

Правительство Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
(НИУ ВШЭ)

Московский институт электроники и математики им. А.Н. Тихонова

ПРАКТИЧЕСКАЯ РАБОТА № 10

ТЕМА РАБОТЫ
«Расчет географических расстояний»

Москва, 2025

Цель работы.....	2
Целевая аудитория.....	3
Идея и концепция.....	3
Содержание практической работы.....	4
О наборе данных и задаче работы.....	4
Работа с данными.....	4
Загрузка и подготовка исходных данных.....	4
Расчет расстояний между координатами.....	6
Анализ результатов (суммарное расстояние, максимумы).....	9
Анализ отдельных сегментов.....	11
Проверка результатов и выводы.....	12
Визуализация результатов.....	12
Построение диаграммы расстояний сегментов.....	12
Анализ диаграммы.....	14
Фоновая картограмма (Choropleth map).....	15
Пример выполненной работы.....	15
Приобретенные навыки.....	16
Обобщенная задача для выполнения индивидуального варианта.....	17
Распределение вариантов.....	18

Цель работы

Изучить способы расчета расстояний между географическими координатами и применение этих расчетов в анализе данных с помощью Altair AI Studio. В ходе работы студенты:

- Ознакомятся с форматом географических данных (координатами широты и долготы) и узнают, как их использовать в аналитических задачах.
- Научатся применять оператор Generate Attributes в RapidMiner для вычисления новых показателей по формуле (на примере формулы расчета расстояния между двумя координатами).
- Реализуют вычисление географического расстояния (в километрах) между точками на основе формулы гаверсинусов (Haversine) или схожей модели, полностью внутри RapidMiner.

- Используют полученные расстояния для практического анализа – например, расчета общей длины маршрута, выявления самого длинного отрезка, сравнения дистанций – и сделают выводы на основе результатов.

Целевая аудитория

Практическая работа рассчитана на студентов, интересующихся анализом данных и его приложениями в географии, логистике или картографических задачах. Аудитория уже знакома с основами работы в RapidMiner (импорт данных, базовые операции) и хочет получить навык интеграции математических формул в процесс анализа данных без необходимости отдельно программировать расчеты.

Идея и концепция

Концепция работы – показать, как данные из реального мира (географические координаты городов или точек маршрута) можно анализировать с помощью инструментов Data Science. В качестве основы берется практический сценарий: расчет расстояния путешествия через несколько городов. Например, рассматривается маршрут через крупные города России (от Москвы до Владивостока). Каждый студент самостоятельно:

- Загрузит предоставленный набор данных с координатами точек маршрута (широта и долгота для последовательности городов).
- Подготовит данные для вычислений (убедится в корректности формата координат, при необходимости преобразует их).
- Настроит вычисление расстояний между заданными точками, используя формулы внутри RapidMiner (без внешних картографических сервисов).
- Применит рассчитанные расстояния для анализа маршрута: найдет общую пройденную дистанцию, определит самый длинный промежуток между остановками, визуализирует распределение расстояний между сегментами маршрута.

В итоге студенты увидят, как на практике можно работать с геоданными в Altair AI Studio, и поймут, как результаты расчета расстояний помогают в реальных задачах (например, оценить дальность маршрута, планировать логистику и пр.).

Содержание практической работы

О наборе данных и задаче работы

Набор данных: Координаты основных точек маршрута через территорию России. Для примера выбран маршрут от Москвы до Владивостока с остановками в крупных городах по пути. Данные представлены в виде таблицы, где каждая строка – отрезок пути (сегмент маршрута) между двумя городами. Столбцы таблицы: StartCity (название начального города сегмента), StartLat (широта начальной точки), StartLon (долгота начальной точки), EndCity (конечный город сегмента), EndLat (широта конечной точки), EndLon (долгота конечной точки). Всего рассматривается, например, 6 сегментов маршрута: Москва → Нижний Новгород, Нижний Новгород → Екатеринбург, Екатеринбург → Новосибирск, Новосибирск → Иркутск, Иркутск → Хабаровск, Хабаровск → Владивосток. Координаты широты/долготы заданы в десятичных градусах (WGS84).

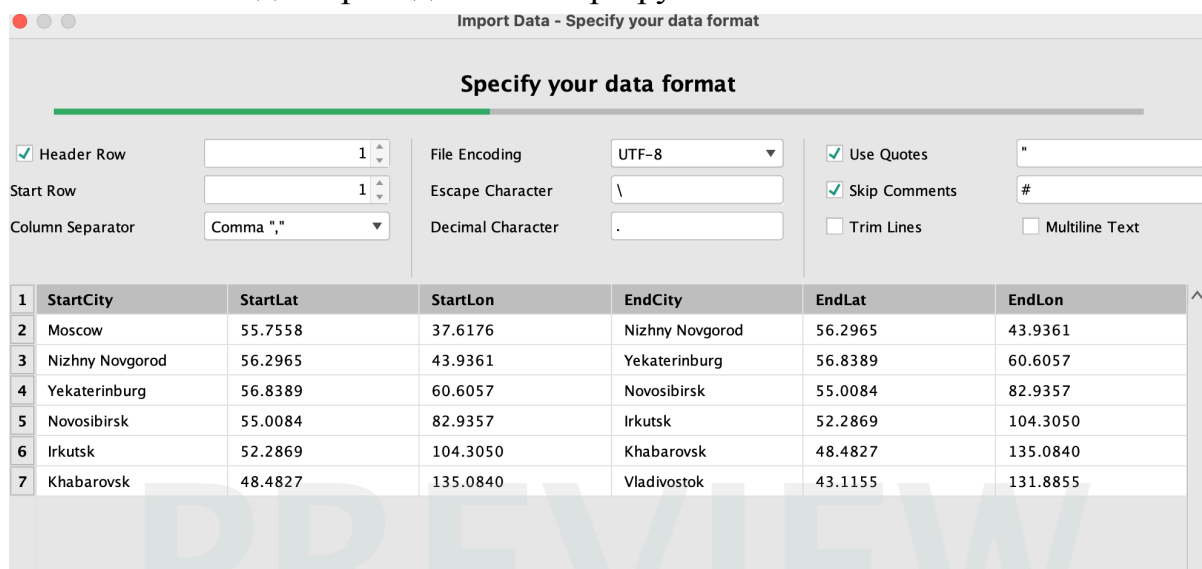
Задача: Собрать данные новостной ленты и проанализировать текст новостей, чтобы определить наиболее часто упоминаемые слова и темы. Это позволит выявить, какие темы находятся в центре внимания в выбранный период. Необходимо продемонстрировать процесс веб-скрейпинга, очистки текста, расчет частот слов и визуализацию результатов в RapidMiner (Altair AI Studio).

Работа с данными

Загрузка и подготовка исходных данных

- 1) Импорт данных с координатами:
 - Используйте оператор Read CSV для загрузки таблицы с координатами маршрута (например, файл route.csv).
 - Перетащите Read CSV на рабочую область и в его параметрах выберите файл с данными.

- Убедитесь, что данные считаны правильно: должно быть 6 строк (по числу сегментов) и 6 столбцов (StartCity, StartLat, StartLon, EndCity, EndLat, EndLon).
- Проверьте корректность: широты в России ~43–56°, долготы ~37–135° для приведенного маршрута.



	StartCity	StartLat	StartLon	EndCity	EndLat	EndLon
1	Moscow	55.7558	37.6176	Nizhny Novgorod	56.2965	43.9361
2	Nizhny Novgorod	56.2965	43.9361	Yekaterinburg	56.8389	60.6057
3	Yekaterinburg	56.8389	60.6057	Novosibirsk	55.0084	82.9357
4	Novosibirsk	55.0084	82.9357	Irkutsk	52.2869	104.3050
5	Irkutsk	52.2869	104.3050	Khabarovsk	48.4827	135.0840
6	Khabarovsk	48.4827	135.0840	Vladivostok	43.1155	131.8855

рис.1: Предварительный просмотр данных при загрузке

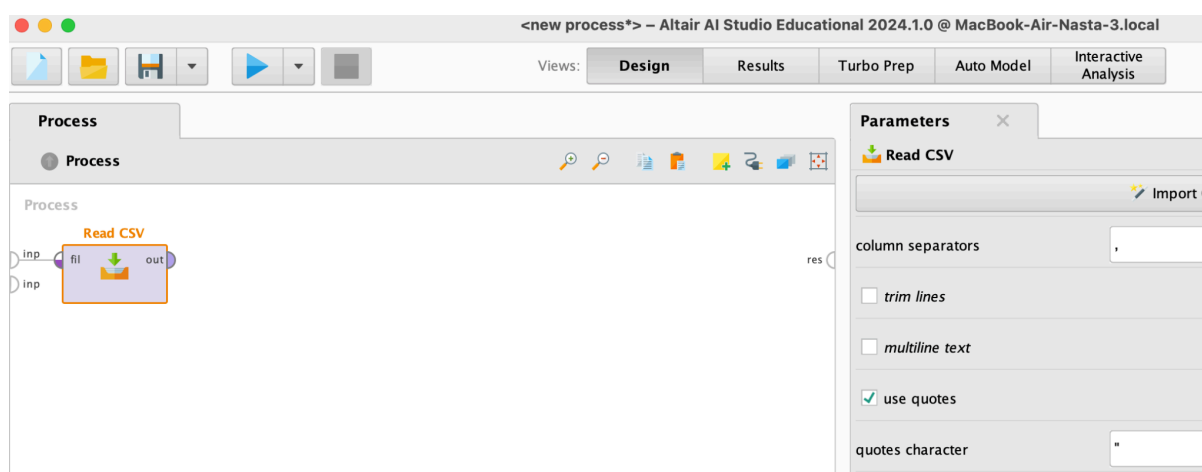


рис.2: Оператор Read CSV на панели процесса

2) Проверка типов данных:

- Нажмите правой кнопкой на выходном порте Read CSV и выберите Show Data.
- Проверьте, что столбцы с широтой и долготой распознаны как числовые (real). Если по каким-то причинам координаты считались как текст (polynomial), их нужно преобразовать.
- В RapidMiner обычно CSV с цифрами распознается корректно, но если нет – можно добавить оператор Type Conversion или прямо в

Read CSV указать тип столбцов. Все координаты должны быть числовыми для дальнейших расчетов.

	StartCity <i>polynomial</i>	StartLat <i>real</i>	StartLon <i>real</i>	EndCity	EndLat	EndLon <i>real</i>
1	Moscow	55.756	37.618			
2	Nizhny Novgorod	56.297	43.936		297	43.936
3	Yekaterinburg	56.839	60.606		839	60.606
4	Novosibirsk	55.008	82.936	Irkutsk	008	82.936
5	Irkutsk	52.287	104.305	Khabarovsk	287	104.305
6	Khabarovsk	48.483	135.084	Vladivostok	483	135.084
					115	131.886

рис.3: Проверка и изменение (при необходимости) типов переменных при просмотре данных в панели Show Data

Расчет расстояний между координатами

1) Настройка вычисления расстояния:

- Расстояние между двумя точками на Земле по широте/долготе можно вычислить по формуле гаверсинов. В RapidMiner мы реализуем эту формулу через оператор Generate Attributes. Найдите Generate Attributes в панели операторов и подключите его после чтения данных. Этот оператор позволит создать новые столбцы по заданным формулам.

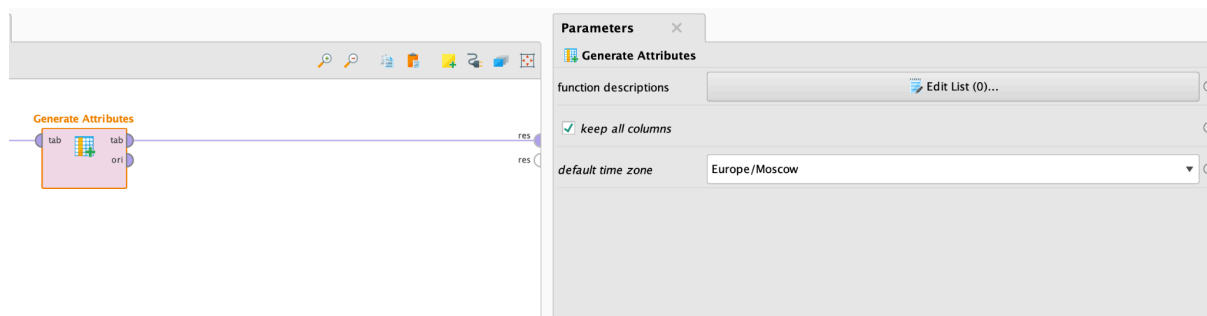


рис.4: Подключение оператора Generate Attributes

2) Преобразование градусов в радианы:

- Внутри Generate Attributes сначала создадим вспомогательные атрибуты – широта/долгота в радианах, поскольку тригонометрические функции ожидают угол в радианах.
- В Generate Attributes каждая новая формула задается отдельной строкой. В RapidMiner (Altair AI Studio) это делается так:

- Откройте оператор Generate Attributes (щелчком или двойным кликом).
- В правой части, в параметрах, найдите поле function descriptions и нажмите Edit List.
- В открывшемся окне вы увидите список формул. Чтобы добавить новую формулу, нажмите Add Entry. Здесь 3.1416/180 – коэффициент для перевода градусов в радианы ($\pi/180$). Новые столбцы *lat1_rad*, *lon1_rad*, *lat2_rad*, *lon2_rad* появятся в данных.

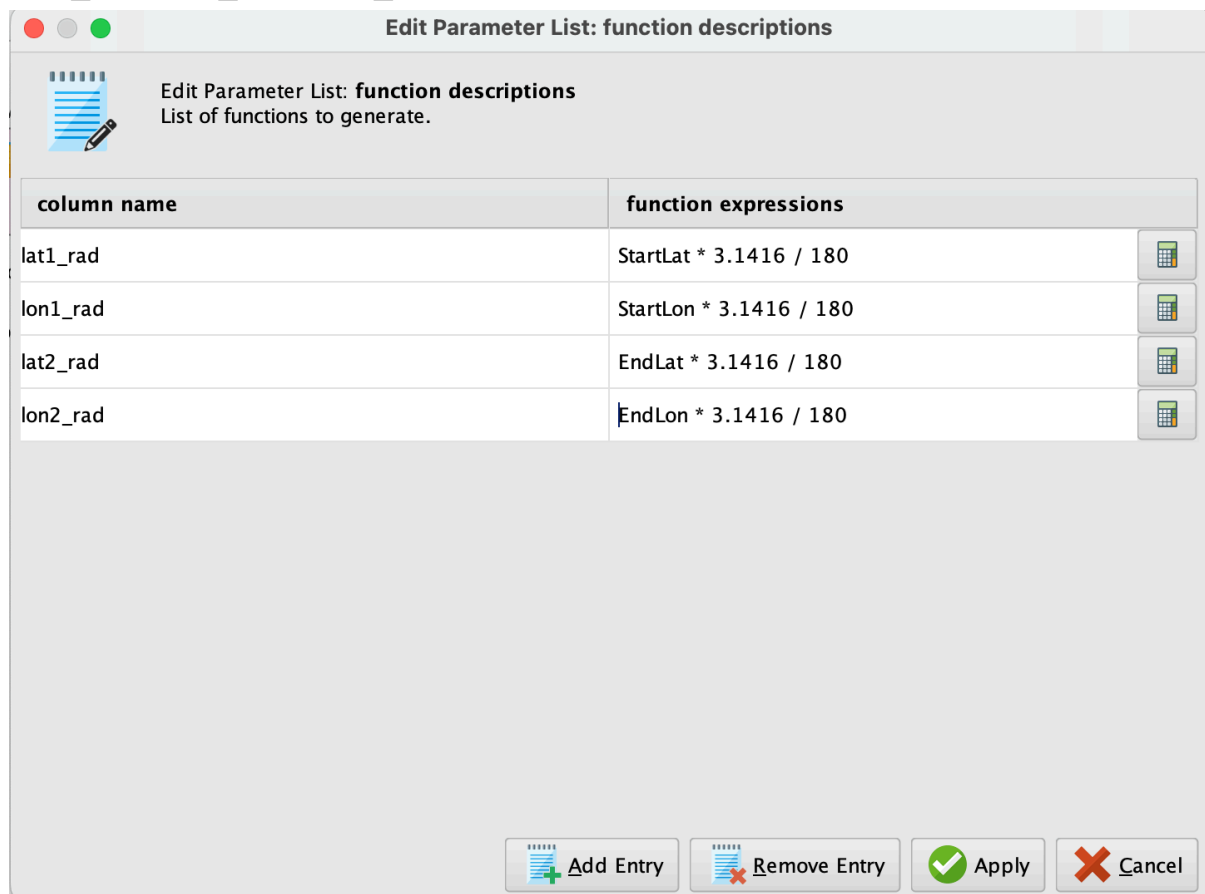


рис.5: Настройка расширенных параметров оператора Generate Attributes, создание новых столбцов на основе математических формул для перевода градусов в радианы

3) Вычисление расстояния (км):

- Далее, все в том же Generate Attributes, добавьте выражение для расчета расстояния по формуле. В Edit Parameter List задайте:
column name: distance_km

function expressions:

$$distance_km = 6371 * acos(sin(lat1_rad) * sin(lat2_rad) + cos(lat1_rad) * cos(lat2_rad) * cos(lon2_rad - lon1_rad)).$$

Здесь 6371 – средний радиус Земли в километрах. Функции sin и cos

берут аргументы в радианах, что обеспечено предыдущим шагом. В результате вычисляется `distance_km` – геодезическое расстояние между начальной и конечной точкой сегмента (по дуге большого круга).

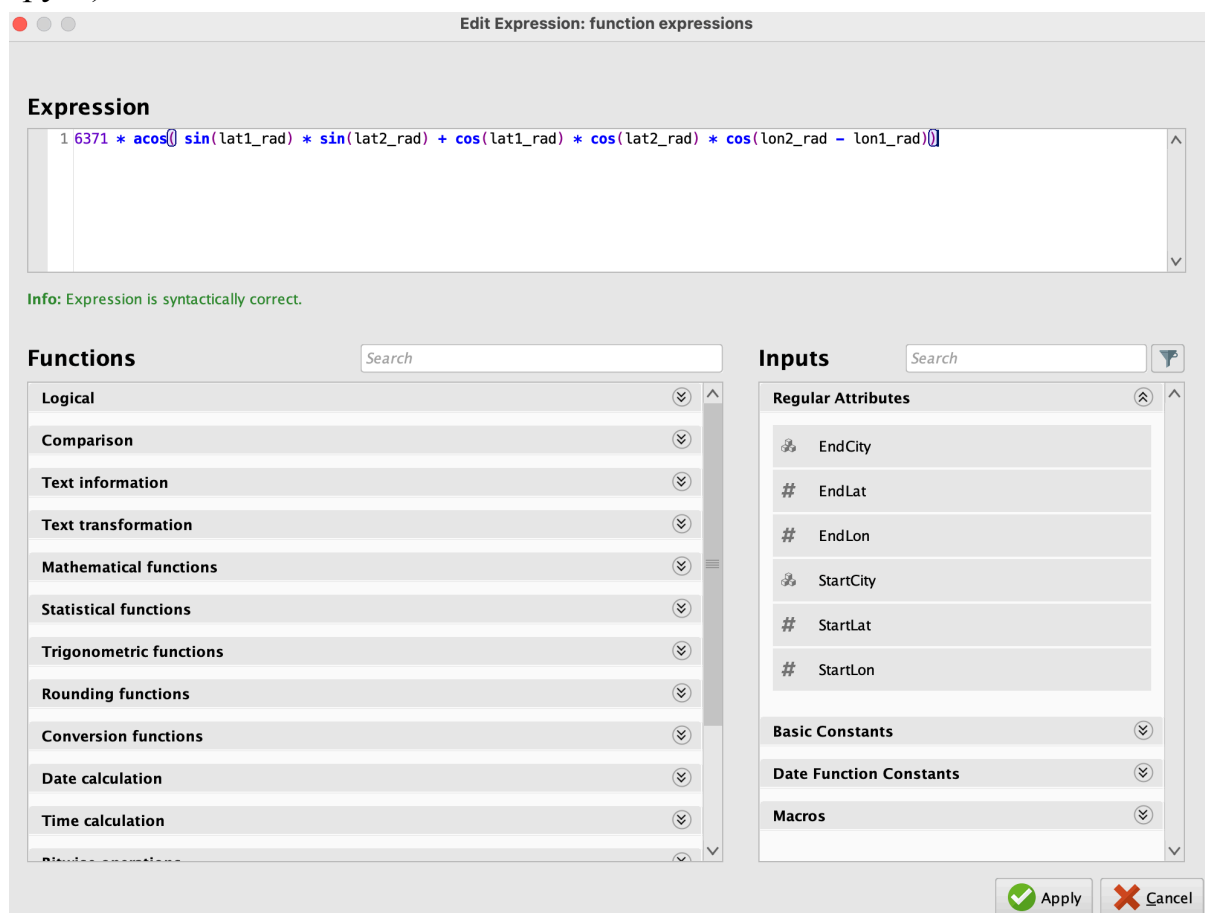
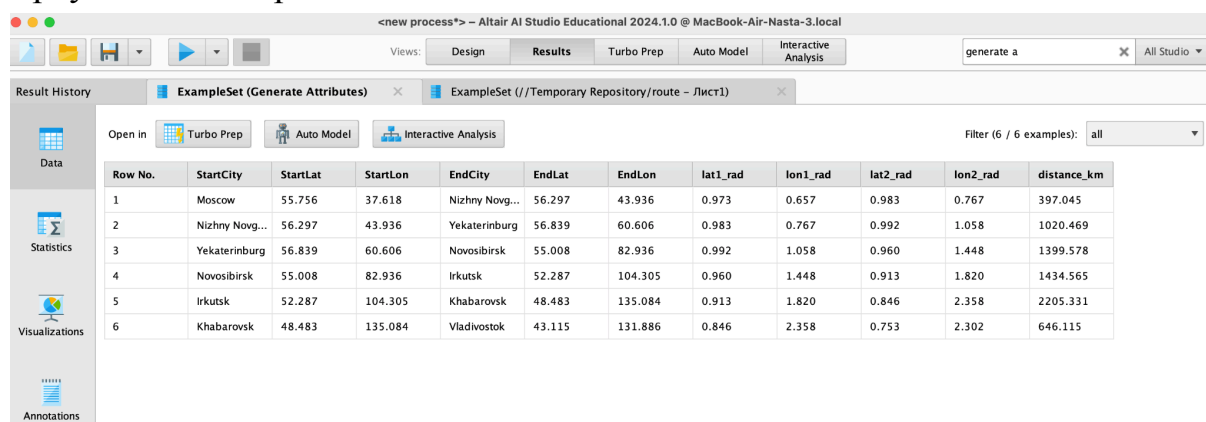


рис.6: Настройка расширенных параметров оператора *Generate Attributes*, создание нового столбца (*distance_km*) для расчета геодезического расстояния между начальной и конечной точкой сегмента маршрута

-
- 4) Применение оператора и просмотр результатов:
 - Запустите процесс (Run). На выходе оператора *Generate Attributes* теперь каждый сегмент маршрута дополнен вычисленным полем `distance_km`.
 - Подключите выход *Generate Attributes* к *result*-порту и выполните процесс, если еще не сделали этого.
 - Откройте полученную таблицу сегментов и убедитесь, что у каждого есть число расстояния (в километрах). Например, для сегмента Москва – Нижний Новгород ожидается порядка 400 км, а самые длинные сегменты (на Дальнем Востоке) могут превышать 2000 км. Если результаты выглядят реалистично, расчет выполнен верно. (Важно: маленькие погрешности возможны из-за сферической

модели Земли). В нашем примере:
Москва → Нижний Новгород ≈ 400 км,
Иркутск → Хабаровск свыше 2000 км



The screenshot shows the Altair AI Studio interface. The main window displays a data table with 12 columns: Row No., StartCity, StartLat, StartLon, EndCity, EndLat, EndLon, lat1_rad, lon1_rad, lat2_rad, lon2_rad, and distance_km. The table contains 6 rows of data representing a route from Moscow to Vladivostok via several intermediate cities. The 'distance_km' column shows the distance between consecutive cities.

Row No.	StartCity	StartLat	StartLon	EndCity	EndLat	EndLon	lat1_rad	lon1_rad	lat2_rad	lon2_rad	distance_km
1	Moscow	55.756	37.618	Nizhny Novg...	56.297	43.936	0.973	0.657	0.983	0.767	397.045
2	Nizhny Novg...	56.297	43.936	Yekaterinburg	56.839	60.606	0.983	0.767	0.992	1.058	1020.469
3	Yekaterinburg	56.839	60.606	Novosibirsk	55.008	82.936	0.992	1.058	0.960	1.448	1399.578
4	Novosibirsk	55.008	82.936	Irkutsk	52.287	104.305	0.960	1.448	0.913	1.820	1434.565
5	Irkutsk	52.287	104.305	Khabarovsk	48.483	135.084	0.913	1.820	0.846	2.358	2205.331
6	Khabarovsk	48.483	135.084	Vladivostok	43.115	131.886	0.846	2.358	0.753	2.302	646.115

рис.7: Просмотр преобразованного набора данных после запуска процесса, в который включен оператор *Generate Attributes*, в столбце *distance_km* – расстояния между городами (*StartCity* – *EndCity*)

Анализ результатов (суммарное расстояние, максимумы)

Суммарная длина маршрута: добавьте оператор *Aggregate* для подсчета общих статистик по расстояниям. Перетащите *Aggregate* на поле процесса и подключите его после *Generate Attributes*. В настройках *Aggregate* не указывайте атрибут группировки (оставьте группировку пустой, чтобы агрегировать по всему набору сразу) и не используйте «use default aggregation». Нажмите кнопку *Edit List*, в появившемся окне добавьте две записи (кнопка *Add Entry*):

- aggregation attribute: *distance_km*, aggregation functions: *sum*
- aggregation attribute: *distance_km*, aggregation functions: *maximum*
- Нажмите *Apply* и вернитесь к параметрам оператора.

Это настроит расчет суммы всех значений расстояний (итоговый маршрут) и максимального значения. Запустите процесс. На выходе *Aggregate* получится одна строка с двумя новыми столбцами: общая протяженность маршрута, самое большое расстояние среди сегментов. Например, суммарно весь маршрут Москва–Владивосток (через выбранные остановки) может получиться около ~7100 км, а максимальный сегмент ~2200 км.

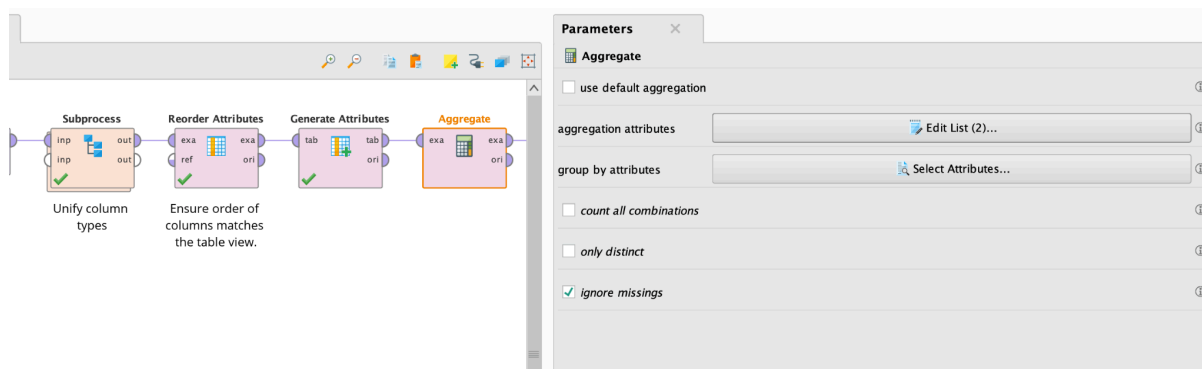


рис.8: Добавление оператора Aggregate в исходных процесс на панель

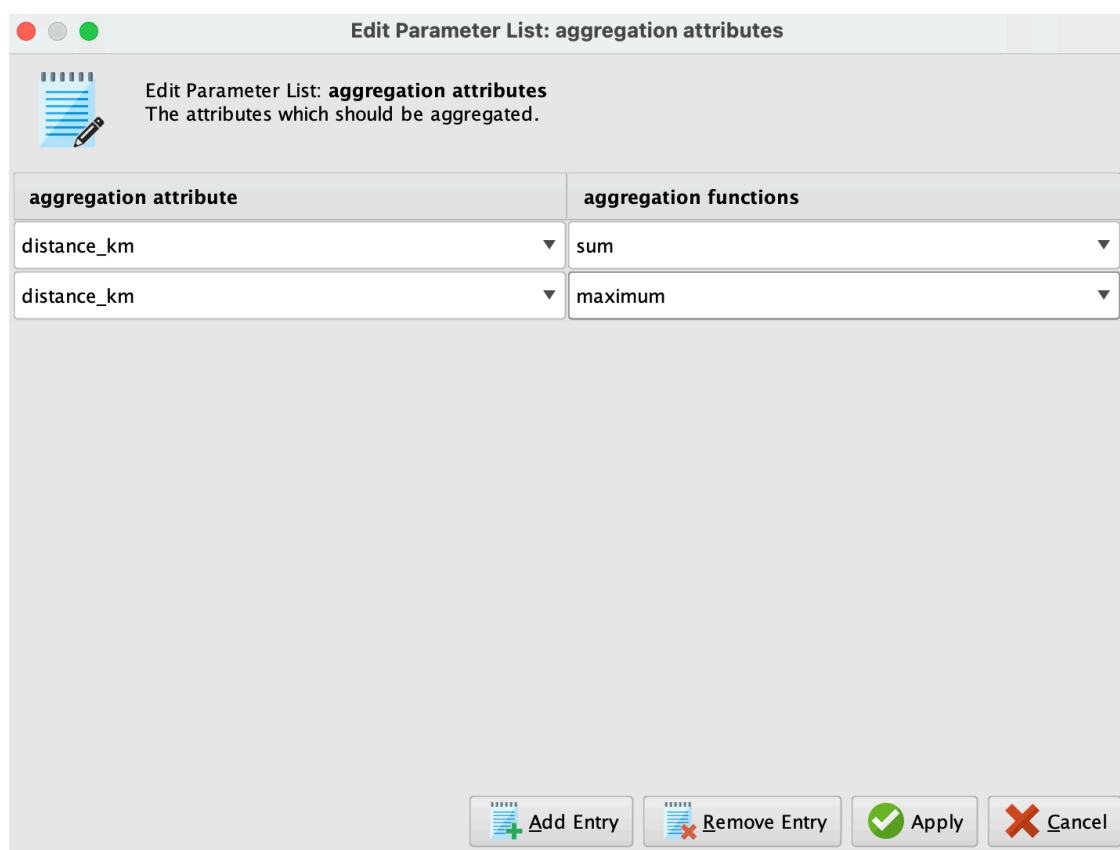




рис.9: Настройка расширенных параметров оператора Aggregate, включающая агрегацию distance_km по суммарной функции (протяженность маршрута) и функции поиска максимума (самый длинный маршрут между двумя точками)

Result History




Data




Statistics

ExampleSet (Aggregate) ✕

Open in



Turbo Prep



Auto Model

Row No.	sum(distan...	maximum(...
1	7103.103	2205.331

рис.10: Результат запуска процесса после включения оператора Aggregate, получено два значения – суммарная протяженность маршрута и максимальная дистанция между городами

Анализ отдельных сегментов

Вернитесь к таблице со всеми сегментами и сравните расстояния. Можно отсортировать таблицу по столбцу distance_km (по убыванию) для наглядности. Самый верхний в списке будет самый протяженный отрезок пути. Например, согласно расчетам, сегмент Иркутск – Хабаровск может оказаться самым длинным (~2200 км). Самый короткий – вероятно, Москва – Нижний Новгород (~400 км). Сопоставьте эти результаты с географией маршрута: логично, что наибольшие расстояния между городами на Дальнем Востоке, где города расположены далеко друг от друга.

Open in [Turbo Prep](#) [Auto Model](#) [Interactive Analysis](#) Filter (6 / 6 examples): [all](#)

Row No.	StartCity	StartLat	StartLon	EndCity	EndLat	EndLon	lat1_rad	lon1_rad	lat2_rad	lon2_rad	distanc... ↓
5	Irkutsk	52.287	104.305	Khabarovsk	48.483	135.084	0.913	1.820	0.846	2.358	2205.331
4	Novosibirsk	55.008	82.936	Irkutsk	52.287	104.305	0.960	1.448	0.913	1.820	1434.565
3	Yekaterinburg	56.839	60.606	Novosibirsk	55.008	82.936	0.992	1.058	0.960	1.448	1399.578
2	Nizhny Novg...	56.297	43.936	Yekaterinburg	56.839	60.606	0.983	0.767	0.992	1.058	1020.469
6	Khabarovsk	48.483	135.084	Vladivostok	43.115	131.886	0.846	2.358	0.753	2.302	646.115
1	Moscow	55.756	37.618	Nizhny Novg...	56.297	43.936	0.973	0.657	0.983	0.767	397.045

рис. 11: Сортировка данных по убыванию по параметру `distance_km` для отображения маршрутного километража между точками

Важно: такой способ сортирует данные только в окне просмотра. При этом сам набор данных в процессе не изменяется, и если вы используете эти данные дальше, они могут не быть отсортированными.

Проверка результатов и выводы

Используя полученные значения, сделайте вывод о маршруте: какова его общая длина, сколько в среднем составляет один перегон между городами, какие части пути наиболее длинные или, наоборот, незначительные по расстоянию. Например: «Общая протяженность маршрута составляет около 7100 км. Самый длинный перегон – между Иркутском и Хабаровском (~2200 км), самый короткий – между Москвой и Нижним Новгородом (~400 км). Большая часть сегментов имеет длину около 1000–1500 км.» Такие выводы могут быть полезны, например, для понимания, где путешественнику предстоит самый длинный безостановочный переезд.

Визуализация результатов

Построение диаграммы расстояний сегментов

Для наглядного представления различий в расстояниях можно воспользоваться визуализацией:

- Перейдите во вкладку **Charts** результатов процесса (после вычисления расстояний для всех сегментов, до агрегирования).
- Выберите тип графика **Bar Chart**.

- По оси X задайте категорию – можно использовать названия сегментов. Проще всего создать категорию сегмента объединением названий городов: вернитесь в процесс и добавьте оператор Generate Attributes перед визуализацией (или используйте тот же, что уже есть) с выражением, например:

В поле column name введите: *segment_name*,

В поле function expressions введите выражение: *concat(StartCity, " - ", EndCity)*

Это соединит название начального и конечного города в одну строку (например, "Москва - Нижний Новгород").

- Теперь в Charts выберите по оси X атрибут *segment_name*, а по оси Y – *distance_km*. Вы получите столбчатый график, на котором каждый столбец – длина соответствующего отрезка маршрута.
- Таким образом, вы получите наглядное отображение отрезков пути и сможете быстро сравнить, какой сегмент маршрута самый длинный или короткий.

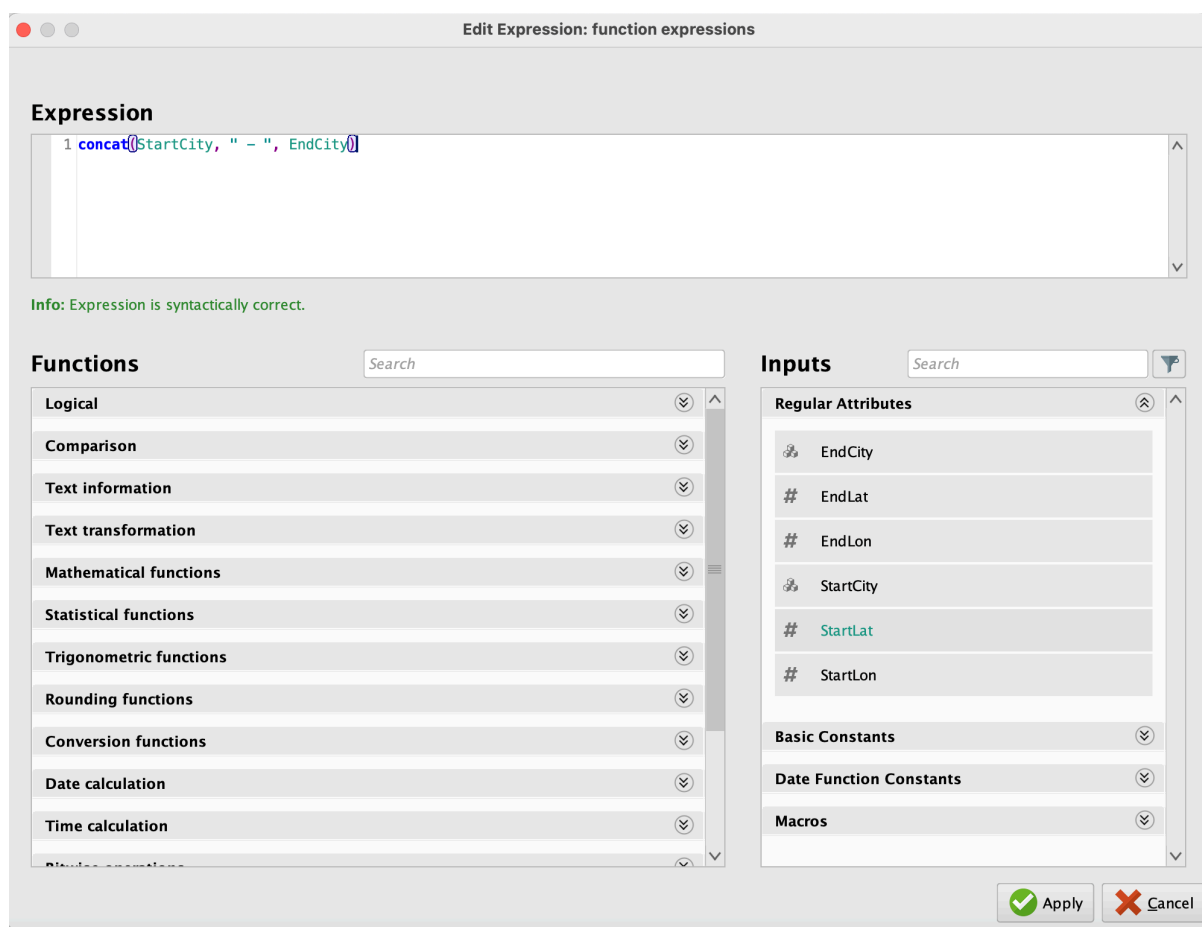


рис.12: Создание сегментов для визуализации с помощью объединения *StartCity* –

EndCity вариаций (Начальная точка – Конечная точка)

Row No.	StartCity	StartLat	StartLon	EndCity	EndLat	EndLon	lat1_rad	lon1_rad	lat2_rad	lon2_rad	distance_km	segment_n...
1	Moscow	55.756	37.618	Nizhny Novg...	56.297	43.936	0.973	0.657	0.983	0.767	397.045	Moscow – Ni...
2	Nizhny Novg...	56.297	43.936	Yekaterinburg	56.839	60.606	0.983	0.767	0.992	1.058	1020.469	Nizhny Novg...
3	Yekaterinburg	56.839	60.606	Novosibirsk	55.008	82.936	0.992	1.058	0.960	1.448	1399.578	Yekaterinbu...
4	Novosibirsk	55.008	82.936	Irkutsk	52.287	104.305	0.960	1.448	0.913	1.820	1434.565	Novosibirsk ...
5	Irkutsk	52.287	104.305	Khabarovsk	48.483	135.084	0.913	1.820	0.846	2.358	2205.331	Irkutsk – Kh...
6	Khabarovsk	48.483	135.084	Vladivostok	43.115	131.886	0.846	2.358	0.753	2.302	646.115	Khabarovsk ...

рис.13: Модифицированные данные после создания столбца segment_name с помощью функции Generate Attributes, который объединил граничные точки маршрутов

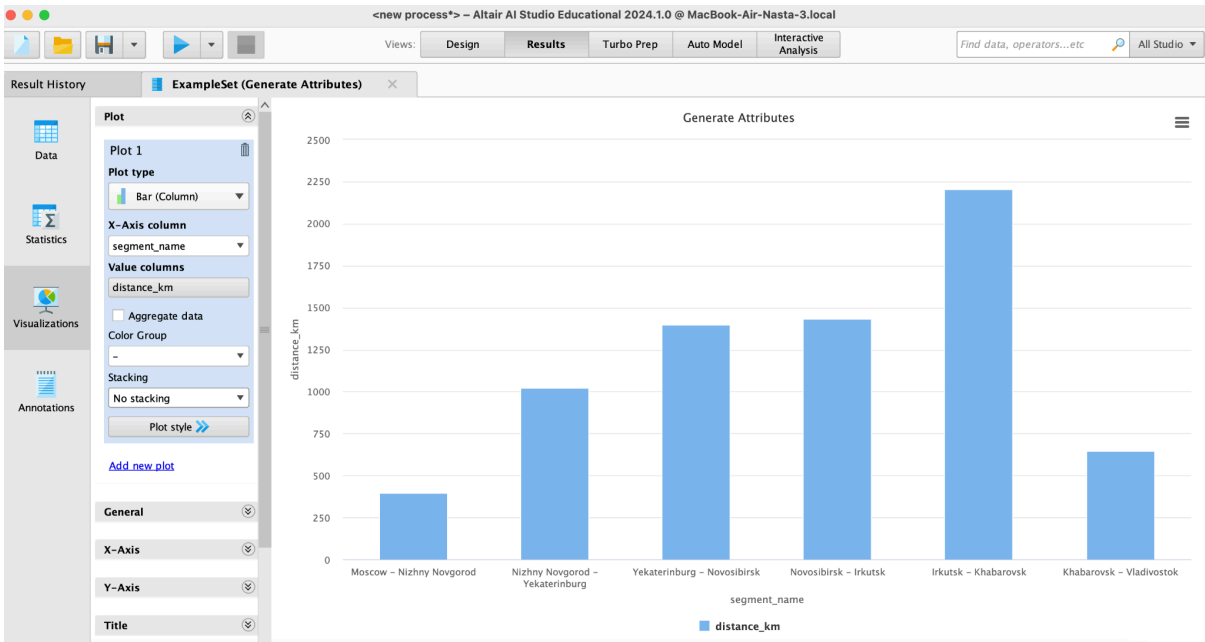


рис.14: Построение столбчатой диаграммы, позволяющей посмотреть протяженности различных маршрутов, основываясь на созданном ранее столбце segment_name, который объединил граничные точки маршрутов

Анализ диаграммы

Глядя на полученный график расстояний, легко увидеть, какой сегмент самый длинный (самый высокий столбец) и как распределяются остальные расстояния. Диаграмма подтверждает расчетные значения: один столбец заметно выше остальных (самый длинный отрезок), один значительно ниже (самый короткий отрезок), остальные – среднего размера. В отчете по работе можно включить эту визуализацию и кратко описать, что она

показывает. Если необходимо, дополнительно отметьте, почему некоторые отрезки такие протяженные (например, в восточной части России города находятся далеко друг от друга).

Фоновая картограмма (Choropleth map)

В окне результатов выберите вкладку Charts (или Visualizations). В списке Plot type найдите Choropleth Map.

- Select map: Russia.
- Value column: выберите числовой столбец, по которому будете окрашивать регионы (например, distance_km).
- Join by / Join key: RapidMiner будет сопоставлять ваши строки с polygon-данными карты по названию субъекта РФ.

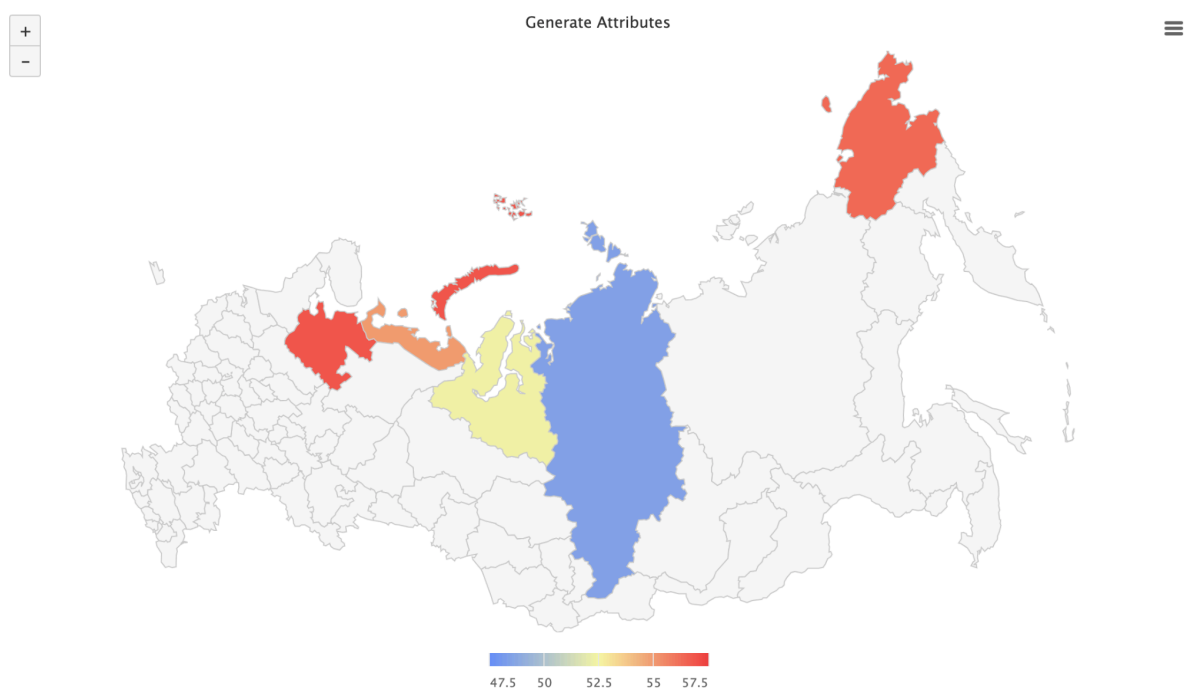


рис.15: Построение картограммы, которая покажет регионы маршрута

Пример выполненной работы

- Шаг 1: Задан CSV-файл с координатами основных точек маршрута. С помощью оператора Read CSV данные успешно загружены в

RapidMiner: получена таблица из 6 строк (сегментов) и 6 столбцов (названия городов и координаты).

- Шаг 2: Подготовка данных не потребовала сложных преобразований, так как все координаты были числовыми. Столбцы широты и долготы были проверены и автоматически распознаны как числовые типы. Данные готовы для расчета расстояний.
- Шаг 3: Реализован расчет расстояний между точками. В процесс добавлен оператор Generate Attributes: сначала выполнено преобразование градусов в радианы, затем по формуле вычислено поле `distance_km`.
- Шаг 4: Выполнен анализ маршрута на основе вычисленных расстояний. Оператор Aggregate подсчитал суммарную дистанцию для всего маршрута. Сопоставление со списком сегментов подтвердило, что именно отрезок между определенными городами является самым протяженным, а минимальное расстояние выявлено между первыми пунктами маршрута.
- Шаг 5: Построена столбчатая диаграмма расстояний сегментов маршрута и подготовлены выводы. График наглядно показал разницу между сегментами: дальневосточный участок резко выделяется по длине.
- Работа завершается выводом о общей сложности путешествия и о том, какие участки являются определяющими в его протяженности.

Приобретенные навыки

В результате выполнения данной лабораторной работы студенты:

- Научились работать с географическими данными в RapidMiner: загрузка координат, понимание формата широты/долготы, базовая проверка корректности геоданных.
- Освоили применение операторов для вычисления по формуле: использовали Generate Attributes для расчета нового показателя (расстояния) на основе существующих столбцов.
- Закрепили знания о тригонометрических функциях и их использовании: применили синус, косинус, арккосинус в контексте реальной задачи, убедились в необходимости перевода градусов в радианы.

- Получили опыт решения практической задачи расчета расстояний без программирования, полностью в визуальной среде Altair AI Studio, что развивает навык быстрого прототипирования аналитических задач.
- Научились интерпретировать результаты вычислений: суммарные показатели, экстремальные значения, а также представлять результаты в наглядной форме (диаграмма), делая выводы, полезные для прикладного использования (например, в логистике или планировании путешествий).

Обобщенная задача для выполнения индивидуального варианта

Задача – научиться проводить анализ географических маршрутов с использованием реальных данных, содержащих географические координаты (широта и долгота). В рамках задания необходимо:

Загрузка и предобработка данных:

- Импортировать набор данных с координатами объектов (городов, остановок маршрута или точек интереса).
- Провести проверку и провести преобразование формата координат для корректного проведения расчетов.

Расчет географических расстояний:

- Реализовать вычисление расстояния между точками, используя математические формулы, реализованные средствами платформы (например, с помощью оператора Generate Attributes).
- Создать новые атрибуты, переводящие градусы в радианы и рассчитывающие расстояния в километрах между парами точек.

Анализ и интерпретация результатов:

- Провести агрегирование данных для получения суммарной длины маршрута, а также определить максимальные, минимальные и средние расстояния между точками.
- Провести анализ отдельных сегментов маршрута, выявив наиболее протяженные и, наоборот, короткие отрезки.

Визуализация:

- Построить столбчатые диаграммы для наглядного сравнения расстояний между сегментами маршрута.

- Создать картограммы (Choropleth maps) для визуализации распределения рассчитанных расстояний на карте, чтобы увидеть географические особенности маршрута.

Распределение вариантов



