

# Самостоятельная работа 1, Тема 4

---

Выполнил: Гневнов А.Е., ИВТ 2.1

## 1. Графики

### 1.1 Свечной график (OHLC)

- **Описание:** Отображает динамику цен (открытие, максимум, минимум, закрытие) в виде «свечей».
- **Использование программистами:**
  - Анализ биржевых данных, алгоритмический трейдинг, финансовые приложения.
- **Особенности:**
  - Зеленые/красные свечи показывают рост/падение цены.
  - Подходит для анализа волатильности.
- **Рекомендации:**
  - Используйте библиотеки: **Plotly** (Python/JavaScript) для интерактивных графиков, **TradingView API** для интеграции с торговыми платформами .
  - Для веб-приложений добавьте инструменты зумирования и подсказки.
- **Доп. материалы:**
  - [Свечной график](#).
  - Примеры реализации на GitHub: [Candlestick Chart with D3.js](#).
- **Изображение визуализации**

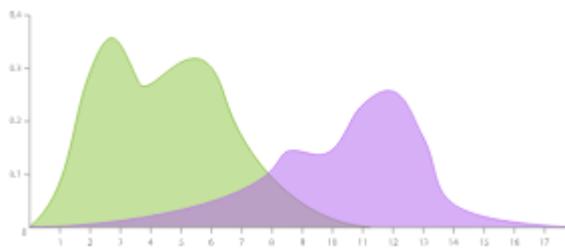


---

### 1.2 График плотности

- **Описание:** Показывает распределение данных через сглаженную кривую.
- **Использование программистами:**
  - Анализ плотности распределения (например, время отклика API, возраст пользователей).
- **Особенности:**
  - Чувствителен к выбору параметра сглаживания (bandwidth).
- **Рекомендации:**
  - **Seaborn** (Python) для автоматического построения, **ggplot2** (R) для кастомных настроек.
  - Используйте цветовые градиенты для выделения пиков .
- **Доп. материалы:**
  - [График плотности](#).
  - Статья: «[Визуализация распределений в Python](#)» на Habr .

- **Изображение визуализации**



### 1.3 График баров (OHLC, Open-High-Low-Close)

- **Описание:** Альтернатива свечному графику с вертикальными линиями для отображения цен.
- **Использование программистами:**
  - Финансовые приложения, анализ исторических данных.
- **Рекомендации:**
  - **Matplotlib** (Python) для статических графиков, **Highcharts** (JavaScript) для веба.
  - Добавьте легенду для пояснения цветов линий .
- **Доп. материалы:**
  - [График баров](#).
  - Пример: [Реализация OHLC на Plotly](#).
- **Изображение визуализации**



### 1.4 Линейный график

- **Описание:** Отображает зависимость данных между двумя переменными.
- **Использование программистами:**
  - Мониторинг временных рядов (загрузка сервера, активность пользователей).
- **Рекомендации:**
  - **D3.js** для кастомных решений, **Plotly** для интерактивности.
  - Избегайте наложения более 5 линий для сохранения читаемости .
- **Доп. материалы:**
  - [Линейный график](#).
- **Изображение визуализации**



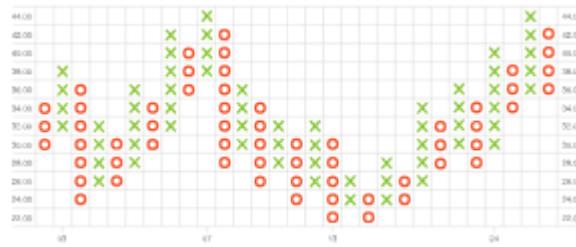
## 1.5 График Каги

- Описание:** Фильтрует «шум» данных, акцентируя значимые изменения.
- Использование программистами:**
  - Анализ долгосрочных трендов (например, цены акций).
- Рекомендации:**
  - Реализация через кастомные скрипты на Python/R.
  - Используйте библиотеку **pandas** для обработки данных .
- Доп. материалы:**
  - [График Каги](#).
- Изображение визуализации**



## 1.6 График «крестики-нолики»

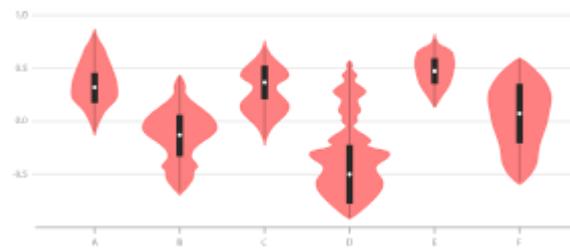
- Описание:** Фиксирует изменение цены без учета времени.
- Использование программистами:**
  - Выявление уровней поддержки/сопротивления в трейдинге.
- Рекомендации:**
  - Pandas** для обработки данных, **Plotly** для визуализации.
  - Добавьте параметр минимального изменения цены для фильтрации шума .
- Доп. материалы:**
  - [График «крестики-нолики»](#).
- Изображение визуализации**



## 1.7 Скрипичный график

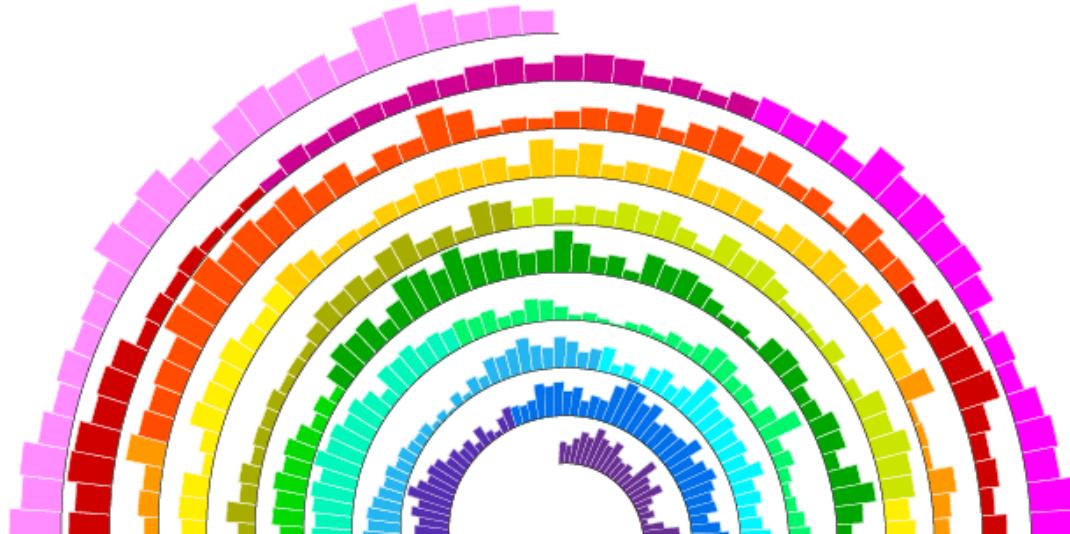
- Описание:** Комбинирует гистограмму и график плотности.
- Использование программистами:**
  - Сравнение распределений между категориями (например, доходы по регионам).
- Рекомендации:**
  - Seaborn** (Python) для автоматического построения, **Plotly** для интерактивности.
  - Используйте цветовые палитры для разделения категорий .
- Доп. материалы:**

- [Скрипичный график.](#)
- Пример: [Violin Plot Tutorial](#).
- **Изображение визуализации**



## 1.8 Спиральный график

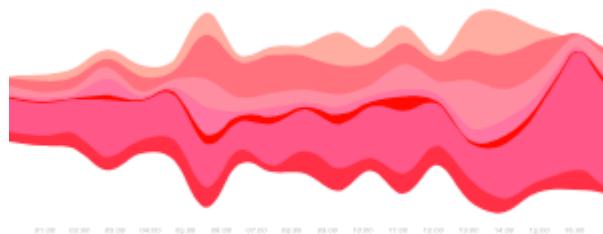
- **Описание:** Отображает данные в виде спирали с временной осью по кругу.
- **Использование программистами:**
  - Анализ циклических данных (например, активность по часам).
- **Рекомендации:**
  - Кастомные решения на [D3.js](#) или [Plotly](#).
  - Добавьте анимацию для демонстрации временных изменений .
- **Доп. материалы:**
  - [Спиральный график.](#)
  - Пример: [Spiral Visualization with D3.js](#).
- **Изображение визуализации**



## 1.9 Потоковый график

- **Описание:** Визуализирует изменение долей данных во времени.
- **Использование программистами:**
  - Анализ структуры данных (например, распределение трафика по источникам).
- **Рекомендации:**
  - [Plotly](#) для интерактивных графиков, [Highcharts](#) для статических.
  - Используйте прозрачность для наложения областей .
- **Доп. материалы:**

- Потоковый график.
- Изображение визуализации

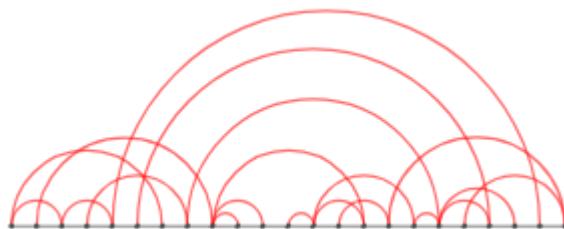


---

## 2. Диаграммы

### 2.1 Дуговая диаграмма

- Описание: Отображает связи между элементами в виде дуг.
- Использование программистами:
  - Визуализация сетевых данных (социальные графы, зависимости в коде).
- Рекомендации:
  - D3.js для кастомной реализации, NetworkX (Python) для анализа графов.
  - Избегайте перегрузки: ограничьте число связей до 50 .
- Доп. материалы:
  - Дуговая диаграмма.
  - Пример: Arc Diagram in D3.js.
- Изображение визуализации



---

### 2.2 Диаграмма с областями

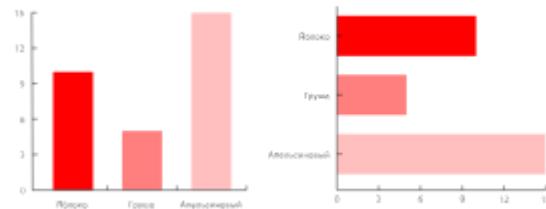
- Описание: Закрашенная область под линией для отображения накопленных данных.
- Использование программистами:
  - Визуализация общего дохода по месяцам, трафика сайта.
- Рекомендации:
  - Highcharts для веба, Matplotlib для статических отчетов.
  - Используйте разные цвета для каждой категории .
- Доп. материалы:
  - Диаграмма с областями.

- **Изображение визуализации**



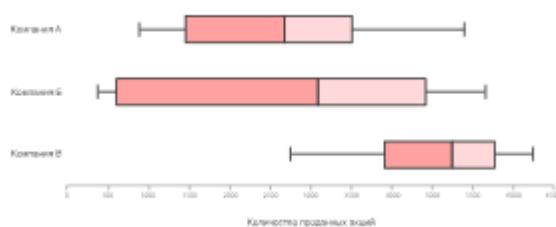
## 2.3 Столбиковая диаграмма

- **Описание:** Сравнивает категории через высоту столбцов.
- **Использование программистами:**
  - Сравнение метрик (например, количество ошибок по модулям).
- **Рекомендации:**
  - **Chart.js** для простых решений, **Plotly** для 3D-визуализации.
  - Добавьте подписи значений над столбцами .
- **Доп. материалы:**
  - [Столбиковая диаграмма](#).
- **Изображение визуализации**



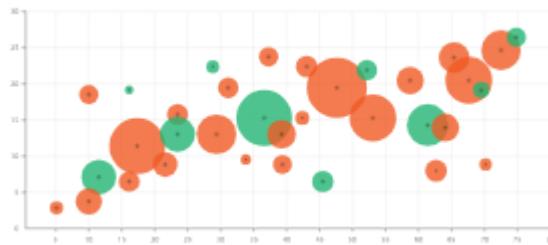
## 2.4 Диаграмма размаха («ящик с усами»)

- **Описание:** Показывает медиану, квартили и выбросы.
- **Использование программистами:**
  - Анализ распределения данных (например, время выполнения запросов).
- **Рекомендации:**
  - **Seaborn** (Python) для автоматического построения, **Plotly** для интерактивности.
  - Используйте горизонтальную ориентацию для сравнения множества категорий .
- **Доп. материалы:**
  - [Диаграмма размаха](#).
- **Изображение визуализации**



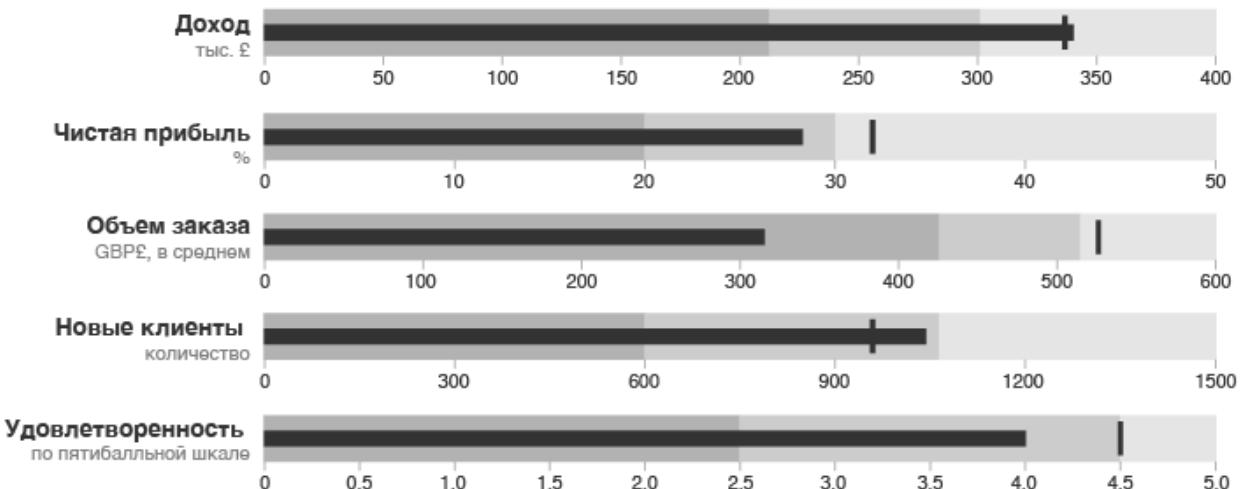
## 2.5 Пузырьковая диаграмма

- Описание:** Отображает три переменные через координаты X, Y и размер пузырька.
- Использование программистами:**
  - Анализ многомерных данных (например, доход, расход, прибыль).
- Рекомендации:**
  - D3.js для кастомных решений, Plotly для анимации.
  - Нормализуйте размер пузырьков для избежания искажений .
- Доп. материалы:**
  - Пузырьковая диаграмма.
  - Пример: [Bubble Chart with Animation](#).
- Изображение визуализации**



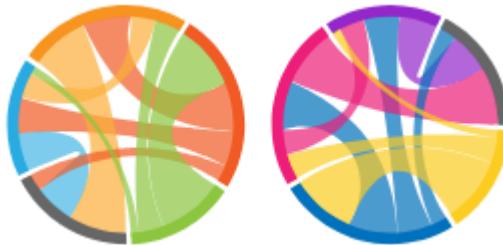
## 2.6 Пулевая диаграмма

- Описание:** Сравнивает фактические значения с целевыми через «пули».
- Использование программистами:**
  - Отслеживание KPI (например, выполнение плана продаж).
- Рекомендации:**
  - Power BI для бизнес-аналитики, Tableau для визуализации.
  - Используйте цветовые индикаторы для превышения/недостижения целей .
- Доп. материалы:**
  - Пример: [Bullet Chart Design Guide](#).
- Изображение визуализации**



## 2.7 Хордовая диаграмма

- **Описание:** Визуализирует взаимосвязи между объектами.
- **Использование программистами:**
  - Анализ потоков данных (миграция пользователей).
- **Рекомендации:**
  - **D3.js** для интерактивных решений, **Python NetworkX** для статических.
  - Избегайте перегрузки: ограничьте число узлов до 20 .
- **Доп. материалы:**
  - [Хордовая диаграмма](#).
  - Пример: [Chord Diagram Tutorial](#).
- **Изображение визуализации**



## 2.8 Кольцевая диаграмма

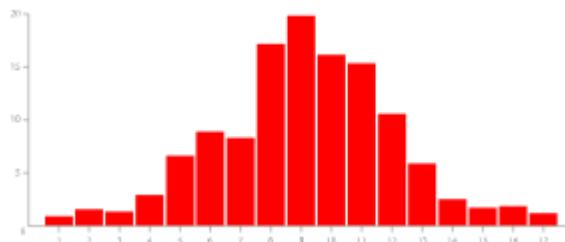
- **Описание:** Аналог круговой диаграммы с вырезанным центром.
- **Использование программистами:**
  - Отображение долей (например, распределение ресурсов).
- **Рекомендации:**
  - **Chart.js** для веба, **Matplotlib** для статических отчетов.
  - Используйте подписи внутри секторов для наглядности .
- **Доп. материалы:**
  - [Кольцевая диаграмма](#).
- **Изображение визуализации**



## 2.9 Гистограмма

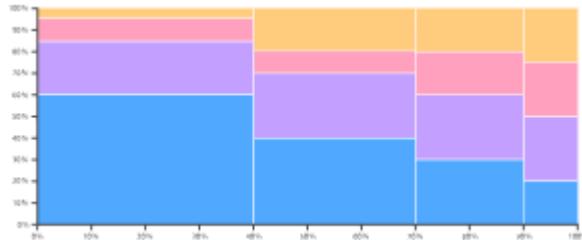
- **Описание:** Показывает распределение данных по интервалам.
- **Использование программистами:**
  - Анализ частотности событий (например, время ответа сервера).
- **Рекомендации:**
  - **Seaborn** для автоматического подбора бинов, **Excel** для быстрых отчетов.
  - Используйте прозрачные столбцы для наложения распределений .
- **Доп. материалы:**
  - [Гистограмма](#).

- **Изображение визуализации**



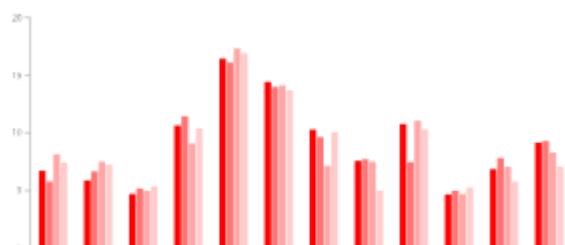
## 2.10 Диаграмма Маримекко

- **Описание:** Комбинирует столбцы разной ширины и высоты.
- **Использование программистами:**
  - Анализ двумерных данных (продажи по регионам и категориям).
- **Рекомендации:**
  - D3.js для кастомной реализации, Tableau для готовых решений.
  - Добавьте легенду для пояснения цветов и размеров .
- **Доп. материалы:**
  - [Диаграмма Маримекко](#).
  - Пример: [Marimekko Chart in D3.js](#).
- **Изображение визуализации**



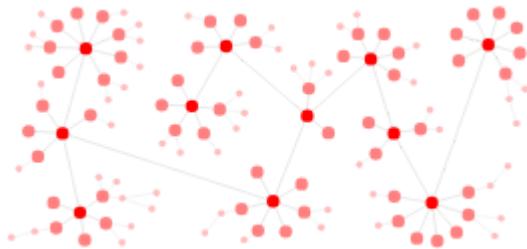
## 2.11 Столбиковая диаграмма с группировкой

- **Описание:** Сравнивает подкатегории внутри категорий.
- **Использование программистами:**
  - Анализ данных с вложенными категориями.
- **Рекомендации:**
  - Plotly для интерактивности, Excel для простых задач.
  - Используйте разные цвета для каждой подкатегории .
- **Доп. материалы:**
  - [Столбиковая диаграмма с группировкой](#).
- **Изображение визуализации**



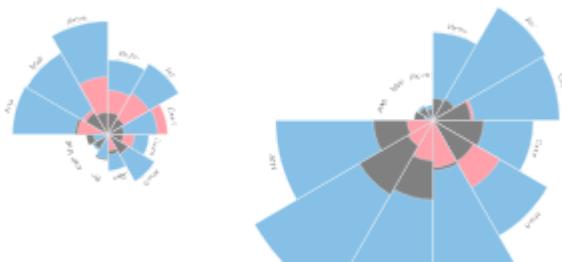
## 2.12 Сетевая диаграмма

- **Описание:** Визуализирует связи в сетевых структурах.
- **Использование программистами:**
  - Анализ графов (социальные сети, зависимости в коде).
- **Рекомендации:**
  - **NetworkX** (Python) для анализа, **Gephi** для визуализации.
  - Используйте алгоритмы размещения узлов (например, ForceAtlas2).
- **Доп. материалы:**
  - [Сетевая диаграмма](#).
  - Пример: [Network Graph Tutorial](#).
- **Изображение визуализации**



## 2.13 Диаграмма «роза Найтингейл»

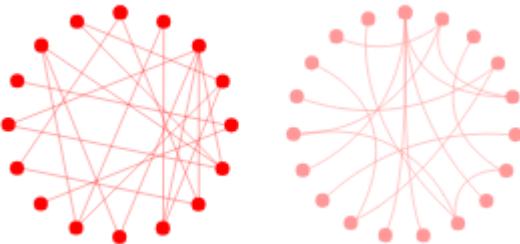
- **Описание:** Отображает данные в виде лепестков разного размера и угла.
- **Использование программистами:**
  - Визуализация циклических данных (например, активность по часам).
- **Рекомендации:**
  - **Python Matplotlib** для построения, **D3.js** для интерактивности.
  - Используйте радиальные оси для точного отображения углов.
- **Доп. материалы:**
  - [Диаграмма «роза Найтингейл»](#).
  - Пример: [Polar Chart in Python](#).
- **Изображение визуализации**



## 2.14 Неленточная хордовая диаграмма

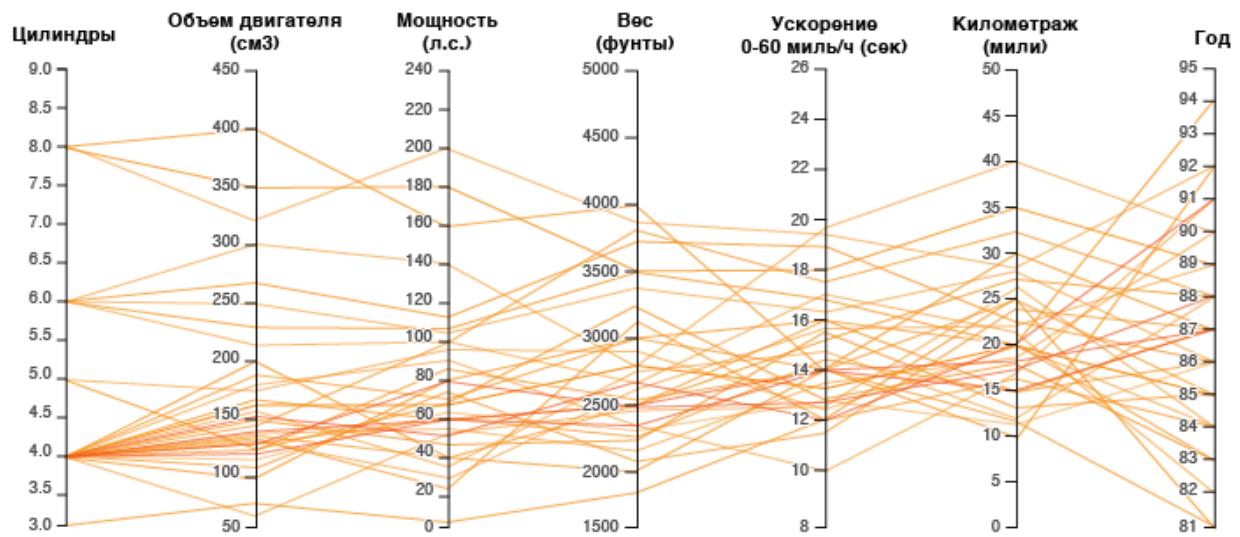
- **Описание:** Упрощенная версия хоровой диаграммы без лент.
- **Использование программистами:**
  - Визуализация связей в компактном формате.
- **Рекомендации:**
  - **D3.js** для реализации, **Python Plotly** для статических версий.

- Ограничьте число узлов до 10 для читаемости .
- **Доп. материалы:**
  - Неленточная хордовая диаграмма.
  - Пример: [Chord Diagram without Ribbons](#).
- **Изображение визуализации**



## 2.15 Диаграмма с параллельными координатами

- **Описание:** Отображает многомерные данные через параллельные оси.
- **Использование программистами:**
  - Анализ взаимосвязей между множеством переменных.
- **Рекомендации:**
  - **Plotly** для интерактивности, **Python Pandas** для обработки данных.
  - Нормализуйте данные для сравнения шкал .
- **Доп. материалы:**
  - [Диаграмма с параллельными координатами](#).
  - Пример: [Parallel Coordinates in D3.js](#).
- **Изображение визуализации**



## 2.16 Пиктографическая диаграмма

- **Описание:** Использует иконки для представления данных.
- **Использование программистами:**
  - Визуализация данных для неспециалистов (например, инфографика).
- **Рекомендации:**
  - **Canva** для дизайна, **Tableau** для интеграции с аналитикой.

- Используйте единый стиль иконок .
- **Доп. материалы:**
  - Пиктографическая диаграмма.
- **Изображение визуализации**



## 2.17 Круговая диаграмма

- **Описание:** Показывает доли данных в виде секторов круга.
- **Использование программистами:**
  - Отображение процентного соотношения (например, распределение трафика).
- **Рекомендации:**
  - **Chart.js** для веба, **Excel** для быстрых отчетов.
  - Ограничьте число секторов до 5–7 для читаемости .
- **Доп. материалы:**
  - Круговая диаграмма.
- **Изображение визуализации**

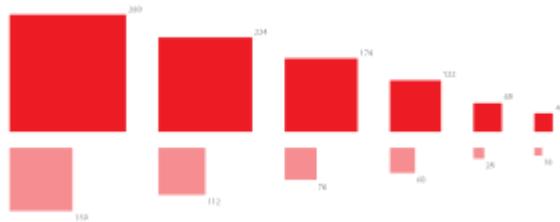



---

## 2.18 Диаграмма с пропорциональными областями

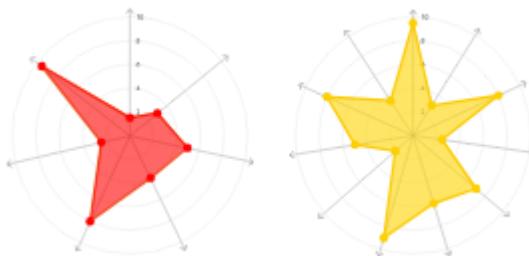
- **Описание:** Размер области соответствует значению данных.
- **Использование программистами:**
  - Сравнение величин через площадь (например, население стран).
- **Рекомендации:**
  - **D3.js** для кастомных решений, **Tableau** для автоматизации.
  - Используйте логарифмическую шкалу для больших диапазонов .
- **Доп. материалы:**
  - [Диаграмма с пропорциональными областями](#).
  - Пример: [Bubble Map in D3.js](#).

- **Изображение визуализации**



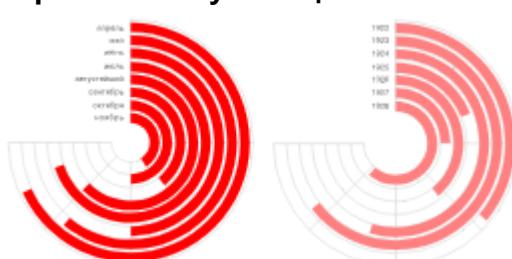
## 2.19 Радиальная диаграмма

- **Описание:** Отображает данные в полярной системе координат.
- **Использование программистами:**
  - Визуализация циклических данных (например, активность по дням недели).
- **Рекомендации:**
  - **Plotly** для интерактивности, **Matplotlib** для статических графиков.
  - Добавьте сетку для улучшения точности восприятия .
- **Доп. материалы:**
  - [Радиальная диаграмма](#).
  - Пример: [Radial Bar Chart in Python](#).
- **Изображение визуализации**



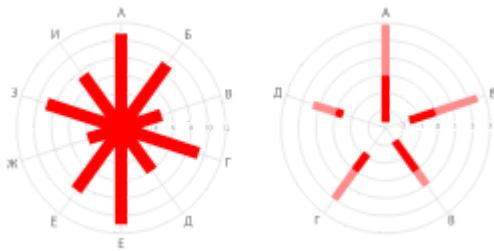
## 2.20 Радиальная полосчатая диаграмма

- **Описание:** Полосы расположены по кругу.
- **Использование программистами:**
  - Сравнение категорий в компактном формате.
- **Рекомендации:**
  - **D3.js** для кастомных решений, **Excel** для простых задач.
  - Используйте цветовые коды для категорий .
- **Доп. материалы:**
  - [Радиальная полосчатая диаграмма](#).
- **Изображение визуализации**



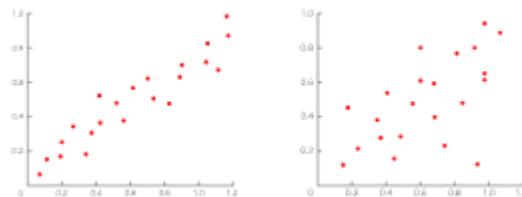
## 2.21 Радиальная столбчатая диаграмма

- **Описание:** Столбцы расположены по кругу.
- **Использование программистами:**
  - Визуализация данных с циклической природой.
- **Рекомендации:**
  - **Highcharts** для веба, **Python Matplotlib** для статических графиков.
  - Избегайте перекрытия столбцов .
- **Доп. материалы:**
  - [Радиальная столбчатая диаграмма](#).
- **Изображение визуализации**



## 2.22 Диаграмма рассеяния

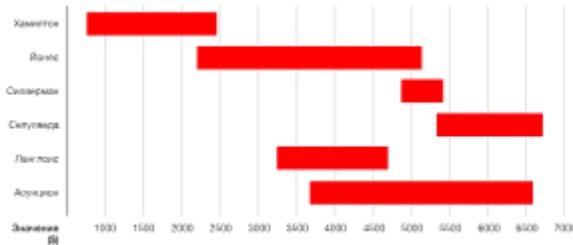
- **Описание:** Отображает взаимосвязь двух переменных через точки.
- **Использование программистами:**
  - Поиск корреляций (например, зависимость дохода от возраста).
- **Рекомендации:**
  - **Seaborn** для автоматической регрессии, **Plotly** для 3D-рассеяния.
  - Добавьте линию тренда для визуализации тренда .
- **Доп. материалы:**
  - [Диаграмма рассеяния](#).
  - Пример: [Scatterplot Matrix in Python](#).
- **Изображение визуализации**



## 2.23 Диаграмма диапазонов

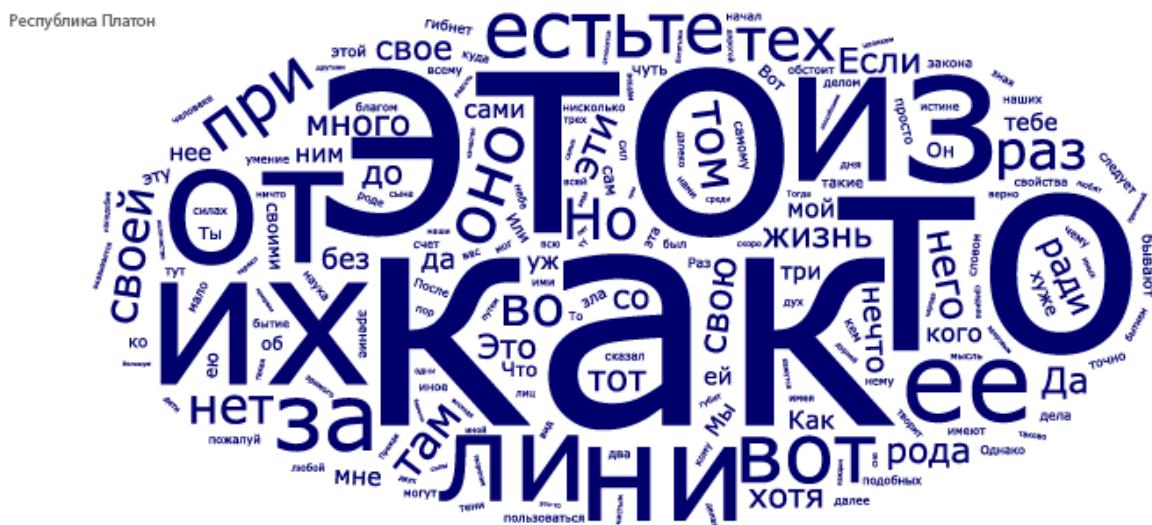
- **Описание:** Показывает минимальные и максимальные значения в данных.
- **Использование программистами:**
  - Анализ волатильности (например, температура за месяц).
- **Рекомендации:**
  - **Highcharts** для интерактивности, **Excel** для статических отчетов.
  - Используйте заливку между линиями для наглядности .

- **Доп. материалы:**
    - Диаграмма диапазонов.
    - Пример: Range Area Chart.
  - **Изображение визуализации**



## 2.24 Облако слов

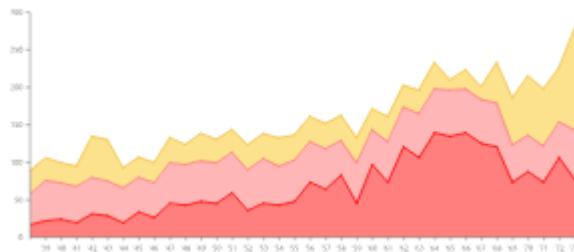
- **Описание:** Визуализирует частоту слов через размер шрифта.
  - **Использование программистами:**
    - Анализ текстовых данных (например, отзывы пользователей).
  - **Рекомендации:**
    - **WordCloud** (Python) для генерации, **Tableau** для интеграции с аналитикой.
    - Фильтруйте стоп-слова для повышения информативности .
  - **Доп. материалы:**
    - [Облако слов](#).
    - Пример: [Word Cloud in Python](#).
  - **Изображение визуализации**



## 2.25 Накопительная диаграмма с областями

- **Описание:** Показывает накопление данных по категориям.
  - **Использование программистами:**
    - Анализ структуры данных во времени.
  - **Рекомендации:**
    - **Plotly** для интерактивности, **Excel** для статических отчетов.
    - Используйте прозрачность для наложения областей.

- **Доп. материалы:**
  - [Накопительная диаграмма с областями.](#)
  - Пример: [Stacked Area Chart in D3.js](#).
- **Изображение визуализации**



## 2.26 Диаграмма «стебель-листья»

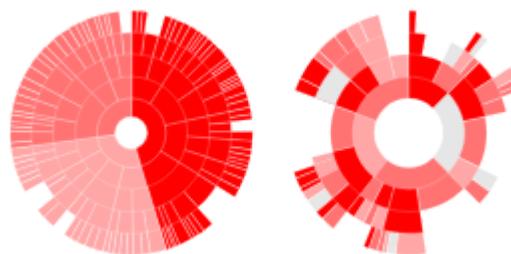
- **Описание:** Отображает распределение данных через цифры.
- **Использование программистами:**
  - Быстрый анализ небольших наборов данных.
- **Рекомендации:**
  - **Pandas** (Python) для обработки данных, кастомные скрипты.
  - Используйте сортировку для упрощения восприятия .
- **Доп. материалы:**
  - [Диаграмма «стебель-листья».](#)
- **Изображение визуализации**

Северное направление	Время	Южное направление
45 20 03	5	05 48
55 49 32 20 13 01	6	02 23 35 57
58 53 49 44 38 32 25 19 13 08 02	7	00 07 16 20 26 30 37 46 52 59
59 57 54 50 47 44 39 35 31 28 24 21 18 14 09 05 00	8	01 08 12 17 21 29 31 35 39 44 49 53 58
52 48 44 39 34 29 23 18 12 05	9	03 10 18 27 32 37 45 51 58
53 47 41 37 32 27 22 15 07	10	00 07 14 21 30 39 48 57
55 49 35 29 23 16 08 01	11	06 11 19 27 34 41 50 59
56 48 44 39 32 27 21 14 05	12	02 15 30 45 57
50 45 35 30 25 20 15 05	13	03 10 18 23 29 37 45 56
52 43 32 24 12 03	14	00 09 18 27 39 48 57
58 44 31 26 15 06	15	01 17 29 41 55
56 40 30 22 11	16	10 25 38 50
55 41 32 23 14 01	17	00 20 34 53
58 49 42 36 28 22 16 09	18	05 14 21 29 37 45 56
57 51 46 39 33 28 23 17 13 08 02	19	02 09 14 19 23 27 32 36 40 44 48 53 57
52 43 30 21 15 06	20	09 17 26 34 40 49 55
45 30 16 03	21	10 20 30 40 50
50 30 10	22	15 35 55

## 2.27 Диаграмма «Солнечные лучи»

- **Описание:** Визуализирует иерархию данных в виде лучей.
- **Использование программистами:**
  - Анализ древовидных структур (например, файловая система).
- **Рекомендации:**
  - **D3.js** для реализации, **Python Plotly** для простых задач.
  - Ограничите глубину иерархии для читаемости .
- **Доп. материалы:**

- [Диаграмма «Солнечные лучи».](#)
- Пример: [Sunburst Chart in D3.js](#).
- **Изображение визуализации**



## 2.28 Диаграмма Венна

- **Описание:** Показывает пересечения множеств через перекрывающиеся круги.
- **Использование программистами:**
  - Анализ пересечений данных (например, аудитория в соцсетях).
- **Рекомендации:**
  - **Matplotlib** для статических диаграмм, **Venn.js** для веба.
  - Используйте прозрачность для визуализации перекрытий .
- **Доп. материалы:**
  - [Диаграмма Венна](#).
  - Пример: [Venn Diagram in Python](#).
- **Изображение визуализации**

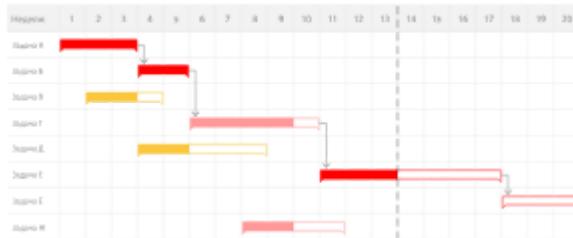


## 3. Шкала времени (диаграммы времени)

### 3.1 Диаграмма Ганта

- **Описание:** Визуализирует этапы проекта и сроки.
- **Использование программистами:**
  - Управление задачами в Agile, трекинг дедлайнов.
- **Рекомендации:**
  - **Jira** для интеграции с задачами, **Microsoft Project** для сложных проектов.

- Добавьте зависимости между задачами .
- **Доп. материалы:**
  - [Диаграмма Ганта.](#)
- **Изображение визуализации**



### 3.2 Хронологическая шкала

- **Описание:** Отображает события в хронологическом порядке.
- **Использование программистами:**
  - Визуализация истории версий ПО, логов изменений.
- **Рекомендации:**
  - [Timeline.js](#) для веба, [Excel](#) для простых задач.
  - Используйте интерактивные элементы для детализации .
- **Доп. материалы:**
  - [Хронологическая шкала.](#)
  - Пример: [Timeline in D3.js](#).
- **Изображение визуализации**

### Хронологическая шкала Первой Мировой войны

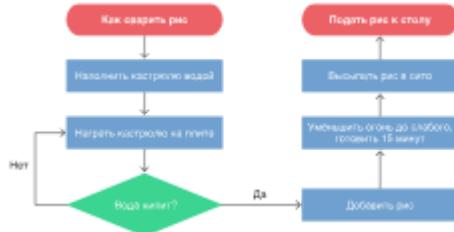


## 4. Блок-схемы (диаграммы визуализации процесса)

### 4.1 Блок-схема

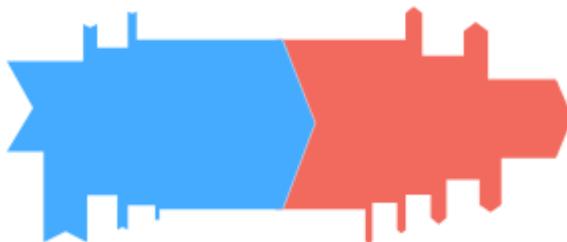
- **Описание:** Визуализирует алгоритмы или процессы.
- **Использование программистами:**
  - Проектирование логики приложений, документирование рабочих процессов.
- **Рекомендации:**

- **Draw.io** для простых схем, **Lucidchart** для командной работы.
- Используйте стандартные символы (ромбы для решений, прямоугольники для действий).
- **Доп. материалы:**
  - [Блок-схема](#).
- **Изображение визуализации**



## 4.2 Диаграмма Санкея

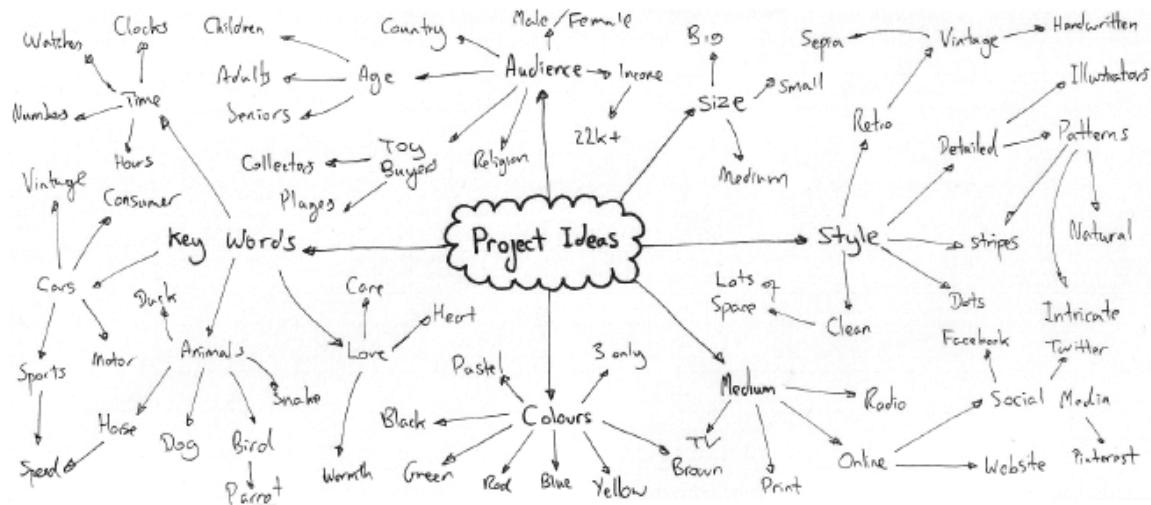
- **Описание:** Отображает потоки данных между узлами.
- **Использование программистами:**
  - Анализ перемещения ресурсов (например, трафик сайта).
- **Рекомендации:**
  - **D3.js** для кастомных решений, **Tableau** для готовых шаблонов.
  - Используйте цветовые коды для разных типов потоков.
- **Доп. материалы:**
  - [Диаграмма Санкея](#).
  - Пример: [Sankey Diagram in D3.js](#).
- **Изображение визуализации**



## 4.3 Мозговой штурм

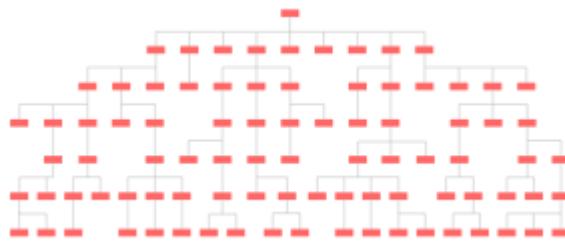
- **Описание:** Визуализирует идеи и их связи.
- **Использование программистами:**
  - Планирование архитектуры проекта, генерация идей.
- **Рекомендации:**
  - **Miro** для совместной работы, **XMind** для структурирования.
  - Добавьте метки и комментарии к узлам.
- **Доп. материалы:**
  - [Мозговой штурм](#).

- **Изображение визуализации**



#### 4.4 Древовидная диаграмма

- **Описание:** Отображает иерархические данные в виде дерева.
  - **Использование программистами:**
    - Визуализация структуры каталогов, зависимостей в коде.
  - **Рекомендации:**
    - D3.js для интерактивности, Treemap для компактного отображения.
    - Используйте раскрывающиеся узлы для детализации .
  - **Доп. материалы:**
    - Древовидная диаграмма.
    - Пример: Tree Diagram in Python.
  - **Изображение визуализации**



## 5. Матрицы

## 5.1 Календарь

- **Описание:** Визуализирует данные в формате календаря.
  - **Использование программистами:**
    - Отслеживание активностей по дням (например, коммиты в Git).
  - **Рекомендации:**
    - **FullCalendar.js** для веба, **Heatmap.js** для цветового кодирования.
    - Добавьте всплывающие подсказки для деталей .
  - **Доп. материалы:**

- Календарь.
- Изображение визуализации

МАРТ					АПРЕЛЬ								
ВС	СН	ВТ	СР	ЧТ	ПТ	СБ	ВС	СН	ВТ	СР	ЧТ	ПТ	СБ
				1	2	3				1	2	3	4
4	5	6	7	8	9	10	5	6	7	8	9	10	11
11	12	13	14	15	16	17	12	13	14	15	16	17	18
18	19	20	21	22	23	24	19	20	21	22	23	24	25
25	26	27	28		29		26	27	28	29	30	31	

## 5.2 Точечная матричная диаграмма

- Описание: Отображает данные в виде матрицы точек.
- Использование программистами:
  - Анализ корреляций между категориями.
- Рекомендации:
  - Seaborn (Python) для визуализации, Excel для простых задач.
  - Используйте размер точек для третьей переменной .
- Доп. материалы:
  - Точечная матричная диаграмма.
  - Пример: Scatterplot Matrix in Seaborn.
- Изображение визуализации



## 5.3 Тепловая карта (матрица)

- Описание: Цветовая интенсивность отражает значение данных.
- Использование программистами:
  - Анализ активностей пользователей (например, heatmap кликов).
- Рекомендации:
  - Seaborn (Python) для статических карт, D3.js для интерактивности.
  - Нормализуйте данные для точной цветовой интерпретации .
- Доп. материалы:
  - Тепловая карта.
  - Пример: Heatmap in D3.js.

- Изображение визуализации**



## 5.4 Древовидная карта

- Описание:** Визуализирует иерархические данные через вложенные прямоугольники.
- Использование программистами:**
  - Анализ структуры данных (например, размер файлов).
- Рекомендации:**
  - D3.js для кастомных решений, Treemap для автоматизации.
  - Используйте цвет для обозначения категорий, размер — для значений .
- Доп. материалы:**
  - [Древовидная карта](#).
  - Пример: [Treemap in Python](#).
- Изображение визуализации**



## 5.5 Расписание

- Описание:** Отображает данные в формате расписания.
- Использование программистами:**
  - Планирование задач (например, расписание деплоев).
- Рекомендации:**
  - FullCalendar.js для веба, Google Calendar API для интеграции.
  - Добавьте фильтры для категорий задач .
- Доп. материалы:**
  - [Расписание](#).

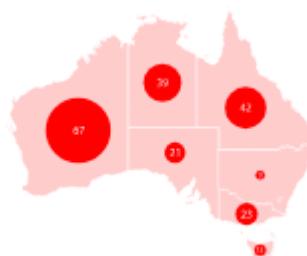
- **Изображение визуализации**

с понедельника по пятницу		Поезда из Ландан Ватерлеэ в Басингстоне																						
Станция		★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★		
Ландан Ватерлеэ	08:50	09:12	09:29	09:20	09:39	09:42	09:55	10:09	09:55	10:12	10:23	10:20	10:39	10:42	10:53	11:09	10:50	11:12	11:29	11:20	11:39	11:42		
Ваукхолл	08:55		09:24		09:29	09:46			09:54		10:22		10:29		10:54		11:24	11:34	11:54	11:59	12:00	12:12	12:25	
Цветной Юнитиби	08:50	09:19			09:29				09:56		10:22		10:29		10:55		11:19	11:29	11:46	12:00	12:02	12:19	12:28	
Барнслифрид	09:00				09:32				09:53		10:22		10:29		10:52		11:02	11:29	11:32	12:02	12:02	12:20	12:28	
Шамблесфорд	09:05				09:36				09:58		10:25		10:36		10:58		11:08	11:36	11:41	12:06	12:06	12:30	12:41	
Суарнти	09:15	09:31			09:41	09:44			10:02	10:11	10:14	10:30	10:42	10:44	11:02	11:11	11:14	11:30	11:44	12:00	12:11	12:14		
Бакк Хардинг	09:19	↓			09:48				10:07		10:22		10:48		11:18		11:21		11:51			12:18	12:48	
Манчестер-Таймс	09:21	09:38			09:51				10:11		10:25		10:54		11:08		11:24	11:37	11:54	12:07	12:21	12:51	12:54	
Шамблесфорд	09:24	09:41			09:54				10:21		10:34		10:57		11:12		11:31	11:41	12:00	12:11	12:27	12:41	12:57	
Барнсли & Нью Хэм	09:27				09:57				10:24		10:51		11:04		11:21		11:35	11:51	11:59	12:22	12:35	12:51	13:09	
Уест Бифлеэт	09:30	↓			09:51	10:03			10:18	10:28	10:33	10:40	10:48	10:50	11:09	↓	11:18	11:29	11:38	11:46	11:50	12:38	12:58	
Уест Бифлеэт	09:36	09:48			09:59	10:08																		
Уотинг	09:50	10:00			10:17				10:20	10:30	10:38		10:50	11:20		11:20	11:30	11:35	11:50	12:00		12:20	13:00	
Уорсливудон					10:29				10:33		10:50		11:03		11:23		11:53		12:03	12:23	12:33		12:50	13:09
Гуклфорд	09:50	10:04			10:29				10:33		10:50		11:03		11:23		11:53		12:03	12:23	12:33		12:50	13:25
Бромхэмбод	09:58	10:56			↓	10:26	10:38	↓		10:58	11:08		↓	11:25	11:36	↓	11:55	12:06	↓	12:25	12:36	↓	12:55	13:08
Нью Тайл					10:14				10:46		11:04		11:20		11:55		12:14		12:44			13:14	13:25	
Эдвардст					10:20				10:50		11:06		11:25		11:50		12:25		12:50			13:25	13:28	
Таркилд					10:25				10:56		11:08		11:22		11:55		12:29		12:58			13:02	13:07	
Болтвил					↓				11:03		11:10		11:24		11:57		12:27		13:02			13:16	13:37	
Альтане					10:37				↓		11:10		11:24		11:57		12:27		13:16			13:16	13:37	
Барборуорт	10:04				10:15	10:33	10:45		11:09		11:19	11:33	11:45		12:03		12:15	12:39	12:45			13:05		
Минифрид	10:08				10:19	10:38			11:06		11:19	11:38			12:08		12:19	12:39				13:08		
Росс	10:14				↓	10:44				11:18		11:46		12:18		12:28		12:48				13:17		
Басингстоне	10:28				10:51	10:55	10:58		12:28		11:31	11:59	11:57		12:28		12:31	12:56	12:58			13:26		

## **6. Карты и картограммы**

## 6.1 Пузырьковая карта

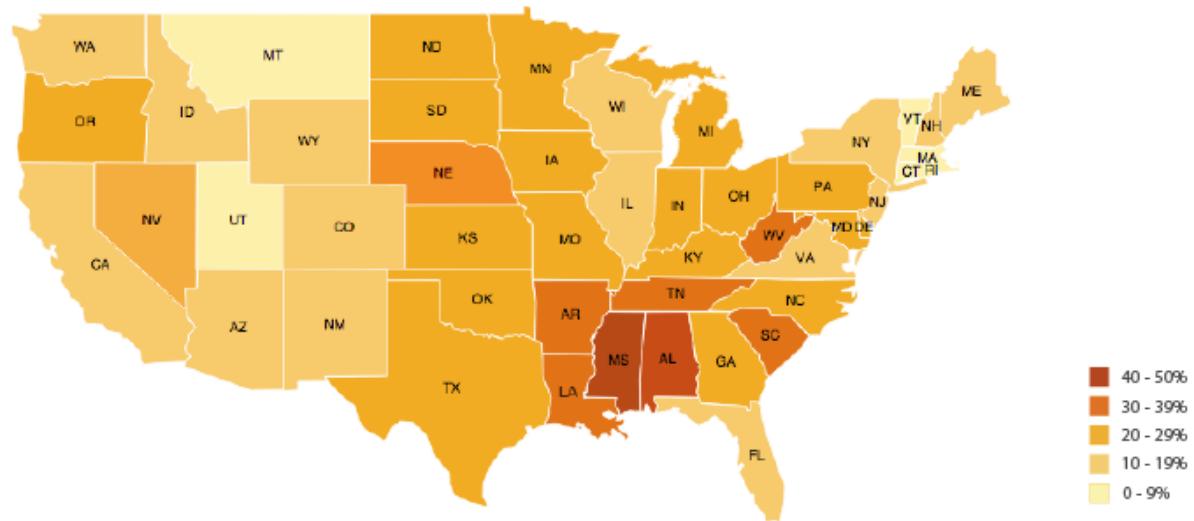
- **Описание:** Совмещает географию с размерами пузырьков.
  - **Использование программистами:**
    - Визуализация данных по регионам (например, ВВП).
  - **Рекомендации:**
    - Leaflet.js + **GeoJSON** для веба, **Tableau** для аналитики.
    - Используйте анимацию для динамических данных .
  - **Доп. материалы:**
    - Пузырьковая карта.
  - **Изображение визуализации**



## 6.2 Фоновая картограмма (хороплет)

- **Описание:** Закрашивает регионы в зависимости от значения показателя.
  - **Использование программистами:**
    - Отображение статистики по территориям (например, уровень безработицы).
  - **Рекомендации:**
    - D3.js для кастомных решений, Tableau для автоматизации.
    - Используйте градиентные цвета для плавных переходов .
  - **Доп. материалы:**
    - Хороплет.

- Пример: [Choropleth Map in D3.js](#).
- **Изображение визуализации**



### 6.3 Карта взаимосвязей

- **Описание:** Визуализирует связи между географическими точками.
- **Использование программистами:**
  - Анализ логистических маршрутов, миграции данных.
- **Рекомендации:**
  - Leaflet.js для базовых карт, Mapbox для кастомного дизайна.
  - Добавьте интерактивные линии для маршрутов .
- **Доп. материалы:**
  - [Карта взаимосвязей](#).
  - Пример: [Connection Map in D3.js](#).
- **Изображение визуализации**



### 6.4 Карта потоков

- **Описание:** Отображает перемещение данных или ресурсов.

- **Использование программистами:**
  - Визуализация сетевого трафика, логистических потоков.
- **Рекомендации:**
  - **D3.js** для анимации потоков, **Power BI** для бизнес-аналитики.
  - Используйте стрелки для указания направления .
- **Доп. материалы:**
  - [Карта потоков](#).
  - Пример: [Flow Map in D3.js](#).
- **Изображение визуализации**



## 6.5 Точечная карта

- **Описание:** Отображает данные через точки на карте.
- **Использование программистами:**
  - Анализ локаций событий (например, пользовательские активности).
- **Рекомендации:**
  - **Google Maps API** для интеграции, **Leaflet.js** для кастомных решений.
  - Используйте кластеризацию для больших наборов данных .
- **Доп. материалы:**
  - [Точечная карта](#).
- **Изображение визуализации**



## 6.6 Карта пробок в Москве, «Яндекс»

- **Описание:** Визуализирует текущую дорожную ситуацию.
- **Использование программистами:**

- Интеграция данных о пробках в приложения.
- **Рекомендации:**
  - **API Яндекс.Карт** для получения данных в реальном времени.
  - Добавьте слои для отображения альтернативных маршрутов .
- **Доп. материалы:**
  - Кarta пробок.
  - Документация: [Яндекс.Карты API](#).

**• Изображение визуализации**