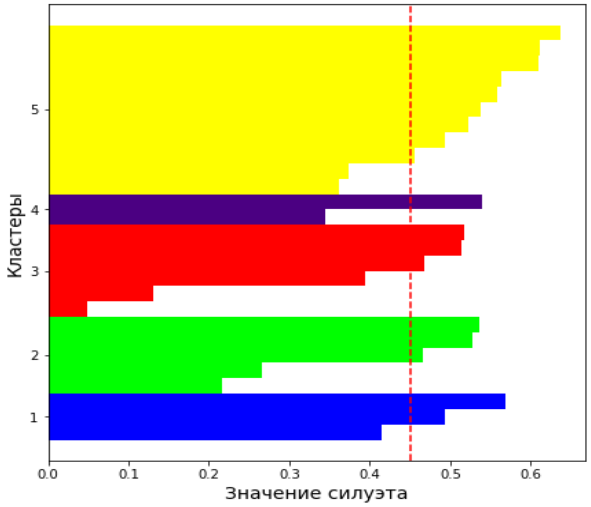

Рисунок 20 – Силуэт для каждого элемента выборки агломеративной кластеризации по методу Варда по логарифмированной биомассе таксономических групп зообентоса для пяти кластеров

Для кластеризации по семейства зообентоса для двух кластеров имеют место следующие значения силуэтов выборки (рис. 21).

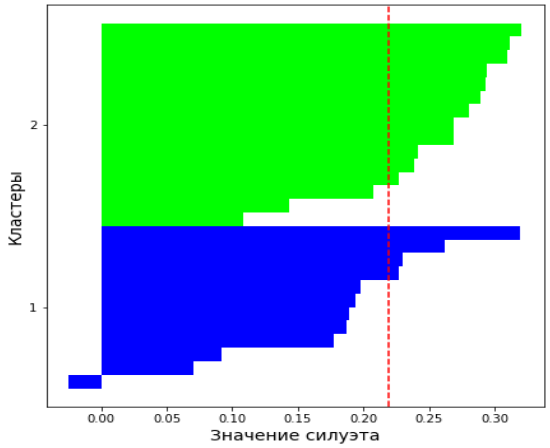

Рисунок 21 – Силуэт для каждого элемента выборки агломеративной кластеризации по методу Варда по логарифмированной численности семейств зообентоса для пяти кластеров

Можно заметить нечеткое деление на два кластера. Первый кластер менее плотный чем второй. Это видно через сильно отличающиеся значения силуэтов выборки. Также общее значение силуэтов мало по сравнению с другими кластеризациями логарифмированной биомассы и численности зообентоса. Из-за большого количества семейств сложно определить возможный принцип разделения на кластеры. Однако в первый кластер (выделен синим цветом на рис. 21) выделяются отсутствие следующих семейств: caenidae, taeniopterycidae, brachycentridae, valvatidae, planorbidae, bivalvia, lumbriculidae, tipulidae. Во втором кластере (выделен зелёным цветом на рис. 21) отсутствуют другие семейства: chloroperlidae, pteronarcyidae, glossosomatidae.

Также если рассмотреть кластеризацию данных на уровне трёх кластеров, то можно заметить, что кластеры согласуются с географическим районированием реки Кан и сменой гидрологических условий. Однако, силуэт показывает уменьшение значение при трёх кластера. Что может быть следствием того, что силуэт плохо определяет в таких случаях пространственную динамику.

Представим результаты в виде таблицы попадания в кластеры элементов выборки (табл. 15).

Таблица 15 – Результаты агломеративной кластеризации по методу Варда на логарифмированной численности по трём кластерам, логарифмированной биомассе по трём кластерам и логарифмированным семействам по двум кластерам.

Индексы	Численность (2 кластера)	Биомасса (5 кластеров)	Семейства (2 кластера)	Семейства (3 кластера)
1,1	1	1	1	1
1,2	1	1	1	1
1,3	1	2	1	1
2,1	1	3	1	2
2,2	1	3	1	2
2,3	1	3	1	2
3,1	2	4	1	2
3,2	2	4	1	2
3,3	1	2	1	2
4,1	2	5	2	3
4,2	1	3	2	3
4,3	1	3	1	2
5,1	2	5	2	3
5,2	2	5	2	3
5,3	2	5	2	3
6,1	2	5	2	3
6,2	2	5	2	3
6,3	1	2	2	3
7,1	2	5	2	3
7,2	1	3	2	3
7,3	2	5	2	3
8,1	1	1	1	1
8,2	1	2	1	1
8,3	1	2	2	3
9,1	2	5	2	3
9,2	2	5	2	3
9,3	2	5	2	3

Теперь рассмотрим кластеризацию над долями зообентоса (рис. 22-24).

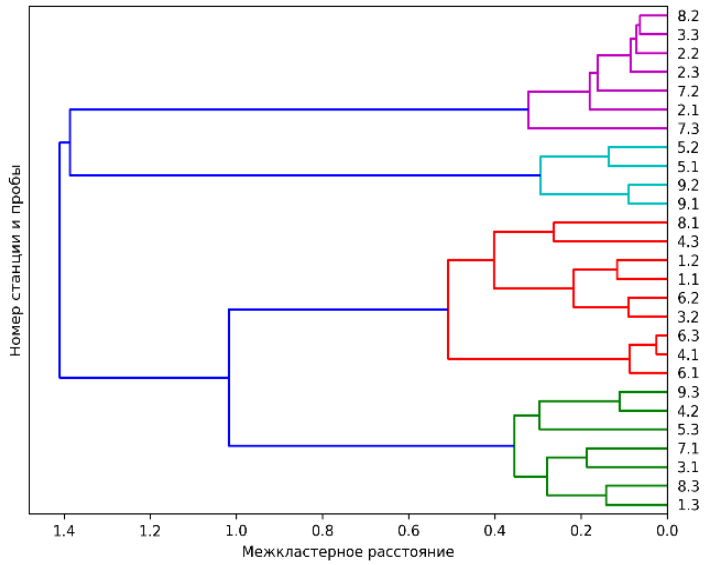

Рисунок 22 – Дендрограмма агломеративной кластеризации по методу Варда на долях численности крупных таксономических групп зообентоса.

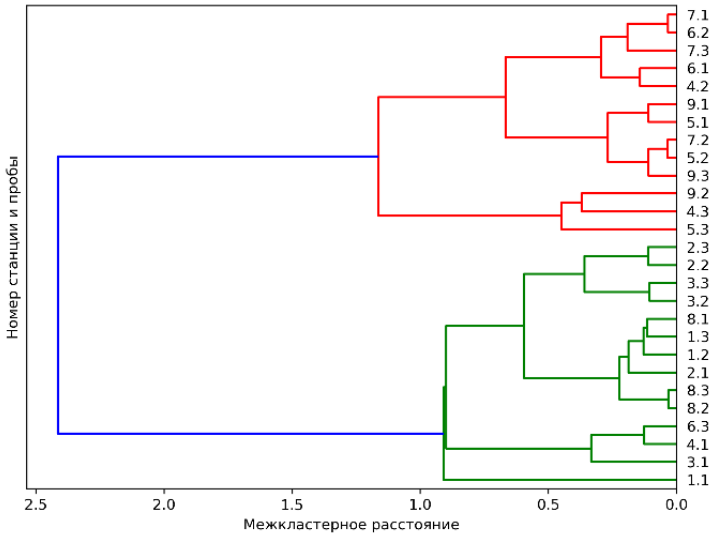

Рисунок 23 – Дендрограмма агломеративной кластеризации по методу Варда на долях биомассы крупных таксономических групп зообентоса.

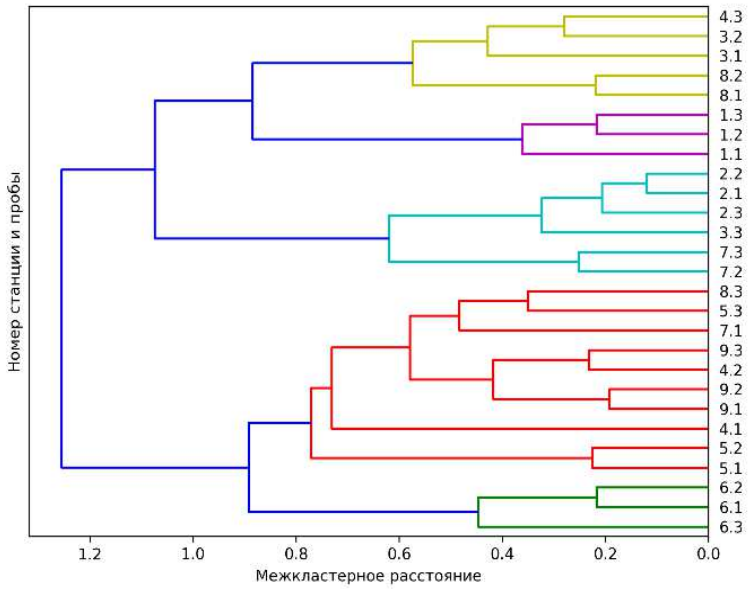

Рисунок 24 – Дендрограма агломеративной кластеризации по методу Варда на долях численности семейств зообентоса.

Используем коэффициент силуэт для определения возможного количества кластеров. Далее будут приводиться только результаты. Схема действий проходит такая же как проводилась выше. Результаты критерия силуэт и общая раскраска элементов выборки по попаданию в кластеры рис. 25-27 и в табл. 16.

Кластеризация на долях численности крупных таксономических групп (рис. 25) при четырёх кластерах выявляет деление таксонов по характеру доминирования той или иной группы беспозвоночных (поденки, ручейники, двукрылые и прочие). В первом кластере (выделен синим цветом) присутствует доминирование ручейников. Во втором кластере (выделен зелёным цветом) выполняется доминирование поденок. В третьем кластере (выделен синим цветом на) происходит доминирование двукрылых. И в оставшемся четвёртом кластере (выделен фиолетовым цветом) присутствует доминирование прочих.

В случае кластеризации биомассы разделение на два кластера происходит по принципу доминирования биомассы зообентоса. В первый кластер (изображен синим цветом на рис. 26) входят пробы, у которых доминируют биомасса прочих или подёнок. Во второй кластер (изображен зеленым цветом на рис. 26) входят пробы, у которых доминируют биомасса остальных представителей таксономических групп.

В кластеризации долей численности семейств трудно определить характер кластеризации, идея кластеризации по принципу доминирования таксонов не подходит так как в одних и тех же кластерах выполняется доминирование разных семейств. Кластеризация по принципу отсутствия семейств в пробе тоже не выполняется. Вероятнее всего, биологического смысла в данной кластеризации нет.

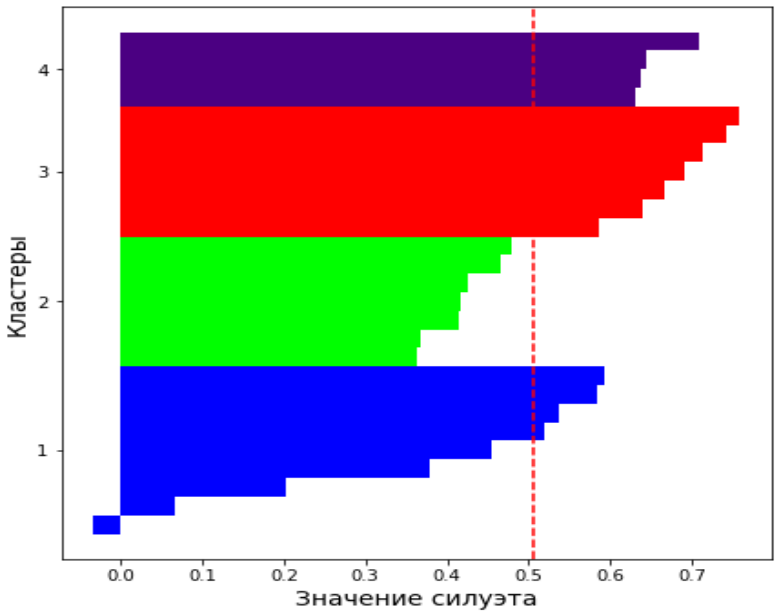

Рисунок 25 – Силуэт для каждого элемента выборки агломеративной кластеризации по методу Варда по долям численности таксономических групп зообентоса

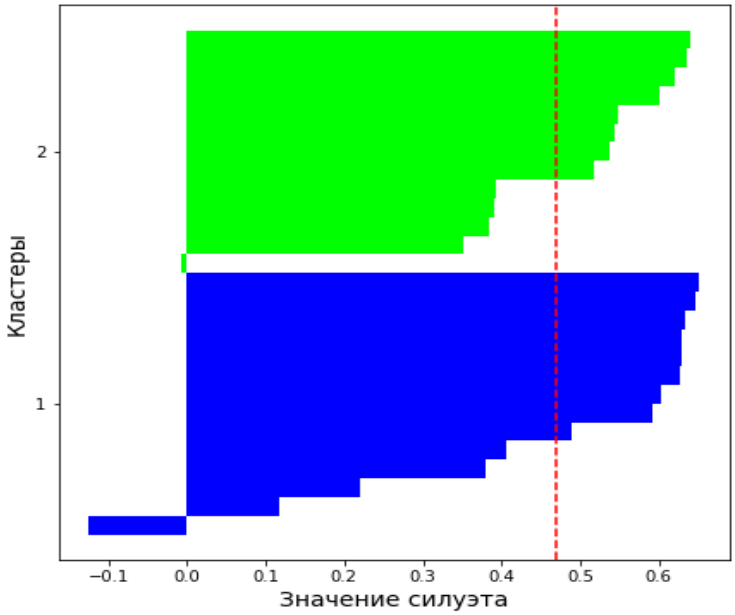

Рисунок 26 – Силуэт для каждого элемента выборки агломеративной кластеризации по методу Варда по долям биомассы таксономических групп зообентоса

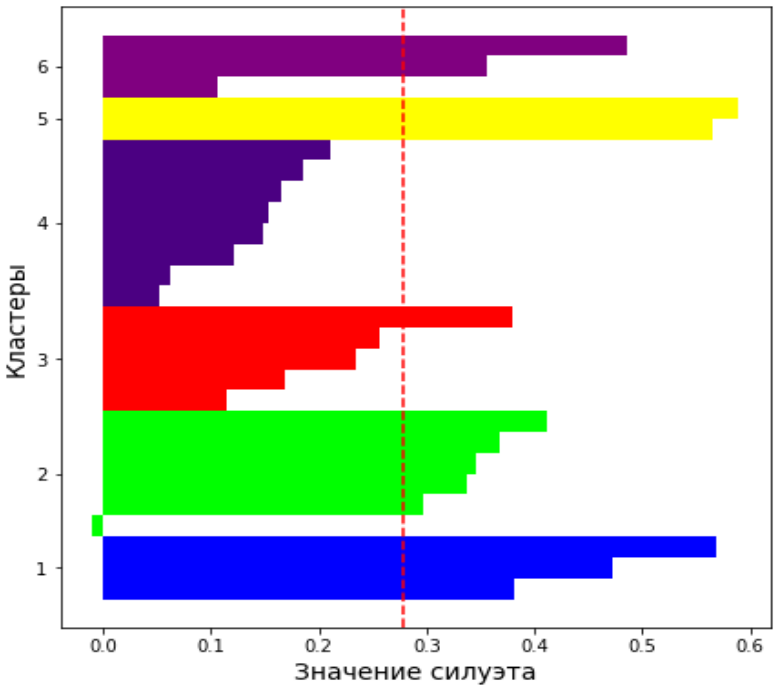

Рисунок 27 – Силуэт для каждого элемента выборки агломеративной кластеризации по методу Варда по долям численности семейств зообентоса

Таблица 16. Таблица попадания в кластеры элементов выборки кластеризации над долями численности и биомассы таксономических групп и долей численности семейств

Индексы	Доли численности таксонов (4 кластера)	Доли биомассы таксонов (2 кластера)	Доли численности семейств (6 кластеров)
1,1	1	1	1
1,2	1	1	1
1,3	2	1	1
2,1	3	1	2
2,2	3	1	3
2,3	3	1	3
3,1	2	1	4
3,2	1	1	4
3,3	3	1	5
4,1	1	1	6
4,2	2	2	5
4,3	1	2	4
5,1	4	2	5
5,2	4	2	5
5,3	2	2	5
6,1	1	2	5
6,2	1	2	5
6,3	1	1	5
7,1	2	2	5

Окончание таблицы 16

Индексы	Доли численности таксонов (4 кластера)	Доли биомассы таксонов (2 кластера)	Доли численности семейств (6 кластеров)
7,2	3	2	2
7,3	3	2	2
8,1	1	1	4
8,2	3	1	4
8,3	2	1	5
9,1	4	2	5
9,2	4	2	5
9,3	2	2	5

Можно заметить, что в кластеризации долей численности три кластера хорошо выделяются, тогда как четвертый кластер (обозначен синим цветом) имеет большой разброс по коэффициенту силуэт. В кластеризации долей биомассы хорошо выделяются два кластера. Хотя у одного присутствуют, элементы у которых значение силуэта мало по сравнению с другими значениями. Также кластеризация долей численности семейств показала, что хорошо выделяются шесть кластеров. Однако среднее значение силуэта мало относительно других кластеризации.

В заключении нашего анализа р. Кан приведем выводы, которые оказались значимы для биологов.

Река Кан (крупный приток Енисея в его среднем течении) на исследованном участке представляет собой ритраль, заселенную холодолюбивыми реобионтными организмами, среди которых качественно и количественно преобладают личинки амфибиотических насекомых (веснянки, поденки, ручейники и двукрылые). Максимальным числом видов представлены хирономиды и ручейники. Температура воды является одним из существенных факторов, определяющим численность веснянок, двукрылых и группы “прочие” (до 30% объясненной дисперсии). Для ручейников отмечено более существенное влияние не температуры, а растворенного в воде кислорода (20% объясненной дисперсии). Кластеризация данных на уровне численности семейств выявила согласованность с географическим районированием р. Кан и сменой гидрологических условий. По продольному профилю реки от верховья наблюдалась смена доминирующих семейств среди поденок в ряду Heptageniidae – Ephemerellidae – Ephemeridae; семейств ручейников – от Glossosomatidae к Hydropsychidae. Ландшафтно-геоморфологические особенности р. Кан в нижнем течении обуславливают выход на лидирующие позиции видов, характерных для верхнего течения.

2.3 Анализ зообентоса группы рек бассейна реки Енисея

Далее будет приведет набор результатов по всем река бассейна реки Енисея.

2.3.1 Корреляционный анализ

Корреляционный анализ структуры бентосных сообществ, основанный на соотношении величин численности основных крупных таксонов, выявил ряд статистически достоверных закономерностей (рис. 28-39). В частности, в большинстве рек согласованно варьировали показатели численности структурообразующих отрядов насекомых (поденок, веснянок и ручейников). Данная закономерность подтвердилась при анализе совокупной базы данных по всем исследованным рекам (рис. 40-41, табл. 17-18).

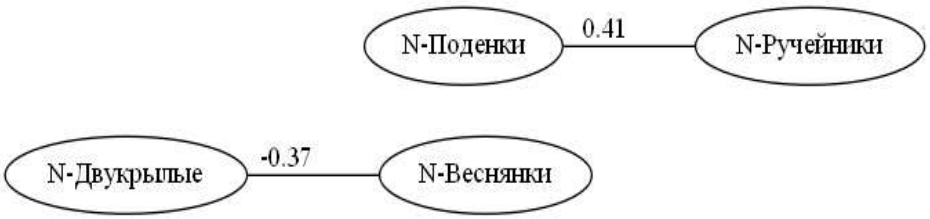

Рисунок 28 – Корреляционный граф для численности таксономических групп реки Абакан

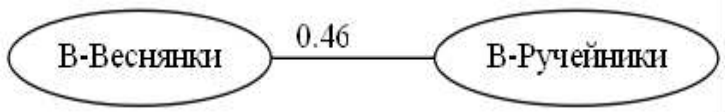

Рисунок 29 – Корреляционный граф для биомассы таксономических групп реки Абакан

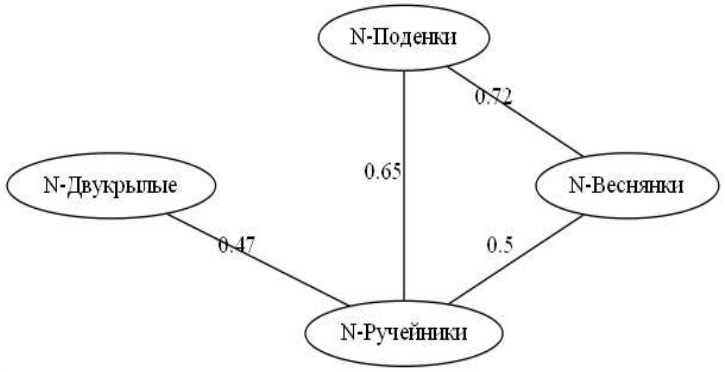

Рисунок 30. – Корреляционный граф для численности таксономических групп реки Мана

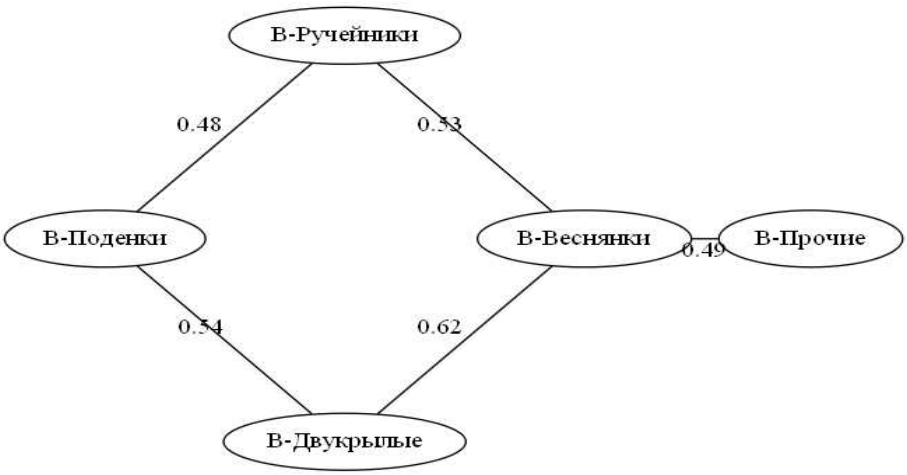

Рисунок 31 – Корреляционный граф для биомассы таксономических групп реки Мана

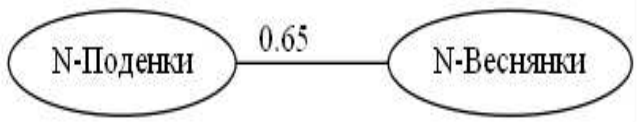

Рисунок 32 – Корреляционный граф для численности таксономических групп реки Агул

Рисунок 33 – Корреляционный граф для биомассы таксономических групп реки Агул

Рисунок 34 – Корреляционный граф для численности таксономических групп реки Кунгус

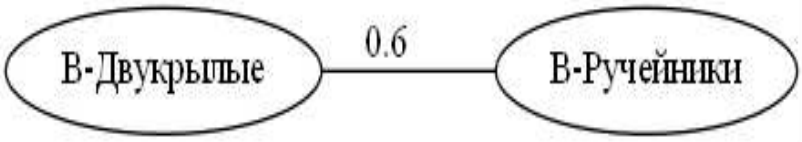

Рисунок 35 – Корреляционный граф для биомассы таксономических групп реки Кунгус

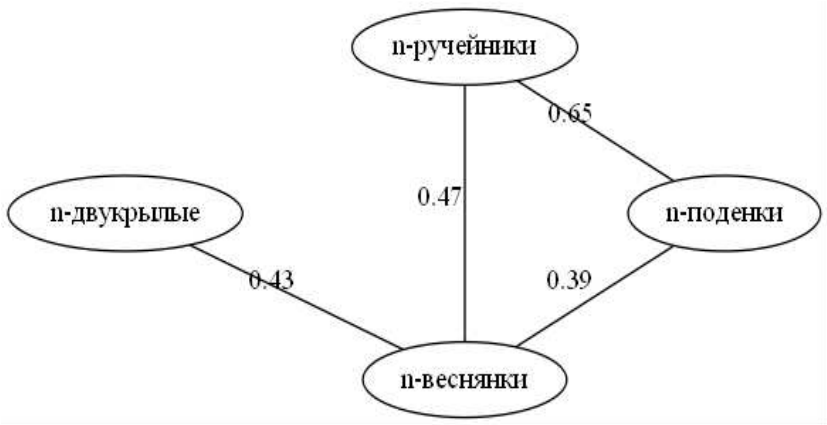

Рисунок 36 – Корреляционный граф для численности таксономических групп реки Кан

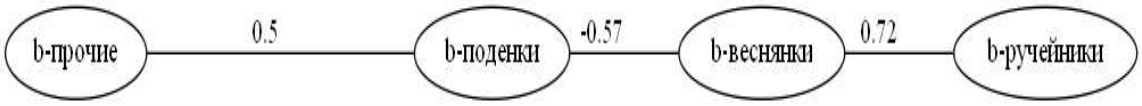

Рисунок 37 – Корреляционный граф для биомассы таксономических групп реки Кан

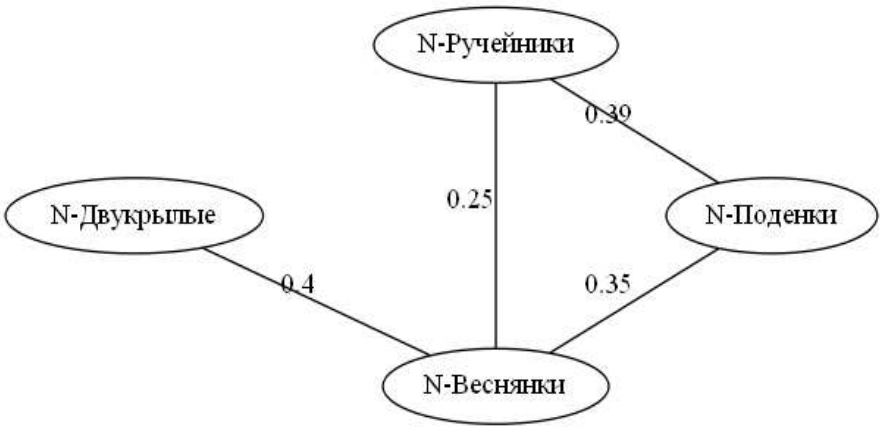

Рисунок 38 – Корреляционный граф для численности таксономических групп рек Кан, Агул и Кунгус

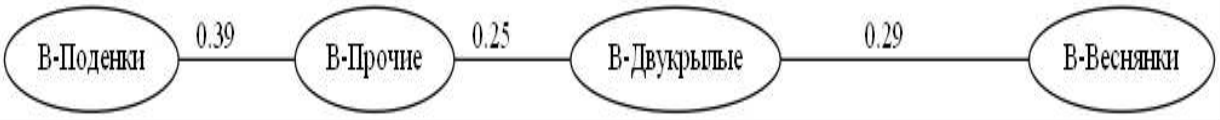

Рисунок 39 – Корреляционный граф для биомассы таксономических групп реки Кан, Агул и Кунгус

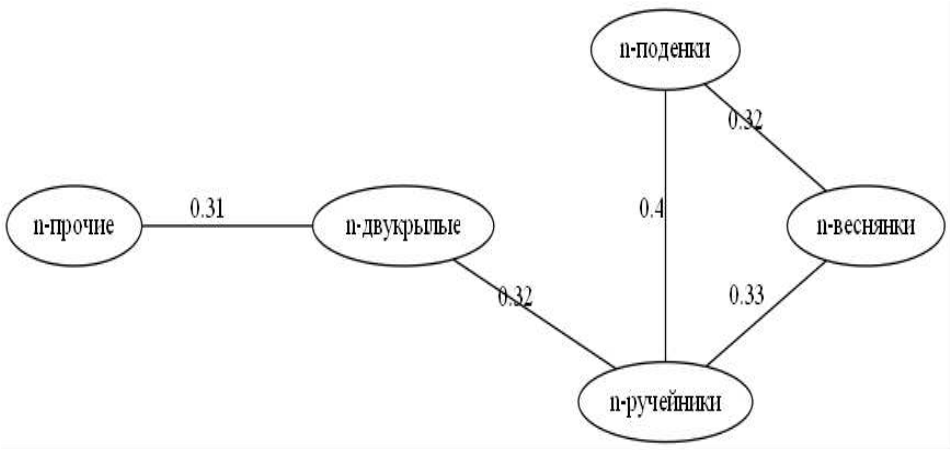

Рисунок 40 – Корреляционный граф для численности таксономических групп всех рек, входящих в анализ

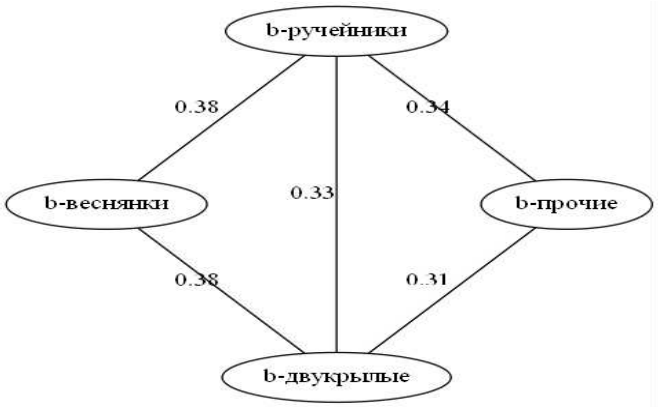

Рисунок 41 – Корреляционный граф для биомассы таксономических групп всех рек, входящих в анализ

Таблица 17 - Таблица достоверных корреляционных связей численности зообентоса по всем рекам бассейна реки Енисея

Достоверные связи							
	Абакан	Кан	Агул	Кунгус	Мана	Кан+Агул+Кунгус	Все реки
Поденки-Ручейники	+	+	-	-	+	+	+
Поденки-Веснянки	-	-	+	-	+	+	+
Веснянки-Ручейники	+	+	-	+	-	+	-
Двукрылые-Прочие	-	-	-	+	-	-	+

Таблица 18 – Таблица достоверных корреляционных связей биомассы по всем рекам бассейна реки Енисея

Достоверные связи							
	Абакан	Кан	Агул	Кунгус	Мана	Кан+Агул+Кунгус	Все реки
Поденки-Ручейники	-	-	-	-	+	-	-
Поденки-Веснянки	-	-	+	-	-	-	-
Веснянки-Ручейники	+	-	+	-	+	-	+
Двукрылые-Веснянки	-	-	-	-	+	+	+
Двукрылые-Поденки	-	-	-	-	+	-	-
Двукрылые-Ручейники	-	-	-	+	-	-	+
Поденки-Прочие	-	-	-	-	-	+	-
Веснянки-Прочие	-	-	-	-	+	-	-
Ручейники-Прочие	-	-	-	-	-	-	+

2.3.2 Регрессионный анализ

В результате линейного регрессионного анализа определена степень влияния температуры воды и содержания кислорода на численность и биомассу основных таксономических групп зообентоса (двукрылые, поденки, веснянки, ручейники).

Среди двух факторов наиболее значимо проявлялось влияние температуры воды, которая обуславливала до 70% дисперсии численности и биомассы отдельных таксонов беспозвоночных. Максимальные R^2 выявлены в р. Кунгус для поденок: 0,7 по численности и 0,73 по биомассе. Кроме того, относительно высокие R^2 (0,39–0,57) зафиксированы в р. Мана для поденок, веснянок и ручейников. Следует отметить, что в р. Мана в период исследования диапазон колебаний температуры воды (8,4–21,7°С) оказался шире, чем в других реках, что, вероятно, и объясняет высокие значения R^2 сразу для нескольких таксонов. В остальных реках температура воды объясняла максимум 33% дисперсии численности и биомассы беспозвоночных.

Для поденок во всех исследованных реках (за исключением р. Кан) выявлены достоверные коэффициенты детерминации численности/биомассы и температуры воды (R^2 колебался от 0,16 до 0,73).

Содержание кислорода в воде обуславливало до 80% дисперсии численности и биомассы отдельных таксонов беспозвоночных. Максимальные R^2 выявлены, как и для температуры, в р. Кунгус для поденок: 0,80 по численности и 0,59 по биомассе.

При анализе совокупной базы данных выявлено, что содержание кислорода в воде достоверно влияло лишь на численность и биомассу ручейников: R^2 составил 0,18 и 0,29 соответственно.

2.3.3 Кластерный анализ

Анализ закономерностей пространственного распределения сообществ гидробионтов относится к фундаментальным задачам экологии и гидробиологии. Известно, что в реках по мере удаления от истока происходят закономерные изменения среды обитания гидробионтов, что влечет за собой гетерогенность видовой структуры сообществ по продольному профилю реки. Кластерный анализ помогает проверить гипотезу о пространственной неоднородности в структурной организации зообентоса по продольному профилю реки.

Анализ структуры донных сообществ на различных участках исследованных рек методом агломеративной кластеризации на основе доли крупных таксонов в общей численности и биомассе зообентоса выявил наличие 3-4 кластеров в каждой реке. Каждый кластер характеризуется доминированием той или иной группы беспозвоночных животных в зообентосе (поденки, ручейники, двукрылые или “прочие”).

В р. Абакан (табл. 19) станция в верховье исследованного участка формирует отдельный кластер (№ 1), характеризующийся доминирование отряда двукрылых; далее по течению расположены станции, где доминировали веснянки (кластер № 4), и наконец, весь нижний участок реки объединился в один большой кластер (№ 3), представленный сообществами зообентоса, где лидировали поденки. Кластер № 2 включал единичные пробы, где первостепенную роль играли представители редких и малочисленных таксономических групп, т.е. “прочие”.

Таблица 19 – Результаты агломеративной кластеризации на долях численности и биомассы по реке Абакан

Номер пробы	Доли численности (4 кластера)
1,1	1
1,2	1
1,3	1
1,4	1
1,5	1
1,6	1
2,1	2
2,2	3
2,3	3
3,1	1
3,2	4

Окончание таблицы 19

Номер пробы	Доли численности (4 кластера)
3,3	4
4,1	4
4,2	4
4,3	3
5,1	3
5,2	3
5,3	3
6,1	3
6,2	3
6,3	3
7,1	2
7,2	3
7,3	3
8,1	3
8,2	3
8,3	3
9,1	3
9,2	3
9,3	3

В р. Мана (табл. 20) донные сообщества на 60 % состояли из двукрылых насекомых, что объясняет наличие одного обширного кластера (№ 3). Остальные кластеры включают в себя единичные пробы, в которых на лидирующие позиции выходили поденки (кластер № 2), “прочие” (кластер № 1) или, где доминирующий комплекс состоял из нескольких отрядов (кластер № 4).

Таблица 20 – Результаты агломеративной кластеризации на долях численности по реке Мана

Номер пробы	Доли численности (4 кластера)
1,1	1
1,2	2
1,3	1
2,1	3
2,2	3
3,1	2
4,1	4
4,2	4
5,1	2
5,2	4
6,1	3
6,2	3
6,3	3
7,1	4

Окончание таблицы 19

Номер пробы	Доли численности (4 кластера)
7,2	3
8,1	3
9,1	4
10,1	1
10,2	1
10,3	3

В р. Кан кластеризация (табл. 21) отрядов беспозвоночных животных по относительной численности показала наличие трех равноценных кластеров (№ 1-3), характеризующихся соответственно доминированием ручейников, поденок и двукрылых. Кластер № 4 – самый малочисленный, представлен “прочими” организмами. Чередование кластеров было не связано с продольным профилем реки. Однако кластеризация отрядов беспозвоночных животных по относительной биомассе выявила два обширных кластера, демонстрирующих, что в верховье исследованного участка р. Кан доминировали преимущественно ручейники, которых ниже по течению сменили поденки.

Таблица 21 – Результаты агломеративной кластеризации на долях численности и биомассы по реке Кан

Номер пробы	Доли численности (4 кластера)	Доли биомассы (4 кластера)
1.1	1	1
1.2	1	2
1.3	2	2
2.1	3	2
2.2	3	2
2.3	3	2
3.1	2	2
3.2	1	2
3.3	3	2
4.1	1	2
4.2	2	3
4.3	1	4
5.1	4	3
5.2	4	3
5.3	2	4
6.1	1	3
6.2	1	3
6.3	1	2
7.1	2	3
7.2	3	3
7.3	3	3
8.1	1	2
8.2	3	2

Окончание таблицы 21

8.3	2	2
9.1	4	3
9.2	4	4
9.3	2	3

В р. Агул (табл. 22) по долям численности выявлено два больших кластера, согласующихся с лидированием двукрылых и поденок. Остальные два кластера включают в себя единичные пробы, где так же доминировали двукрылые и поденки, но с вариациями других отрядов. Кластерный анализ по относительной биомассе показал более выраженное лидирование поденок, причем в разных районах реки. Остальные три кластера не столь многочисленны и объединяют пробы зообентоса, где основу биомассы составляли веснянки, двукрылые и другие вариации.

Таблица 22 – Результаты агломеративной кластеризации на долях численности и биомассы по реке Агул

Номер пробы	Доли численности (4 кластера)	Доли биомассы (4 кластера)
1,1	1	1
1,2	1	2
1,3	1	1
2,1	2	3
2,2	2	1
2,3	2	3
3,1	1	1
3,2	1	4
3,3	1	2
4,1	2	1
4,2	2	1
4,3	2	3
5,1	1	1
5,2	1	2
5,3	2	3
6,1	2	2
6,2	1	1
6,3	3	4
7,1	3	2
7,2	1	3
7,3	3	4
8,1	4	3
8,2	2	1
8,3	2	1
9,1	1	3
9,2	2	3
9,3	2	1

В р. Кунгус (табл. 23) кластеризация как по относительной численности, так и по биомассе, продемонстрировала лидирование двукрылых на самой верхней станции исследованного участка (кластер № 1) и смешение комплексов ниже по течению (кластеры № 2 и № 3).

Таблица 23 – Результаты агломеративной кластеризации на долях численности и биомассы по реке Кунгус

Номер пробы	Доли численности 4 кластера	Доли биомассы 4 кластера
10,1	1	1
10,2	1	1
10,3	1	1
11,1	2	2
11,2	3	3
11,3	2	2
12,1	3	2
12,2	2	2
12,3	2	1
13,1	2	1
13,2	2	3
13,3	2	1

Совокупная база данных по всем рекам (Приложение А). При анализе долей численности отрядов выявлено разделение на 4 кластера по принципу доминирования двукрылых (кластер № 1), поденок (кластер № 3) и веснянок (кластер № 4). Кластер № 2 объединил переходные сообщества зообентоса, в которых структуру формировали 2 и более отрядов. Самыми многочисленными являются два кластера – № 1 и № 3; наименьший – кластер № 4, в него вошли пробы лишь из р. Абакан в середине исследованного участка. Кластер № 1 и 3 были представлены всеми водотоками, однако в кластер № 3 из р. Маны вошли лишь единичные пробы. В р. Абакан наблюдалась смена кластеров № 1–4–3 вдоль продольного профиля реки, причем в кластер № 3 вошли большинство станций из среднего и нижнего течения. Аналогичная зональность структуры зообентоса по продольному профилю реки наблюдалась в Кунгусе: кластер № 1 объединил пробы из верховья, кластер № 3 – из низовья исследованного участка.

При анализе относительной биомассы отрядов выявлено разделение на 4 кластера по принципу доминирования поденок (кластер № 1), веснянок (кластер № 3) и ручейников (кластер № 4). Кластер № 2 объединил переходные сообщества зообентоса, в которых биомассу формировали 2 и более отрядов.

При анализе биомассы зообентоса лидирующие позиции завоевывают либо многочисленные животные, либо крупноразмерные.

Анализ структуры донных сообществ на различных участках исследованных рек методом агломеративной кластеризации на основе

логарифмированной численности семейств беспозвоночных животных выявил в большинстве исследованных рек наличие 3 кластеров. Каждый кластер характеризуется доминированием или присутствием и определенной численностью одного или нескольких семейств беспозвоночных животных в зообентосе; база данных состояла из 39 семейств. Распределение по кластерам происходило на основе вариативных связей нескольких семейств, формирующих в той или иной степени основу структуры зообентоса в исследованных реках: поденки Ephemeridae, Ephemerellidae, Heptageniidae, Baetidae; ручейники Hydropsychidae, Arctopsychidae, Stenopsychidae, Leptoceridae; двукрылые Chironomidae; веснянки Perlodidae, Pteronarcyidae. Во всех реках наблюдалась пространственная неоднородность структуры зообентоса, связанная с продольным градиентом вдоль русла. Станции, расположенные в верховье, в середине и низовье исследованных участков рек распределялись, преимущественно, по разным кластерам. Постепенный характер изменений структуры зообентоса вдоль русла рек объясняет наличие промежуточных биоценозов, которые нарушают непрерывность градиента изменений и попадают в кластеры из других пространственных участков.

При анализе совокупной база данных по всем рекам (Приложение Б) наиболее адекватно происходило разделение на 5 кластеров, куда вошли станции из различных районов (верхний, средний, нижний) исследованных участков рек:
Кластер № 1: Абакан (большинство), Агул (выборочно);
Кластер № 2: Кан (верхний), Кунгус (нижний);
Кластер № 3: Абакан (верховье), Агул (большинство), Кунгус (середина);
Кластер № 4: Кан (середина и низовье);
Кластер № 5: Мана (верховье середина), Кунгус (верховье).

Следует обратить внимание, что рекой Кан был сформирован отдельный кластер (№ 4), что указывает на особенность структуры донных сообществ, связанную, вероятно, с антропогенной трансформацией реки. Среди всех исследованных рек именно Кан наиболее сильно подвержен антропогенной нагрузке, вследствие чего вода загрязнена нефтепродуктами и тяжелыми металлами.

Таким образом, кластерный анализ выявил, что, несмотря на сходные геоморфологические и гидрологические характеристики, исследованные реки относительно уникальны, что проявляется в неоднородности состава бентосных сообществ среди них. Во всех исследованных реках прослеживалась смена доминирующих комплексов на уровне семейств по продольному профилю реки.

ЗАКЛЮЧЕНИЕ

В работе были получени следующие результаты:

1. Реализованы методы для упрощения (автоматизации) процесса получения результатов корреляционного, регрессионного и кластерного анализов.

2. Реализованы получения результатов в табличном и графическом виде для корреляционного, регрессионного и кластерного анализов.

3. Проведён анализ данных зообентоса на пяти малых реках бассейна реки Енисея (Кан, Абакан, Мана, Агул, Кунгус) с использованием регрессионного, корреляционного и кластерного анализов.

Реализованные методы получения результатов могут быть использованны в дальнейшем для проведения анализа на иных данных. Были получены теоретические сведения о структурности и пространственной динамики зообентоса на малых реках бассейна реки Енисея.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Крупкина, Т.В Теория вероятностей и математическая статистика, часть 1 [Электронный ресурс] : электронный курс лекций / Т. В. Крупкина. Красноярск: Сибирский федеральный университет. 2011. 159 с.

[2] Крупкина, Т.В Теория вероятностей и математическая статистика, часть 2 [Электронный ресурс] : электронный курс лекций / Т. В. Крупкина. Красноярск: Сибирский федеральный университет. 2011. 237 с.

[3] Кобзарь, А.И. Прикладная математическая статистика. Для инженеров и научных работников / А.И. Кобзарь. – Москва : Физмалит, 2006. – 816 с.

[4] Факторный, дискриминантный и кластерный анализ : пер. с англ. / Дж.-О. Ким, Ч.У. Мьюллер, У.Р. Клекка [и др.]; под ред. И.С. Енюкова. – Москва : Финансы и статистика, 1989. – 215 с.

[5] Arthur, D. K-means++: the advantages of careful seeding / D. Arthur, S. Vasilvitskii // Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms. – 2007. – С. 1027-1035.

[6] Rousseeuw, P.J Silhouettes: a graphical aid to the interpretation and validation of cluster analysis / P.J. Rousseeuw // Journal of computational and applied mathematics. – 1987 – № 20. – С. 53-65.

[7] Shapiro, S.S. An analysis of variance test for (complete samples) / S.S. Shapiro, M.B. Wilk // Biometrika. – 1965. – Т. 52, № 3/4 – С. 591-611.

ПРИЛОЖЕНИЕ А

Результаты агломеративной кластеризации на долях численности таксономических групп бассейна реки Енисея

Таблица А.1 – Результаты попадания в кластеры проб зообентоса на долях численности таксономических групп бассейна реки Енисея

Река	Номер пробы	Доли численности (4 кластера)	Доли биомассы (4 кластера)
Абакан	1.1.1	1	1
Абакан	1.1.2	1	2
Абакан	1.1.3	1	2
Абакан	1.1.4	1	1
Абакан	1.1.5	1	2
Абакан	1.1.6	2	2
Абакан	1.2.1	2	1
Абакан	1.2.2	3	1
Абакан	1.2.3	2	3
Абакан	1.3.1	1	3
Абакан	1.3.2	4	3
Абакан	1.3.3	4	3
Абакан	1.4.1	4	2
Абакан	1.4.2	4	2
Абакан	1.4.3	3	1
Абакан	1.5.1	3	1
Абакан	1.5.2	3	1
Абакан	1.5.3	3	1
Абакан	1.6.1	3	1
Абакан	1.6.2	3	3
Абакан	1.6.3	3	1
Абакан	1.7.1	2	1
Абакан	1.7.2	3	1
Абакан	1.7.3	3	1
Абакан	1.8.1	3	1
Абакан	1.8.2	3	1
Абакан	1.8.3	3	1
Абакан	1.9.1	3	1
Абакан	1.9.2	3	1
Абакан	1.9.3	3	1
Мана	2.1.1	2	4
Мана	2.1.2	3	4
Мана	2.1.3	2	4
Мана	2.2.1	1	4
Мана	2.2.2	1	4
Мана	2.3.1	3	4
Мана	2.4.1	4	4
Мана	2.4.2	1	4

Продолжение таблицы А.1

Река	Номер пробы	Доли численности (4 кластера)	Доли биомассы (4 кластера)
Мана	2.5.1	3	2
Мана	2.5.2	2	2
Мана	2.6.1	1	2
Мана	2.6.2	1	2
Мана	2.6.3	1	2
Мана	2.7.1	1	4
Мана	2.7.2	1	4
Мана	2.8.1	1	2
Мана	2.9.2	2	2
Мана	2.10.1	2	2
Мана	2.10.2	2	2
Мана	2.10.3	2	1
Кан	3.1.1	3	3
Кан	3.1.2	3	4
Кан	3.1.3	3	4
Кан	3.2.1	1	4
Кан	3.2.2	1	4
Кан	3.2.3	1	4
Кан	3.3.1	3	1
Кан	3.3.2	3	4
Кан	3.3.3	1	4
Кан	3.4.1	3	1
Кан	3.4.2	3	1
Кан	3.4.3	2	2
Кан	3.5.1	2	1
Кан	3.5.2	2	1
Кан	3.5.3	3	2
Кан	3.6.1	3	1
Кан	3.6.2	3	1
Кан	3.6.3	3	1
Кан	3.7.1	3	1
Кан	3.7.2	1	1
Кан	3.7.3	1	1
Кан	3.8.1	3	4
Кан	3.8.2	1	4
Кан	3.8.3	3	4
Кан	3.9.1	2	1
Кан	3.9.2	2	2
Кан	3.9.3	3	1
Агул	4.1.1	1	1
Агул	4.1.2	1	1
Агул	4.1.3	1	1
Агул	4.2.1	3	3
Агул	4.2.2	3	1

Окончание таблицы А.1

Агул	4.2.3	3	3
Агул	4.3.1	1	1
Агул	4.3.2	1	2
Агул	4.3.3	1	2
Агул	4.4.1	3	1
Агул	4.4.2	3	1
Агул	4.4.3	3	3
Агул	4.5.1	1	1
Агул	4.5.2	1	2
Агул	4.5.3	3	3
Агул	4.6.1	3	3
Агул	4.6.2	1	1
Агул	4.6.3	1	2
Агул	4.7.1	1	2
Агул	4.7.2	1	3
Агул	4.7.3	1	2
Агул	4.8.1	4	3
Агул	4.8.2	3	1
Агул	4.8.3	3	1
Агул	4.9.1	1	3
Агул	4.9.2	3	3
Агул	4.9.3	3	1
Кунгус	5.1.1	2	4
Кунгус	5.1.2	1	4
Кунгус	5.1.3	1	2
Кунгус	5.2.1	1	2
Кунгус	5.2.2	3	2
Кунгус	5.2.3	1	3
Кунгус	5.3.1	3	3
Кунгус	5.3.2	2	3
Кунгус	5.3.3	3	1
Кунгус	5.4.1	3	1
Кунгус	5.4.2	3	2
Кунгус	5.4.3	2	1

ПРИЛОЖЕНИЕ Б

Результаты агломеративной кластеризации на численности семейств зообентоса бассейна Енисея

Таблица Б.1 – Результаты попадания в кластеры проб зообентоса на численнсти семейств зообентоса бассейна Енисея

Река	Номер пробы	Доли численности (4 кластера)
Абакан	1.1.1	3
Абакан	1.1.2	1
Абакан	1.1.3	3
Абакан	1.1.4	3
Абакан	1.1.5	3
Абакан	1.1.6	3
Абакан	1.2.1	1
Абакан	1.2.2	1
Абакан	1.2.3	1
Абакан	1.3.1	3
Абакан	1.3.2	1
Абакан	1.3.3	1
Абакан	1.4.1	1
Абакан	1.4.2	1
Абакан	1.4.3	3
Абакан	1.5.1	1
Абакан	1.5.2	1
Абакан	1.5.3	1
Абакан	1.6.1	1
Абакан	1.6.2	1
Абакан	1.6.3	1
Абакан	1.7.1	1
Абакан	1.7.2	1
Абакан	1.7.3	1
Абакан	1.8.1	3
Абакан	1.8.2	3
Абакан	1.8.3	3
Абакан	1.9.1	1
Абакан	1.9.2	1
Абакан	1.9.3	1
Мана	2.1.1	5
Мана	2.1.2	5
Мана	2.1.3	5
Мана	2.2.1	5
Мана	2.2.2	3
Мана	2.3.1	5
Мана	2.4.1	1
Мана	2.4.2	3

Продолжение таблицы Б.1

Мана	2.5.1	5
Мана	2.5.2	5
Мана	2.6.1	3
Мана	2.6.2	3
Мана	2.6.3	3
Мана	2.7.1	2
Мана	2.7.2	2
Мана	2.8.1	5
Мана	2.9.2	4
Мана	2.10.1	3
Мана	2.10.2	3
Мана	2.10.3	3
Кан	3.1.1	1
Кан	3.1.2	1
Кан	3.1.3	1
Кан	3.2.1	2
Кан	3.2.2	2
Кан	3.2.3	2
Кан	3.3.1	2
Кан	3.3.2	2
Кан	3.3.3	3
Кан	3.4.1	4
Кан	3.4.2	4
Кан	3.4.3	2
Кан	3.5.1	4
Кан	3.5.2	4
Кан	3.5.3	4
Кан	3.6.1	4
Кан	3.6.2	4
Кан	3.6.3	4
Кан	3.7.1	4
Кан	3.7.2	4
Кан	3.7.3	4
Кан	3.8.1	1
Кан	3.8.2	1
Кан	3.8.3	1
Кан	3.9.1	4
Кан	3.9.2	4
Кан	3.9.3	4
Агул	4.1.1	3
Агул	4.1.2	3
Агул	4.1.3	3
Агул	4.2.1	1
Агул	4.2.2	1
Агул	4.2.3	1

Окончание таблицы Б.1

Агул	4.3.1	3
Агул	4.3.2	3
Агул	4.3.3	3
Агул	4.4.1	3
Агул	4.4.2	3
Агул	4.4.3	3
Агул	4.5.1	3
Агул	4.5.2	3
Агул	4.5.3	1
Агул	4.6.1	1
Агул	4.6.2	3
Агул	4.6.3	3
Агул	4.7.1	3
Агул	4.7.2	3
Агул	4.7.3	3
Агул	4.8.1	1
Агул	4.8.2	1
Агул	4.8.3	1
Агул	4.9.1	3
Агул	4.9.2	3
Агул	4.9.3	1
Кунгус	5.1.1	5
Кунгус	5.1.2	5
Кунгус	5.1.3	5
Кунгус	5.2.1	3
Кунгус	5.2.2	3
Кунгус	5.2.3	3
Кунгус	5.3.1	3
Кунгус	5.3.2	3
Кунгус	5.3.3	3
Кунгус	5.4.1	2
Кунгус	5.4.2	2
Кунгус	5.4.3	2

ПРИЛОЖЕНИЕ В

Код программ

Код получения результатов корреляционного анализа:

```
from scipy.stats import spearmanr, pearsonr
from itertools import combinations as comb
import pandas as pd
from correlation.corrgraph import make_and_save_graph_correlation

CORRELATION_METHODS = ["spearman", "pearson"]

def results_correlation(x, method="spearman", n_round=3):
    if method == CORRELATION_METHODS[0]:
        correlation_func = spearmanr
    elif method == CORRELATION_METHODS[1]:
        correlation_func = pearsonr
    else:
        print("Warning: не правильно выбран method")
        return None, None

    n_cols = x.shape[1]
    sp_vals = [[1]*n_cols for _ in range(n_cols)]
    p_vals = [[0]*n_cols for _ in range(n_cols)]

    for _ in comb(range(n_cols), 2):
        n_col1, n_col2 = _
        col1 = x.columns[n_col1]
        col2 = x.columns[n_col2]

        sp, p = correlation_func(x[col1], x[col2])
        sp_vals[n_col1][n_col2] = sp_vals[n_col2][n_col1] = round(sp, n_round)
        p_vals[n_col1][n_col2] = p_vals[n_col2][n_col1] = round(p, n_round)

    sp_vals = pd.DataFrame(sp_vals, columns=x.columns, index=x.columns)
    p_vals = pd.DataFrame(p_vals, columns=x.columns, index=x.columns)
    return sp_vals, p_vals
```

```
def results_correlation_for_data(data,
                                 names,
                                 wb,
                                 result_graph_path,
                                 n_round=3,
                                 max_p_value=0.05,
                                 min_corr_value=0.3,
                                 engine='circo'):

    for d, name in zip(data, names):
        corr_values, p_values = results_correlation(d)
        make_and_save_graph_correlation(corr_values=corr_values,
                                        p_values=p_values,
                                        filename=result_graph_path.format(name),
                                        max_p_value=max_p_value,
                                        min_corr_value=min_corr_value,
                                        engine=engine)

        wb.add_result_of_correlation_result(corr_values=corr_values,
                                    p_values=p_values,
                                    worksheet_name=name,
                                    n_round=n_round,
                                    max_p_value=max_p_value)
```

Код получение корреляционных графов:

```
from graphviz import Graph
import pandas as pd
from itertools import combinations as comb
from scipy.stats import pearsonr
import numpy as np


def make_and_save_graph_correlation(corr_values,
                                    p_values,
                                    filename='Корреляционный граф',
                                    max_p_value=0.05,
                                    min_corr_value=0.3,
                                    engine='circo'):

    g = Graph('Graph', engine=engine, format='png')
```

```
    for _ in comb(range(corr_values.shape[0]), 2):
        n_col1, n_col2 = _
        name1 = corr_values.columns[n_col1]
        name2 = corr_values.columns[n_col2]

        corr_value = corr_values.iloc[n_col1, n_col2]
        p_value = p_values.iloc[n_col1, n_col2]

        if p_value <= max_p_value and abs(corr_value) >= min_corr_value:
            g.edge(name1, name2, label=str(round(corr_value, 2)))

    g.render(filename=filename, cleanup=True)
```

Код получения результатов регрессионного анализа:

```
from sklearn.linear_model import LinearRegression as LR
import numpy as np
import pandas as pd
from regression.graphics import draw_linear_resgression
import statsmodels.api as sm
from scipy.stats import shapiro


def r2_adj(r2, n, k):
    return 1 - (1 - r2) * (n - 1) / (n - k)

def results_of_regression(y,
                          x,
                          with_ln=False,
                          with_F_statistic=True):


    if x.shape[1] > 1:
        result_columns_names = ["R^2", "R^2(adj)",  "F value",
                                "F p-value",  Shapiro", "Shapiro p-value"]
    else:
        result_columns_names = ["R^2", 'F value', "F p-value",
                                "Shapiro", "Shapiro p-value"]

    coefficients = []
    index = []
    n_rows = y.shape[0]
    columns = y.columns
```

```
        def add_coefficients(x, y, coefficients):
            lr = LR()
            lr.fit(x, y)
            y_pred = lr.predict(x)
            errors = y - y_pred
            shap, shap_p_val = shapiro(errors)
            r2 = lr.score(x, y)
            count_of_coefficients = len(lr.coef_) + 1
            new_data = sm.add_constant(x)
            lin_model = sm.OLS(y, new_data)
            results = lin_model.fit()
            f_value = results.fvalue
            f_pvalue = results.f_pvalue

            if x.shape[1] > 1:
                coefficients.append([r2, r2_adj(r2, n_rows, count_of_coefficients),
                                     f_value, f_pvalue, shap, shap_p_val])
            else:
                coefficients.append([r2, f_value, f_pvalue, shap, shap_p_val])

        for column_name in columns:
            y_ = y[column_name]
            add_coefficients(x, y_, coefficients)
            index.append(column_name)

        if with_ln:
            for column_name in columns:
                y_ = y[column_name]
                temp_ln = np.log1p(y_)
                add_coefficients(x, temp_ln, coefficients)
                index.append("ln({}+1)".format(column_name))
        result = pd.DataFrame(coefficients, index=index,
                              columns=result_columns_names)
        return result


    def regression_results_of_data(y,
                                   x,
                                   x_names,
                                   full_x_names,
                                   with_ln,
                                   draw_function,
                                   wb,
```

```
                    results_graphics_path):

    for (x, x_name, full_x_name, is_ln, is_draw) in zip(x, x_names, full_x_names,
                                                        with_ln, draw_function):
        if is_draw:
            draw_linear_resgression(y, x,
                                    results_graphics_path,
                                    full_y_name=full_x_name,
                                    with_ln=is_ln,
                                    name_of_y='({})'.format(x_name))
        results = results_of_regression(y=y,
                                        x=x,
                                        with_ln=with_ln)
        wb.add_result_of_regression_result(results,
                                   worksheet_name="Лин.рег.({})".format(x_name))
```

Код получения графиков линейной регрессии:

```
from sklearn.linear_model import LinearRegression as _LR
import matplotlib.pyplot as _plt
import numpy as _np

def draw_linear_resgression(x, y,
                            filepath,
                            name_of_y,
                            full_y_name,
                            with_ln=False,
                            size_inches=(13, 6)):
    n_cols = x.shape[1]
    columns = x.columns
    width = size_inches[0]
    height = size_inches[1]
    if with_ln:
        for n_col in range(n_cols):
            temp = x.iloc[:, n_col]
            lr = _LR()
            lr.fit(y, temp)

            temp_ln = _np.log1p(temp)
            lr_ln = _LR()
            lr_ln.fit(y, temp_ln)

            fig, (ax1, ax2) = _plt.subplots(1, 2)
            fig.set_size_inches(width, height)
```

```
            ax1.scatter(y, temp, color='red', marker='x')
            ax1.plot(y, lr.predict(y))
            ax1.set_ylabel(columns[n_col], fontsize=10)
            ax1.set_xlabel(full_y_name, fontsize=10)

            ax2.scatter(y, temp_ln, color='red', marker='x')
            ax2.plot(y, lr_ln.predict(y))
            ax2.set_ylabel("ln({}+1)".format(columns[n_col]), fontsize=10)
            ax2.set_xlabel(full_y_name, fontsize=10)

            _plt.savefig(filepath.format(columns[n_col]+name_of_y))
            _plt.close(fig)
    else:
        for n_col in range(n_cols):
            temp = x.iloc[:, n_col]
            lr = _LR()
            lr.fit(y, temp)

            fig, (ax1, ax2) = _plt.subplots(1, 2)
            fig.set_size_inches(width, height)

            _plt.scatter(y, temp, color='red', marker='x')
            _plt.plot(y, lr.predict(y))
            _plt.ylabel(columns[n_col], fontsize=10)
            _plt.xlabel(full_y_name, fontsize=10)
            _plt.savefig(filepath.format(columns[n_col]+name_of_y))
```

Код получения результатов проверки на нормальность распределения:

```
from scipy.stats import shapiro
import matplotlib.pyplot as plt
import pandas as pd

def shapiro_results_with_histograms(data,
                                    file_path):

    shp_val_pval = []
    columns = ['Shapiro val', 'p_val']
    index = []

    for column_name in data.columns:
        value, p_value = shapiro(data[column_name])
        shp_val_pval.append([value, p_value])
        index.append(column_name)
```

```
        plt.hist(data[column_name], density=True)

        plt.xlabel(column_name, fontsize=10)
        plt.ylabel("Относительная частота", fontsize=10)

        plt.savefig(file_path.format(column_name))
        plt.close()

    return pd.DataFrame(shp_val_pval, columns=columns, index=index)
```

Код создания папок, в которые сохраняются результаты методов анализа:

```
import os

def print_paths(dictionary, path=None):
    if path == None:
        for start in dictionary:
            path = start
            print_paths(dictionary[start], path)
    else:
        if dictionary == None:
            print(path)
        else:
            for file in dictionary:
                new_path = (path, file)
                new_path = '\\'.join(new_path)
                print_paths(dictionary[file], new_path)

class MakeFile:

    _SQUAD_NORM_RESULT = {"Гистограммы": None}
    _SQUAD_CORR_RESULT = {"Корреляция Спирмена":
                                        {'Корреляционные графы': None}}
    _SQUAD_REG_RESULT = {"Графики": None}
    _SQUAD_CLUSTER_RESULT = {"Аггломеративная кластеризация":
                                            {"Дендрограммы": None}}
    _FAMILY_CLUSTER_RESULT = {"Аггломеративная кластеризация":
                                            {"Дендрограммы": None}}

    _RIVER = {"{}. {}": {"Отряды":
{"1. Нормальность распределения": _SQUAD_NORM_RESULT,
"2. Корреляция":  _SQUAD_CORR_RESULT,

 "3. Линейная регрессия": _SQUAD_REG_RESULT,
```

```
 "4. Кластеризация": _SQUAD_CLUSTER_RESULT},
 "Семейства": {"1. Кластеризация":_FAMILY_CLUSTER_RESULT}}}

   _CUSTOM_RIVER = {"Отряды": {"1. Нормальность распределения":
_SQUAD_NORM_RESULT,
                        "2. Корреляция": _SQUAD_CORR_RESULT,
                        "3. Линейная регрессия": _SQUAD_REG_RESULT,
                        "4. Кластеризация": _SQUAD_CLUSTER_RESULT},
                "Семейства": {"1. Кластеризация":
_FAMILY_CLUSTER_RESULT}}

   def __init__(self,
                file_path=None,
                with_results_file=False):

      if file_path is None:
         self.file_path = os.getcwd()
      else:
         self.file_path = file_path

      if with_results_file:
         self.file_path += "\\результаты"
         if not os.path.exists(self.file_path):
            os.mkdir(self.file_path)

   def _create_file(self,
                    dictionary,
                    path):

      if dictionary is not None:
         for file in dictionary:
            new_path = (path, file)
            new_path = '\\'.join(new_path)
            if os.path.exists(new_path):
               self._create_file(dictionary[file],
                                 new_path)
            else:
               os.mkdir(new_path)
               self._create_file(dictionary[file],
                                 new_path)

   def create_file(self):
      self._create_file(self._RIVER,
                        self.file_path)
```

```
    def create_results_file(self,
                            files_name):
        for num, name in enumerate(files_name, 1):
            new_key = '{}. {}'.format(num,
                                      name)
            new_river = {new_key:self._CUSTOM_RIVER}
            self._create_file(new_river,
                              self.file_path)
```

Код программы получения результатов корреляционного анализа:

```
from sklearn.cluster import AgglomerativeClustering as AC
import numpy as _np
from cluster.graphics import draw_dendrogram
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.spatial.distance import pdist
from scipy.cluster.hierarchy import linkage, dendrogram

def clustering_results_of_data(data,
                               description,
                               worksheets_and_graphics_names,
                               labels,
                               wb,
                               result_graphics_path,
                               xlabel='Межкластерное расстояние',
                               ylabel='Номер станции и пробы',
                               start=1,
                               end=6,
                               with_description=True):

    for d, name in zip(data, worksheets_and_graphics_names):

        draw_dendrogram(data=d,
                                labels=labels,
                                xlabel=xlabel,
                                ylabel=ylabel)
        plt.savefig(result_graphics_path.format(name),
                    dpi=300,
                bbox_inches='tight')
        plt.close()
        results = ac_range_results(data=d,
```

```
                                    labels=labels,
                                    start=start,
                                    end=end)
        results = results.astype(np.int)
        results = pd.concat([description, results], axis=1)
        wb.add_clustering_result_to_worksheet(data=results,
                                              worksheet_name=name,
                                              with_description=with_description)

def ac_range_results(data,
                     labels,
                     start=1,
                     end=6):

    end = end + 1
    columns = ["{} cluster".format(i) for i in range(start, end)]
    result = None

    for n_clusters in range(start, end):
        ac = AC(n_clusters=n_clusters)
        ac.fit(data)
        y_ac = ac.labels_
        y_ac = _np.reshape(y_ac, newshape=(-1, 1))
        if n_clusters == start:
            result = _np.array(y_ac)
        else:
            result = _np.hstack((result, y_ac))
    result = pd.DataFrame(result, index=labels, columns=columns)
    return result
```

Код получения результатов в табличном виде:

```
import xlsxwriter as _xlsxw
import pandas as _pd
from mylibrary.correlation import results_correaltion


class MakeXlsx(_xlsxw.workbook.Workbook):

    _BACKGROUND_COLORS = ['#0000ff', '#00ff00', '#ff0000', '#4b0082',
                          '#ffff00', '#ffffff', '#800080', '#f032e6',
                          '#87cefa', '#ffe4b5', '#ff6347']
    _FONTS_COLORS = ['#ffffff', '#ffffff', '#ffffff', '#ffffff',
                     '#ff0000', '#ff0000', '#ffffff', '#ffffff',
```

```
                                '#0000ff', '#ff0000', '#ffffff']
_NEUTRAL_COLOR = "#ffff00"
_BAD_GOOD_BACKGROUND_COLORS = ['#ffc7ce', "#c6efce"]
_BAD_GOOD_FONT_COLORS = ['#9c0006', "#006100"]

def __init__(self, filename='WorkBook.xlsx'):
    self.workbook_name = filename
    self.colors_number = len(self._BACKGROUND_COLORS)
    self.sheets_set = set()
    super().__init__(filename)
    self._init_formats()

def _init_formats(self):
    self._cell_formats = [self.add_format() for i in range(self.colors_number)]
    for color_num, cell in enumerate(self._cell_formats):
        cell.set_bg_color(self._BACKGROUND_COLORS[color_num])
        cell.set_font_color(self._FONTS_COLORS[color_num])
        cell.set_align("center")
        cell.set_border()

    self._index_column_format = self.add_format()
    self._index_column_format.set_align("center")
    self._index_column_format.set_border()
    self._neutral_format = self.add_format()
    self._neutral_format.set_bg_color(self._NEUTRAL_COLOR)
    self._neutral_format.set_border()
    self._neutral_format.set_align("center")
    self._bad_good_formats = [self.add_format() for i in range(2)]
    self._bad_good_formats[0].set_bg_color(
    self._BAD_GOOD_BACKGROUND_COLORS[0])
    self._bad_good_formats[0].set_font_color(
    self._BAD_GOOD_FONT_COLORS[0])
    self._bad_good_formats[0].set_border()
    self._bad_good_formats[0].set_align("center")
    self._bad_good_formats[1].set_bg_color(
    self._BAD_GOOD_BACKGROUND_COLORS[1])
    self._bad_good_formats[1].set_font_color(
    self._BAD_GOOD_FONT_COLORS[1])
    self._bad_good_formats[1].set_border()
    self._bad_good_formats[1].set_align("center")

def _renumber_all_labels(self, data, with_description):
    for n_col in range(with_description, data.shape[1]):
        temp = data.iloc[:, n_col]
```

```
        data.iloc[:, n_col] = self._renumber_one_column(temp.values)

    def _renumber_one_column(self, column):
        from copy import deepcopy
        N, new_labels = len(column), deepcopy(column)
        used = [True]*N
        k = 1

        for i in range(N):
            if used[i]:
                temp = deepcopy(new_labels[i])
                for j in range(i, N):
                    if used[j] and new_labels[j] == temp:
                        used[j] = False
                        new_labels[j] = k
                k += 1

        return new_labels

    def _insert_indexes(self, data,
                                worksheet,
                                n_row, n_col,
                                index_name,
                                column_widths):
        if index_name is not None:
            worksheet.write(n_row, n_col,
                                index_name,
                                self._index_column_format)
            column_widths[n_col] = max(column_widths[n_col], len(str(index_name)))
            intend = 1
        else:
            intend = 0

        for index_num, name_index in enumerate(data.index, intend):
            worksheet.write(n_row+index_num, n_col,
                                name_index,
                                self._index_column_format)

            column_widths[n_col] = max(column_widths[n_col],  len(str(name_index)))


    def _insert_columns(self,
                        data,
                        worksheet,
```

```
                    n_row, n_col,
                    with_description,
                    column_widths):

        for column_num, name in enumerate(data.columns[with_description:], 0):
            worksheet.write(n_row, n_col+column_num,
                            name,
                            self._index_column_format)
            column_widths[n_col+column_num] = (
                                                max(column_widths[n_col + column_num],
                                               len(str(name))))

    def _insert_clustering_data(self, data,
                                worksheet,
                                n_row, n_col,
                                n_indexes, n_columns,
                                with_description,
                                column_widths):

        for row in range(n_indexes):
            for col in range(with_description, n_columns):
                value = data.values[row, col]
                worksheet.write(n_row+row, n_col+col,
                                value,
                                self._cell_formats[value-1])
                column_widths[n_col+col] = (max(column_widths[n_col + col],
                                                len(str(value))))

    def _insert_description(self, data,
                            worksheet,
                            n_row, n_col,
                            column_widths):
        worksheet.write(n_row, n_col,  data.columns[0], self._index_column_format)

        column_widths[n_col] = (max(column_widths[n_col],
                                    len(str(data.columns[0]))))

        for row, description in enumerate(data.iloc[:, 0], 1):
            worksheet.write(n_row+row, n_col,
                            description,
                        self._index_column_format)
            column_widths[n_col] = (max(column_widths[n_col],
                                len(str(description))))
```

```
def _insert_correlation_data(self, corr_values, p_values,
                                    worksheet,
                                    n_row, n_col,
                                    n_indexes, n_columns,
                                    n_round, max_p_value,
                                    column_widths):
    for row in range(n_indexes):
        for col in range(n_columns):
            corr_value = corr_values.values[row, col]
            corr_value = round(corr_value, n_round)

            p_value = p_values.values[row, col]
            p_value = round(p_value, n_round)

            bad_good_index = 1 if p_value <= max_p_value else 0

            worksheet.write(n_row + row, n_col + col,  corr_value,
                            self._bad_good_formats[bad_good_index])

            worksheet.write(n_row + n_indexes + 3 + row, n_col + col, p_value,
                             self._bad_good_formats[bad_good_index])
            column_widths[n_col] = max((column_widths[n_col], len(str(p_value)),
                                        len(str(corr_value))))

    def add_clustering_result_to_worksheet(self,
                                           worksheet_name="sheet1",
                                           data=None,
                                           with_index=True,
                                           index_name="Индексы",
                                           with_description=False):

    n_columns = data.shape[1] + with_index
    n_indexes = data.shape[0]
    column_widths = [0]*n_columns
    self.sheets_set.add(worksheet_name)
    worksheet = self.add_worksheet(worksheet_name)
    self._renumber_all_labels(data, with_description)


    if with_index:
        self._insert_indexes(data, worksheet, 0, 0,
                             index_name,
                             column_widths)
```

```
    if with_description:
        self._insert_description(data, worksheet,
                                 0, int(with_index),
                                 column_widths)

    self._insert_columns(data, worksheet, 0, with_index+with_description,
                         with_description,  column_widths)

    self._insert_clustering_data(data, worksheet, 1, with_index, n_indexes,
                                 n_columns-with_index, with_description,
                                 column_widths)

    for n_col in range(n_columns):
        worksheet.set_column(n_col, n_col, column_widths[n_col]+1)

def add_result_of_correlation_result(self, corr_values, p_values,
                                     worksheet_name="sheet1", n_round=3,
                                     max_p_value=0.05):

    n_columns = corr_values.shape[1] + 1
    n_indexes = p_values.shape[0] + 1

    self.sheets_set.add(worksheet_name)
    worksheet = self.add_worksheet(worksheet_name)
    column_widths = [0]*n_columns
    column_widths[0] = len("Коэффициент корреляции")
    worksheet.write(0, 0, "Коэффициент корреляции", self._neutral_format)
    self._insert_columns(corr_values, worksheet, 0, 1, False, column_widths)
    self._insert_indexes(corr_values, worksheet, 1, 0, None,  column_widths)
    worksheet.write(n_indexes+2, 0, "P-значение", self._neutral_format)
    column_widths[0] = max(column_widths[0], len("P-значение"))
    self._insert_columns(p_values, worksheet, n_indexes+2, 1, False,
                         column_widths)

    self._insert_indexes(p_values, worksheet, n_indexes+3, 0, None,
                         column_widths)

    self._insert_correlation_data(corr_values, p_values, worksheet, 1, 1,
                                  n_indexes-1, n_columns-1, n_round,
                         max_p_value,  column_widths)

    worksheet.write(2*n_indexes+4, 0, '', self._bad_good_formats[0])
    worksheet.merge_range(2*n_indexes+4, 1, 2*n_indexes+4, 5,
                          'p-значение > 0.05')
```

```
    worksheet.write(2*n_indexes+5, 0, '', self._bad_good_formats[1])
    worksheet.merge_range(2*n_indexes+5, 1, 2*n_indexes+5, 5,
                          'p-значение <= 0.05')

    for n_col in range(n_columns):
        worksheet.set_column(0, n_col, column_widths[n_col]+1)

def _insert_regression_data(self, data, worksheet, row_place, col_place,
                            columns_width, n_round=3):

    rows_number = data.shape[0]
    cols_number = data.shape[1]

    if cols_number == 5:
        with_r2_adj = False
    else:
        with_r2_adj = True

    for n_row in range(rows_number):
        r2 = round(data.iloc[n_row][0], n_round)

        if with_r2_adj:
            r2_adj = round(data.iloc[n_row][1], n_round)
            f_value = round(data.iloc[n_row][2], n_round)
            f_pvalue = round(data.iloc[n_row][3], n_round)
            shap = round(data.iloc[n_row][4], n_round)
            shap_p_val = round(data.iloc[n_row][5], n_round)
        else:
            f_value = round(data.iloc[n_row][1], n_round)
            f_pvalue = round(data.iloc[n_row][2], n_round)
            shap = round(data.iloc[n_row][3], n_round)
            shap_p_val = round(data.iloc[n_row][4], n_round)

        if f_pvalue < 0.05 and shap_p_val > 0.05:
            format = self._bad_good_formats[1]
        else:
            format = self._bad_good_formats[0]

        worksheet.write(row_place + n_row, col_place, r2, format)
        if with_r2_adj:
            worksheet.write(row_place + n_row, col_place + 1, r2_adj, format)
            worksheet.write(row_place + n_row, col_place + 2, f_value, format)
            worksheet.write(row_place + n_row, col_place + 3, f_pvalue, format)
```

```
            worksheet.write(row_place + n_row, col_place + 4, shap, format)
            worksheet.write(row_place + n_row, col_place + 5, shap_p_val, format)
        else:
            worksheet.write(row_place + n_row, col_place + 1, f_value, format)
            worksheet.write(row_place + n_row, col_place + 2,  f_pvalue, format)
            worksheet.write(row_place + n_row, col_place + 3, shap, format)
            worksheet.write(row_place + n_row, col_place + 4, shap_p_val, format)

def add_result_of_regression_result(self, data, worksheet_name="sheet1",
                                    n_round=3,  min_r2_value=0.25):
    n_columns = data.shape[1] + 1
    n_index = data.shape[0] + 1

    self.sheets_set.add(worksheet_name)
    worksheet = self.add_worksheet(worksheet_name)
    columns_width = [0] * (n_columns + 1)
    self._insert_indexes(data, worksheet, 0, 0, "", columns_width)
    self._insert_columns(data, worksheet, 0, 1, False, columns_width)
    self._insert_regression_data(data, worksheet, 1, 1, columns_width,  n_round)

    for n_col in range(n_columns):
        worksheet.set_column(0, n_col, columns_width[n_col]+1)

    worksheet.write(n_index+2, 0, '', self._bad_good_formats[0])
    worksheet.merge_range(n_index+2, 1, n_index+2, 4, 'p-value >= 0.5')
    worksheet.write(n_index+3, 0, '', self._bad_good_formats[1])
    worksheet.merge_range(n_index+3, 1, n_index+3, 4, 'p-value < 0.5')

def _insert_normality_check_data(self, worksheet, norm_results, min_p_value,
                                 n_round, columns_width):

    n_rows = norm_results.shape[0]
    for n_row in range(n_rows):
        corr_val = norm_results.iloc[n_row, 0]
        p_val = norm_results.values[n_row, 1]
        corr_val = round(corr_val, n_round)
        p_val = round(p_val, n_round)
        bad_good_index = 0 if min_p_value > p_val else 1
        bad_good_format = self._bad_good_formats[bad_good_index]
        worksheet.write(n_row + 1, 1, corr_val, bad_good_format)
        worksheet.write(n_row + 1, 2, p_val, bad_good_format)
        columns_width[1] = max(columns_width[1], len(str(corr_val)))
        columns_width[2] = max(columns_width[1], len(str(p_val)))
```

```
    def add_result_of_normality_checking_result(self, norm_results,
                                                worksheet_name="sheet1",
                                                n_round=3, min_p_value=0.05):
        n_columns  = norm_results.shape[1] + 1
        n_index = norm_results.shape[0] + 1

        self.sheets_set.add(worksheet_name)
        worksheet = self.add_worksheet(worksheet_name)
        columns_width = [0]*n_columns

        self._insert_columns(norm_results, worksheet, 0, 1, False, columns_width)
        self._insert_indexes(norm_results, worksheet, 0, 0, "", columns_width)
        self._insert_normality_check_data(worksheet, norm_results, min_p_value,
                                          n_round, columns_width)

        for n_col in range(n_columns):
            worksheet.set_column(0, n_col, columns_width[n_col])

        worksheet.write(n_index+2, 0, '', self._bad_good_formats[0])
        worksheet.merge_range(n_index+2, 1,  n_index+2, 7,
                              'R^2 <= {}: отклоняем гипотезу о нормальности
распределения'.format(min_p_value))

        worksheet.write(n_index+3, 0, '', self._bad_good_formats[1])
        worksheet.merge_range(n_index+3, 1, n_index+3, 7,
                              'R^2 > {}:не отклоняем гипотезу о нормальности
распределения'.format(min_p_value))
```

Класс, объединяющий все вышеперечисленные методы:

```
import pandas as pd
from filecreator.filecreator import MakeFile
from excel.makexlsx import  MakeXlsx
from normality.results import shapiro_results_with_histograms
from correlation.results import results_correlation_for_data
from regression.results import regression_results_of_data
import numpy as np
from cluster.results import clustering_results_of_data

class Res(MakeFile):

    def __init__(self, file_path=None, with_results=True):
        if file_path is None:
            file_path = os.getcwd()
```

```
    MakeFile.__init__(self, file_path=file_path, with_results_file=with_results)

def make_files(self, file_names):
    self.river_names=file_names
    MakeFile.create_results_file(self, files_name=self.river_names)

def fit_data(self, filepath_result, file_names_squads, file_names_family):
    self.squad = dict()
    self.family = dict()
    for num, data_name in enumerate(file_names_squads, 1):
        key = None
        for river_name in self.river_names:
            if river_name in data_name and data_name[len(river_name)] == ' ':
                key = river_name
                break

        if key is None:
            print("Нету такого реки в названии: {}".format(data_name))
        else:
            result = self.file_path + os.sep + '{}. {}'.format(num, key)
            data = filepath_result + os.sep + data_name
            if 'squad' in data_name:
                self.squad[key] = {"result": result,
                                   "data": data}
            if 'family' in data_name:
                self.family[key] = {"result": result,
                                    "data": data}

    for num, data_name in enumerate(file_names_family, 1):
        key = None
        for river_name in self.river_names:
            if river_name in data_name and data_name[len(river_name)] == ' ':
                key = river_name
                break

        if key is None:
            print("Нету такого реки в названии: {}".format(data_name))
        else:
            result = self.file_path + os.sep + '{}. {}'.format(num, key)
            data = filepath_result + os.sep + data_name
            if 'squad' in data_name:
                self.squad[key] = {"result": result, "data": data}
            if 'family' in data_name:
                self.family[key] = {"result": result, "data": data}
```

```
    print(self.squad)
    print(self.family)

def _make_nbbiotic_columns(self, biotic_columns):
    self.b_biotic_columns = []
    self.n_biotic_columns = []
    for col_name in biotic_columns:
        if "n-" in col_name:
            self.n_biotic_columns.append(col_name)
        elif "b-" in col_name:
            self.b_biotic_columns.append(col_name)
        else:
            print("Warning: неизвестный столбец: {}".format(col_name))

def _check_normality(self):
    check_data = [self.b_biotic_columns, self.n_biotic_columns,
                  self.abiotic_columns]

    for key in self.squad:
        results_path = self.squad[key]["result"]
        data_path = self.squad[key]["data"]
        temp_data = pd.read_csv(data_path)

        results_xlsx = os.sep.join((results_path, "Отряды",
                                    "1. Нормальность распределения",
                                    "({}) Результаты нормальности
                                    распределения.xlsx".format(key)))
        wb = MakeXlsx(results_xlsx)
        filepath_hist = os.sep.join((results_path, 'Отряды',
                                     "1. Нормальность распределения",
                                     "Гистограммы", "{}"))
        to2_columns = ["t", "o2"]
        to2_data = temp_data[to2_columns]
        to2_shapiro = shapiro_results_with_histograms(to2_data, filepath_hist)

        n_biotic_data = temp_data[self.n_biotic_columns]
        n_biotic_shapiro = shapiro_results_with_histograms(n_biotic_data,
                                                           filepath_hist)

        b_biotic_data = temp_data[self.b_biotic_columns]
        b_biotic_results = shapiro_results_with_histograms(b_biotic_data,
                                                           filepath_hist)
```

```
        wb.add_result_of_normality_checking_result(to2_shapiro,
                                        worksheet_name="Температура и кислород")
        wb.add_result_of_normality_checking_result(n_biotic_shapiro,
                                        worksheet_name='Численность')
        wb.add_result_of_normality_checking_result(b_biotic_results,
                                        worksheet_name='Биомасса')
        wb.close()

def _make_correlation(self):
    part_func = lambda s: 'Доля ({})'.format(s)

    for key in self.squad:
        results_path = self.squad[key]['result']
        results_xlsx = os.sep.join((results_path, "Отряды", "2. Корреляция",
                                    "Корреляция Спирмена",
                                    '({}) Результаты корреляции.xlsx'.format(key)))
        results_graph = os.sep.join((results_path, "Отряды", "2. Корреляция",
                                    "Корреляция Спирмена",
                                    "Корреляционные графы", "{}"))

        data_path = self.squad[key]["data"]
        temp_data = pd.read_csv(data_path)
        n_bio = temp_data[self.n_biotic_columns]
        parts_n_bio = n_bio.div(n_bio.sum(axis=1), axis=0)
        parts_n_bio.columns = list(map(part_func, parts_n_bio.columns))
        b_bio = temp_data[self.b_biotic_columns]
        parts_b_bio = b_bio.div(b_bio.sum(axis=1), axis=0)
        parts_b_bio.columns = list(map(part_func, parts_b_bio.columns))
        datas = [n_bio, parts_n_bio, b_bio, parts_b_bio]
        names = ["Численнность", "Доли численности", "Биомасса",
                 "Доли биомассы"]
        wb = MakeXlsx(results_xlsx)

        results_correlation_for_data(datas, names, wb, results_graph)
        wb.close()




def _make_regression(self):
    for key in self.squad:
        results_path = self.squad[key]["result"]
        data_path = self.squad[key]["data"]
        temp_data = pd.read_csv(data_path)
```

```
        results_xlsx = os.sep.join((results_path, "Отряды",
                                    "3. Линейная регрессия",
                                    "({}) Результаты линейной
                                     регрессии.xlsx".format(key)))
        results_graphics = os.sep.join((results_path, "Отряды",
                                        "3. Линейная регрессия",
                                        "Графики", '{}'))
        T = pd.DataFrame(np.reshape(temp_data["t"].values, newshape=(-1, 1)),
                                       columns=['t'])
        O2 = pd.DataFrame(np.reshape(temp_data["o2"].values, newshape=(-1, 1)),
                                        columns=['o2'])
        TO2 = temp_data[['t', "o2"]]
        x = temp_data[self.n_biotic_columns + self.b_biotic_columns]
        predictors = [T, O2, TO2]
        predictors_names = ["t", "o2", "t+o2"]
        full_pred_name = ["Температура", 'Кислород', "Темп.+Кисл."]
        with_ln = [True, True, False]
        draw_regression = [True, True, False]
        wb = MakeXlsx(results_xlsx)

        regression_results_of_data(x, predictors, predictors_names, full_pred_name,
                           with_ln, draw_regression, wb, results_graphics)
        wb.close()

    def _make_cluster(self):
        for key in self.squad:
            result_path = self.squad[key]['result']
            data_path = self.squad[key]['data']
            temp_data = pd.read_csv(data_path)

            results_dendrogramm = os.sep.join((result_path, "Отряды",
                                               "4. Кластеризация",
                                               "Аггломеративная кластеризация",
                                               "Дендрограммы", "{}"))
            results_xlsx = os.sep.join((result_path, "Отряды", "4. Кластеризация",
                                "Аггломеративная кластеризация",
                                "({}) Результаты кластеризации.xlsx".format(key)))
            wb = MakeXlsx(results_xlsx)
            labels = temp_data["station"].values
            description = temp_data["description"]
            description.index = labels
            n_bio = temp_data[self.n_biotic_columns]
            n_bio_ln = np.log1p(n_bio)
```

```
    n_bio_parts = n_bio.div(n_bio.sum(axis=1), axis=0)
    b_bio = temp_data[self.b_biotic_columns]
    b_bio_ln = np.log1p(b_bio)
    b_bio_parts = b_bio.div(b_bio.sum(axis=1), axis=0)
    data = [n_bio, n_bio_ln, n_bio_parts,
            b_bio, b_bio_ln, b_bio_parts]

    worksheets_and_graphics_names = ["Численность",
                                     "log1p(Численность)",
                                     "Доли(Численность)",
                                     "Биомасса", "log1p(Биомасса)",
                                     "Доли(Биомасса)"]
    clustering_results_of_data(data, description,
                               worksheets_and_graphics_names,
                               labels, wb, results_dendrogramm)
    wb.close()

for key in self.family:
    result_path = self.family[key]['result']
    data_path = self.family[key]['data']
    temp_data = pd.read_csv(data_path)

    results_dendrogramm = os.sep.join((result_path, "Семейства",
                                       "1. Кластеризация",
                            "Аггломеративная кластеризация",
                            "Дендрограммы", "{}"))

    results_xlsx = os.sep.join((result_path, "Семейства",
                        "1. Кластеризация",
                        "Аггломеративная кластеризация",
                        "({}) Результаты кластеризации.xlsx".format(key)))

    wb = MakeXlsx(results_xlsx)
    labels = temp_data["station"].values
    description = temp_data["description"]
    description.index = labels


    n_bio = temp_data.copy()
    for column in self.other_columns + ['ground']:
        del n_bio[column]

    n_bio_ln = np.log1p(n_bio)
    n_bio_parts = n_bio.div(n_bio.sum(axis=1), axis=0)
```

```
        data = [n_bio, n_bio_ln, n_bio_parts]

        worksheets_and_graphics_names = ["Численность",
                                         "log1p(Численность)",
                                         "Доли(Численность)",]
        clustering_results_of_data(data, description,
                           worksheets_and_graphics_names,
                           labels, wb, results_dendrogramm)
        wb.close()

def make_results(self, biotic_columns, abiotic_columns, other_columns):

    self._make_nbbiotic_columns(biotic_columns)
    self.abiotic_columns = abiotic_columns
    self.other_columns = other_columns
    self._check_normality()
    self._make_correlation()
    self._make_regression()
    self._make_cluster()
```

Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и фундаментальной информатики
Базовая кафедра вычислительных и информационных технологий

УТВЕРЖДАЮ
Заведующий кафедрой
_____ / В.В. Шайдуров

«17» июня 2019 г.

БАКАЛАВРСКАЯ РАБОТА

Направление 02.03.01 Математика и компьютерные науки

СТАТИСТИЧЕСКИЙ АНАЛИЗ ДАННЫХ ДЛЯ ИССЛЕДОВАНИЯ СТРУКТУРНОЙ ОРГАНИЗАЦИИ И ПРОСТРАНСТВЕННОЙ ДИНАМИКИ СООБЩЕСТВ ЗООБЕНТОСА В РЕКАХ БАССЕЙНА ЕНИСЕЯ

Научный руководитель
кандидат физико-математических наук,
доцент _____ / Е.Д. Карепова
17.06.19

Выпускник _____ / М.П. Лепьявко
17.06.19

Красноярск 2019