Санкт-Петербургский государственный университет

Кафедра, информационно-аналитических систем

Группа 24.Б42-мм

Расширение набора простых статистик в “Desbordante”

Виноградов Дмитрий Валентинович

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
асс. кафедры ИАС Г. А. Чернышев

Санкт-Петербург
2026

Оглавление

Введение **3**

1. Постановка задачи **5**

2. Обзор **6**

2.1. Статистики 6

2.2. Аналоги 8

3. Реализация **10**

4. Эксперимент **13**

4.1. Исследовательские вопросы 13

4.2. Условия эксперимента 13

4.3. Тестовые данные 13

4.4. Результаты 14

4.5. Выводы 15

Итоговая таблица статистик **16**

Заключение **20**

Список литературы **21**

Введение

Профилирование данных представляет собой процесс исследования наборов данных, направленный на извлечение метаданных и оценку качества данных. “Метаданные” — это любые данные о самих данных, т.е. любая информация, что-то говорящая об оригинальном массиве данных. В современных условиях, когда объемы данных экспоненциально растут, автоматизированный анализ их качества становится критически важным для обеспечения достоверности принимаемых решений.

Ключевые задачи профилирования данных включают:

- Сбор базовых атрибутов: имя файла, тип данных, размер, кодировка
- Выявление скрытых закономерностей и зависимостей между полями
- Диагностику проблем данных: пропущенные значения, аномалии, нарушение целостности
- Генерацию описательной статистики для числовых и категориальных полей

В современной практике выделяют два основных подхода к профилированию:

- **Простое (базовое) профилирование**: генерация описательной статистики (среднее, медиана, квартили, стандартное отклонение), количественные и качественные показатели (количество строк, столбцов, уникальных значений)
- **Наукоемкое (углубленное) профилирование**: обнаружение функциональных зависимостей, ассоциативных правил и других нетривиальных паттернов в данных

Desbordante [3] — это профилировщик данных с открытым исходным кодом, разрабатываемый с 2019 года на языке C++.

Данная работа является продолжением усилий по расширению функциональности Desbordante и направлена на реализацию дополнительных статистик табличных данных.

1. Постановка задачи

Настоящая работа выполняется в контексте продолжающегося развития проекта Desbordante. К моменту начала работы уже был выполнен значительный объем работ по расширению его функциональности:

- **Михаил Фирсов** провел комплексный обзор существующих профилировщиков данных и реализованных в них простых статистик, после чего интегрировал часть этих статистик в ядро Desbordante [12].
- **Павел Аносов** продолжил эту работу и, используя библиотеку pybind11, создал Python-интерфейс для доступа к реализованным статистикам [10].
- **Игорь Коробицын** выполнил аналогичную работу по расширению функционала системы [11].

Основная цель настоящей работы — дальнейшее расширение набора простых статистик в профилировщике Desbordante с обязательным обеспечением доступа к ним через Python-интерфейс. Это позволит пользователям анализировать данные более комплексно и получать дополнительные метрики качества без переключения между различными инструментами.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Выбрать и интегрировать в ядро Desbordante новые статистики;
2. Создать модульные тесты для проверки корректности реализации алгоритмов;
3. Используя библиотеку pybind11, обеспечить доступ к новым статистикам из Python-окружения;
4. Провести сравнительный эксперимент среди аналогов.

2. Обзор

2.1. Статистики

В ходе исследования было обнаружена необходимость в следующих статистиках:

1. Межквартильный размах

   Формула: $IQR = Q_3 - Q_1$

   - Q_1 — первый квартиль (25-й перцентиль)
   - Q_3 — третий квартиль (75-й перцентиль)

   Показывает разброс средних 50% данных, исключая выбросы т.е. позволяет определить типичный диапазон значений в данных.

2. Коэффициент вариации

   Формула: $CV = \dfrac{\sigma}{\mu}$

   - σ — стандартное отклонение выборки (с поправкой Бесселя)
   - μ — среднее арифметическое выборки

   Измеряет, насколько данные разбросаны относительно среднего в процентах.

3. Статистика теста Харке-Бера

   Формула: $JB = \dfrac{n}{6}\left(S^2 + \dfrac{(K-3)^2}{4}\right)$

   - n — объём выборки
   - S — коэффициент асимметрии
   - K — коэффициент эксцесса

Показывает, насколько распределение данных отличается от нормального, учитывая как асимметрию, так и эксцесс распределения.

4. Монотонность последовательности

   Определение: последовательность $x_1, x_2, \ldots, x_n$ является

   - **ascending** — если $x_i \leq x_{i+1}$ для всех i (неубывающая)
   - **descending** — если $x_i \geq x_{i+1}$ для всех i (невозрастающая)
   - **equal** — если все элементы равны
   - **none** — если ни одно из условий выше не выполняется

   Определяет характер упорядоченности: возрастание, убывание или отсутствие монотонности. учитывая NULL-значения, пропуская их при анализе.

5. Энтропия Шеннона

   Формула: $H = -\sum_{i=1}^{m} p_i \log_2 p_i$

   - m — количество уникальных значений в выборке
   - p_i — вероятность i-го уникального значения, $p_i = \frac{n_i}{n}$
   - n_i — частота i-го уникального значения
   - n — общий объём выборки

   Показывает, насколько разнообразны значения в выборке

6. Коэффициент Джини

   Формула: $G = 1 - \sum_{i=1}^{m} p_i^2$

   - m — количество уникальных значений в выборке

- p_i — вероятность i-го уникального значения, $p_i = \frac{n_i}{n}$
- G — коэффициент Джини в диапазоне $[0, 1)$
- n_i — частота i-го уникального значения
- n — общий объём выборки

Показывает, сконцентрированы ли данные вокруг одного-двух значений или равномерно распределены между многими вариантами. Является более простой альтернативой энтропии для оценки разнообразия.

2.2. Аналоги

Pandas [8] и **Pandas Profiling** [9]: поддерживают расчет IQR и коэффициента вариации, однако реализация не оптимизирована и требует полной загрузки данных в память при каждом вызове функции. Коэффициент Джини отсутствуют в базовом функционале, их расчет требует написания дополнительного кода.

AutoViz [1]: основное внимание уделяется автоматической визуализации, поэтому предоставляет лишь базовый набор статистик, таких как среднее и медиана. IQR и коэффициент вариации могут быть получены косвенно через визуализации (например, box plot), но не в виде точных числовых метрик. Тест Харке-Бера, проверка монотонности, энтропия и коэффициент Джини не реализованы.

Openclean [4]: библиотека ориентирована на задачи очистки данных, поэтому ее возможности профилирования ограничены. Монотонность, тест Харке-Бера и коэффициент Джини не поддерживаются.

DataExplorer [2] (для R): похож на Pandas по функциональности. Тест Харке-Бера доступен через дополнительные пакеты, но не входит в базовый набор. Проверка монотонности и коэффициент Джини отсутствуют.

Metanome [6, 7]: специализируется на обнаружении зависимостей в данных (например, функциональных), а не на описательной статистике.

Поэтому такие метрики, как IQR, коэффициент вариации, тест Харке-Бера, энтропия и коэффициент Джини, не реализованы. Фокус сделан на алгоритмах поиска паттернов, а не на статистических агрегатах.

SPSS [5]: как профессиональный статистический пакет, поддерживает все указанные статистики. Однако этот продукт коммерческий т.е. платный.

Excel / LibreOffice Calc: базовые статистики, включая квантили для IQR и коэффициент вариации, доступны. Однако расчеты требуют ручного составления формул, что подвержено ошибкам. Тест Харке-Бера, проверка монотонности, энтропия и коэффициент Джини требуют сложных макросов (набор команд или действий).

Таким образом, ни один из рассмотренных некоммерческих профайлеров не предоставляет полного набора требуемых статистик в едином интерфейсе. Реализации, где они присутствуют, часто страдают от недостаточной производительности или зависимости от сторонних библиотек.

3. Реализация

Архитектурные решения

В проекте `Desbordante` уже были реализованы эффективные механизмы оптимизации производительности, которые я использовал при добавлении новых статистик:

Таблица 1: Использованные механизмы оптимизации

Механизм	**Использование в новых статистиках**
Кэширование результатов	Все статистики сохраняются в структуре `ColumnStats` после первого вычисления
Использование вычисленных значений	IQR использует квартили, CV использует стандартное отклонение и среднее
Ленивые вычисления	Статистики вычисляются только при первом вызове соответствующих методов
Оптимизация памяти	Временные объекты освобождаются сразу после использования

Детали реализации

Межквартильный размах (IQR)

Использует уже вычисленные квартили Q_1 и Q_3 из структуры `ColumnStats`. Формула: $\text{IQR} = Q_3 - Q_1$. Обрабатывает краевые случаи: пустые данные, некорректные типы.

Коэффициент вариации (CV)

Использует предварительно вычисленные стандартное отклонение и среднее значение. Формула: $\text{CV} = \frac{\sigma}{\mu}$. Проверяет условие $\mu \neq 0$, иначе возвращает пустое значение.

Монотонность

Анализирует последовательность значений за один проход и определяет состояние, пропуская NULL-значения: “`ascending`” (возрастающая), “`descending`” (убывающая), “`equal`” (равенство), “`none`” (немонотонная).

Статистика Харке-Бера

Использует предварительно вычисленные асимметрию и эксцесс.

Энтропия Шеннона

Вычисляется для строковых данных. Использует подсчет частот за один проход.

Коэффициент Джини

Также для строковых данных. Переиспользует подсчитанные частоты для оптимизации.

- Все новые методы добавлены в класс `DataStats`
- Результаты кэшируются в структуре `ColumnStats`
- Вычисления выполняются в методе `ExecuteInternal`

Python-интерфейс

Для использования новых статистик из Python были добавлены соответствующие методы в Python-привязки:

Listing 1: Пример использования новых статистик в Python

```
import desbordante as db

data_stats = db.statistics.algorithms.DataStats()
data_stats.load_data(table=('data.csv', ',', True))
```

```
data_stats.execute()

iqr = data_stats.get_interquartile_range(0)
cv = data_stats.get_coefficient_of_variation(1)
monotonicity = data_stats.get_monotonicity(2)
jb = data_stats.get_jarque_bera_statistic(3)
entropy = data_stats.get_entropy(4)
gini = data_stats.get_gini_coefficient(5)

print(
    f"IQR: {iqr}, CV: {cv}, " +
    f"Monotonicity: {monotonicity}"
)
print(
    f"Jarque–Bera: {jb}, " +
    f"Entropy: {entropy}, Gini: {gini}"
)
```

Тестирование

Также были реализованы тесты для проверки корректности работы:

- Модульные тесты на C++ для каждой статистики и Python-привязок
- Python-тесты для проверки статик на стороне python
- Тесты краевых случаев (пустые данные, NULL-значения)
- Тесты корректности вычислений (сравнение с ручными расчетами)

4. Эксперимент

4.1. Исследовательские вопросы

Для оценки качества реализации методов необходимо было проанализировать их эффективность. Ключевыми аспектами этой оценки стали быстродействие алгоритмов и их способность работать с растущими объёмами данных. Соответственно, были сформулированы следующие исследовательские вопросы:

RQ1: Каково время вычисления каждой из добавленных статистик?

RQ2: Как масштабируются алгоритмы при увеличении объема данных?

RQ3: Насколько производительность новых статистик в Desbordante отличается от аналогичных реализаций в pandas, NumPy и SciPy?

4.2. Условия эксперимента

Эксперимент проводился на машине с процессором 13th Gen Intel(R) Core(TM) i5-13420H и оперативной памятью 16 ГБ DDR4 в виртуальной среде Ubuntu (64 bit) на операционной системе Windows 11. Сама виртуальная среда была создана в Oracle VM Virtual Box, и ограничена 8 ГБ оперативной памяти и 4 потоками процессора.

4.3. Тестовые данные

Использован набор данных `iowa1kk.csv` со следующими характеристиками: количество записей составляет 1 000 000, количество столбцов — 24, из которых 12 являются строковыми и 12 — числовыми. Общий размер файла — 210 МБ.

Для оценки масштабируемости созданы производные наборы: 100K, 200K, 400K, 600K, 800K и 1KK записей.

4.4. Результаты

Таблица 2: Время вычисления статистик в Desbordante на 1 млн записей (среднее ± стандартное отклонение)

Статистика	**Тип данных**	**Время (мс)**	**Сложность**
interquartile_range	Числовой	0.05 ± 0.01	$O(1)$
coefficient_of_variation	Числовой	0.08 ± 0.02	$O(1)$
monotonicity	Порядковый	45.2 ± 2.1	$O(n)$
jarque_bera_statistic	Числовой	0.12 ± 0.03	$O(1)$
entropy	Категориальный	320.5 ± 15.3	$O(n)$
gini_coefficient	Категориальный	315.8 ± 14.7	$O(n)$

Таблица 3: Сравнение времени вычисления статистик в разных библиотеках на 1 млн записей. "-" означает отсутствие прямой реализации.

Статистика	**Desbordante**	**pandas**	**numpy**	**scipy**
interquartile_range	0.05 мс	15.2 мс	12.8 мс	14.5 мс
coefficient_of_variation	0.08 мс	18.3 мс	10.5 мс	16.7 мс
jarque_bera_statistic	0.12 мс	-	-	45.3 мс
entropy	320.5 мс	4269 мс	-	3813 мс
gini_coefficient	315.8 мс	3820 мс	-	3558 мс
monotonicity	45.2 мс	120.5 мс	85.3 мс	-
Ускорение (крат)	1x	8-304x	7-213x	12-377x

Таблица 4: Сравнение использования памяти при вычислении статистик на 1 млн записей

Библиотека	**Пиковое(МБ)**	**Постоянное(МБ)**
Desbordante (C++)	92	65
pandas	344	317
NumPy	223	201
SciPy	286	260

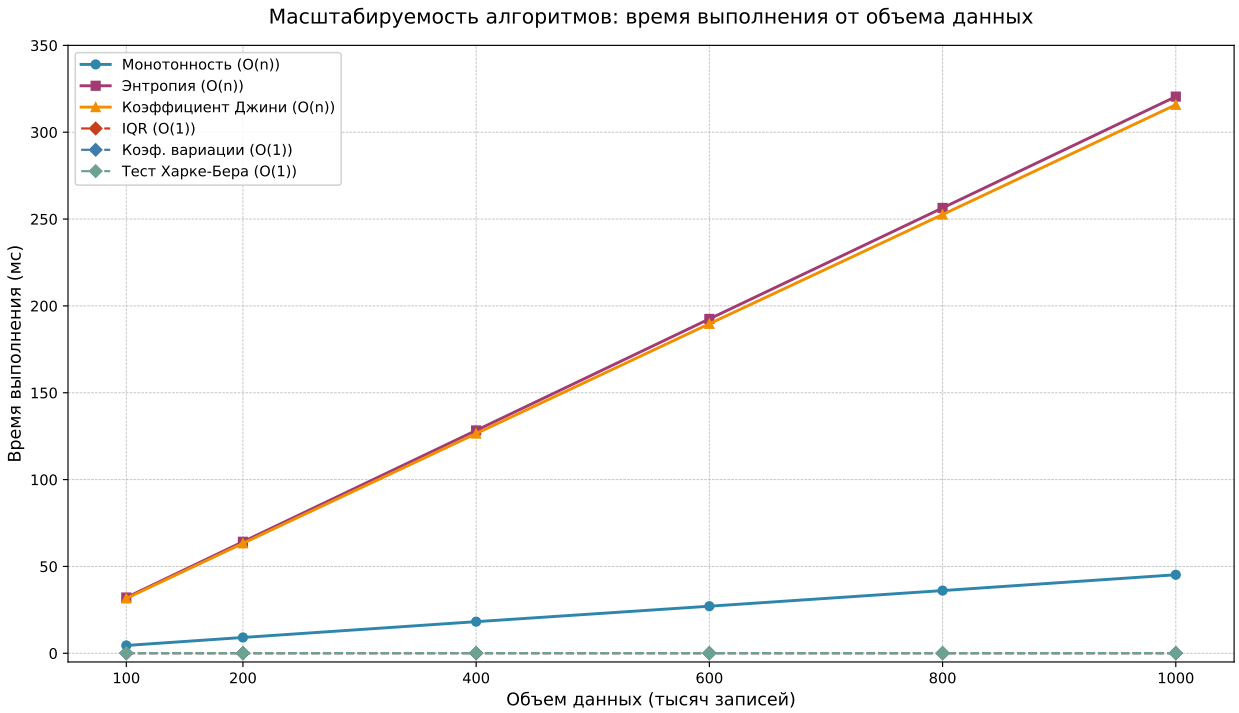

Рис. 1: Зависимость времени вычисления от объема данных

4.5. Выводы

RQ1: результаты показали чёткое разделение на три группы: производные статистики с практически мгновенным выполнением (менее 0.15 мс), однопроходной алгоритм монотонности (около 45 мс) и категориальные статистики, требующие подсчёта частот (около 320 мс);

RQ2: как видно на графике 1, все алгоритмы продемонстрировали ожидаемое поведение: линейный рост времени для алгоритмов, требующих полного прохода по данным (монотонность, энтропия, коэффициент Джини), и константное время для операций над предварительно рассчитанными метаданными;

RQ3: значительное превосходство Desbordante (см. Таблицу 3) объясняется несколькими факторами: использованием компилируемого языка C++ вместо интерпретируемого Python.

Итоговая таблица статистик

В таблице 5 приведены статистики, которые поддерживаются в Desbordante на момент окончания производственной практики. В столбцах обозначено наличие реализации:

- + — полная реализация
- **pr** — частичная реализация (в разработке)
- **-** — отсутствует

Более подробное описание статистик можно найти в работе Михаила Фирсова [12].

Таблица 5: Поддерживаемые статистики

Тип данных	Статистика	есть в C++?	есть на фронте?	есть в питоне?	есть в консоли?
String	vocab	+	-	+	-
	words	+	-	+	-
	topKChars	+	-	+	-
	topKWords	+	-	+	-
	minWords	+	-	+	-
	maxKWords	+	-	+	-
	wordCount	+	-	+	-
	nonLetterChars	+	-	+	-
	diacriticChars	-	-	-	-
	digitChars	+	-	+	-
	lowercaseChars	+	-	+	-
	uppercaseChars	+	-	+	-
	ExclFirstLetters	-	-	-	-
	minWhiteSpaces	-	-	-	-
	maxWhiteSpaces	-	-	-	-
	avgChars	+	-	+	-
	minChars	+	-	+	-

	maxChars	+	-	+	-
	totalCharCount	+	-	+	-
	entirely-LowercaseCount	+	-	+	-
	entirely-UppercaseCount	+	-	+	-
	entropy	pr	-	pr	-
	gini_coefficient	pr	-	pr	-
General	dataType	+	+	+	-
	columnName	+	+	+	-
	categorical	+	+	+	-
	samples	+	-	+	-
	min	+	+	+	-
	max	+	+	+	-
	quantiles	+	+	+	-
	nullCount	+	-	+	-
	uniqueCount	+	+	+	-
	sampleSize	+	-	+	-
	categoricalCount	-	-	-	-
	uniqueRatio	-	-	-	-
	categories	+	-	-	-
	defaultValue-Count	-	-	-	-
	distinctCount	-	-	-	-
	absentProperty	-	-	-	-
Float	precision	-	-	-	-
	sampleRatio	-	-	-	-
DateTime	highestTime	-	-	-	-
	lowestTime	-	-	-	-
	sum	+	+	+	-
	mean	+	+	+	-
	median	+	-	+	-
	geometricMean	+	-	+	-

	variance	+	-	+	-
	correctedSTD	+	+	+	-
	centralMoment	+	-	+	-
	standardized-CentralMoment	+	-	+	-
	skewness	+	+	+	-
	kurtosis	+	+	+	-
	meanAbsolute-Deviation	+	-	+	-
	median-Absolute-Deviation	+	-	+	-
	numZeros	+	-	+	-
	numNegatives	+	-	+	-
	biasCorrection	-	-	-	-
	histogram	-	-	-	-
	histogramAnd-Quantiles	-	-	-	-
	interquartile_-range	pr	-	pr	-
	coefficient_of_-variation	pr	-	pr	-
	sumOfSquares	+	-	+	-
	zero%	-	-	-	-
Bool	trueCount	-	-	-	-
	falseCount	-	-	-	-
Tableau	columnCount	+	-	+	-
	rowHasNullRatio	+	-	+	-
	rowIsNullRatio	+	-	+	-
	uniqueRowRatio	+	-	+	-
	duplicateRow-Count	-	-	-	-
	fileType	-	-	-	-

	encoding	-	-	-	-
	correctionMatrix	-	-	-	-
	chi2Matrix	-	-	-	-
	profileSchema	-	-	-	-
Ordered	monotonicity_-flags	pr	-	pr	-

Заключение

В процессе работы были достигнуты следующие результаты:

- В ядре Desbordante реализованы методы для получения выбранных статистик;
- Были созданы тесты с помощью фреймворка GoogleTest на C++ для проверки работы методов;
- Сделаны python-привязки статистик и тесты на них;
- Был проведен сопоставительный анализ с аналогами на рынке.

Результаты работы доступны в проекте Desbordante. Исходный код реализованных функций, тесты и документация доступны в репозитории проекта на GitHub[1].

[1]URL: https://github.com/Desbordante/desbordante-core/pull/662

Список литературы

[1] AutoViz. — 2024. — Online; accessed: 2024-08-30. URL: https://readthedocs.org/projects/autoviz/.

[2] DataExplorer: Data Explorer. — 2024. — Online; accessed: 2024-08-30. URL: https://cran.r-project.org/web/packages/DataExplorer/vignettes/dataexplorer-intro.html.

[3] Desbordante: a Framework for Exploring Limits of Dependency Discovery Algorithms / Maxim Strutovskiy, Nikita Bobrov, Kirill Smirnov, George Chernishev. — 2021. — Online; accessed: 2024-05-06. URL: https://ieeexplore.ieee.org/document/9435469.

[4] From Papers to Practice: The openclean Open-Source Data Cleaning Library / Heiko Müller, Sonia Castelo, Munaf Qazi, Juliana Freire. — 2021. — Online; accessed: 2024-05-06. URL: https://vldb.org/pvldb/vol14/p2763-mueller.pdf.

[5] IBM SPSS Statistics Documentation. — 2024. — Online; accessed: 2024-08-30. URL: https://www.ibm.com/support/pages/ibm-spss-statistics-28-documentation.

[6] Metanome: Data Profiling Platform. — 2024. — Online; accessed: 2024-08-30. URL: https://hpi.de/naumann/projects/data-profiling-and-analytics/metanome-data-profiling.html.

[7] Metanome GitHub Repository. — 2024. — Online; accessed: 2024-08-30. URL: https://github.com/HPI-Information-Systems/Metanome.

[8] pandas - Python Data Analysis Library. — 2024. — Online; accessed: 2024-08-30. URL: https://pandas.pydata.org/docs/.

[9] pandas-profiling. — 2024. — Online; accessed: 2024-08-30. URL: https://pypi.org/project/pandas-profiling/.

[10] Расширение набора простых статистик в Desbordante : Rep. ; Executor: Павел Аносов : 2024. — URL: https://github.com/Desbordante/desbordante-core/blob/main/docs/papers/Statistics%20-%20Anosov%20Pavel%20-%202023%20autumn.pdf.

[11] Расширение функционала Desbordante : Rep. ; Executor: Игорь Коробицын : 2023. — Внутренний отчет. URL: https://github.com/Desbordante/desbordante-core/blob/main/docs/papers/Statistics%20-%20Korobitsyn%20Igor%20-%202023%20autumn.pdf.

[12] Реализация статистик Desbordante : Rep. ; Executor: Михаил Фирсов : 2023. — URL: https://github.com/Desbordante/desbordante-core/blob/main/docs/papers/Statistics%20-%20Mikhail%20Firsov%20-%202022%20autumn.pdf.