

Диаграммы в контексте: инфопанели и презентации

Цель занятия

После освоения темы:

- вы узнаете интерактивные средства взаимодействия с диаграммами, паттерны организации диаграмм и правила создания презентаций на основе диаграмм;
- сможете создавать презентации на основе диаграмм с помощью движка Reveal.js, встраивая в него диаграммы из Yandex DataLens;
- сможете создавать интерактивные инфопанели, применяя функционал библиотеки Altair.

План занятия

1. [Интерактивные средства и связанные представления](#)
2. [Презентации на основе диаграмм. Общие практики](#)
3. [Подготовка инфопанелей и презентаций с помощью Yandex DataLens](#)
4. [Подготовка инфопанелей и презентаций с помощью библиотеки Altair](#)

Конспект занятия

1. Интерактивные средства и связанные представления

До сих пор мы строили статичные диаграммы, создавали их как картинку и мало говорили о средствах взаимодействия пользователя с диаграммами или про то, как несколько диаграмм объединить в историю.

Виды диаграмм

Даже на основе одной статичной картинке можно создать историю, поскольку есть возможность выделить на этой картинке то, на что следует обратить внимание пользователя. Если диаграмма одна, мы можем поместить ее внутрь текста и уже текстом прописать, как взаимодействовать с этой диаграммой пользователю.

Диаграмму можно наделить интерактивом, если она не на бумаге, а в вебе или внутри программного инструмента. Тогда мы даем пользователю возможность самому управлять повествованием — фильтровать данные на диаграмме, масштабировать и т. п. При этом пользователь сам определяет, как взаимодействовать с диаграммой, то есть сам управляет историей.

Можно взять не одну диаграмму, а несколько. Тогда можно организовать **скроллитинг** — веб-страницы, имеющие линейное повествование. Механизм очень похож на презентацию по слайдам, но данные разбиты не по нескольким экранам, а собраны на одной веб-странице, по которой пользователь «скроллит». При этом возможен некоторый интерактив.

Презентация по слайдам — стандартное «старое» средство демонстрации данных. Предполагается, что есть докладчик, который голосом комментирует происходящее и демонстрирует соответствующие слайды, чтобы визуально поддержать свое повествование. Докладчик имеет полный контроль над тем, что показывает, и мало использует интерактив, поскольку, как правило, он заранее продумывает историю. Иногда лектор голосом может сделать дополнительные пояснения и ответить на вопросы.

В случае **дашбордов (инфопанелей)** историю полностью создает сам пользователь. Дашборды нужны для того, чтобы находить в данных новые истории, определять на месте, что интересно узнать в данный момент, и обращать внимание на соответствующие диаграммы внутри дашборда. В дашбордах обязательно должен присутствовать интерактив, поскольку автор дашборда редко контролирует, на каких данных в определенный момент этот дашборд будет построен, что именно привлечет внимание пользователя. Пользователь должен иметь достаточно средств, чтобы взаимодействовать с диаграммами.

Вид	Сколько диаграмм	Кто управляет повествованием
Статичная диаграмма	Одна	—
Интерактивная диаграмма	Одна	Пользователь
Скроллителлинг	Несколько	Автор
Презентация по слайдам	Несколько	Автор
Дашборд (инфопанель)	Несколько	Пользователь

Средства взаимодействия пользователя с диаграммами

Рассмотрим, какие интерактивные возможности существуют и что нужно пользователю чаще всего. Приведенная ниже таблица — своеобразный перечень того, что можно предложить пользователю в качестве интерактива.

Класс	Средство	Что меняется
Выбор (selection)	Выбор по щелчку, по близости к указателю, ...	Представление (view)
Навигация (navigation)	Панорамирование (pan), масштабирование (zoom)	Представление (view)
Перерасположение (spatial arrangement)	Перетаскивание (click&drag), сортировка (sorting)	Представление (view)
Изменение отображения данных в визуальные атрибуты	Переключение вида диаграммы, формы глифов, цветовой палитры, ...	Отображение (mapping)
Агрегирование, (aggregation) фильтрация (filtering)	Выбор из множества, отсечка (threshold), ...	Отображаемые данные (data)

Выбор (selection) используется в таблицах, чтобы при наведении на ячейку курсора подсветить ее.

Навигация включает в себя два элемента: панорамирование (pan) и масштабирование (zoom). Данная возможность предполагает, что мы можем перемещаться по графику, двигать его, увеличивать масштаб. Также можно выделять элементы и подсвечивать их. Сами данные, которые мы отображаем, не меняются.

Масштабирование включает в себя еще **семантическое масштабирование (зумирование)**, которое популярно в картографических системах. При семантическом масштабировании в зависимости от масштаба мы показываем разные версии. Например, при большом масштабе показываем только подписи к городам, при приближении показываем названия улиц внутри города, при еще большем приближении — номера домов. Так реализованы Google- и Яндекс-карты.

Следующий вариант взаимодействия с диаграммой — возможность **перерасположить** элементы на этой диаграмме. Например, переместить столбики внутри столбчатой диаграммы. В этом случае мы имеем возможность пересортировать данные.

Примером **изменения отображения данных в визуальные атрибуты** может быть изменение палитры на тепловой карте.

В случае **агрегирования или фильтрации** данных мы меняем уже сами данные, которые отображаем. Например, выбираем отрезок времени, за период которого нужно отобразить данные. Часто эта возможность предоставляется именно пользователю, поскольку заранее невозможно предугадать, в какой степени агрегации пользователь заметит закономерности.

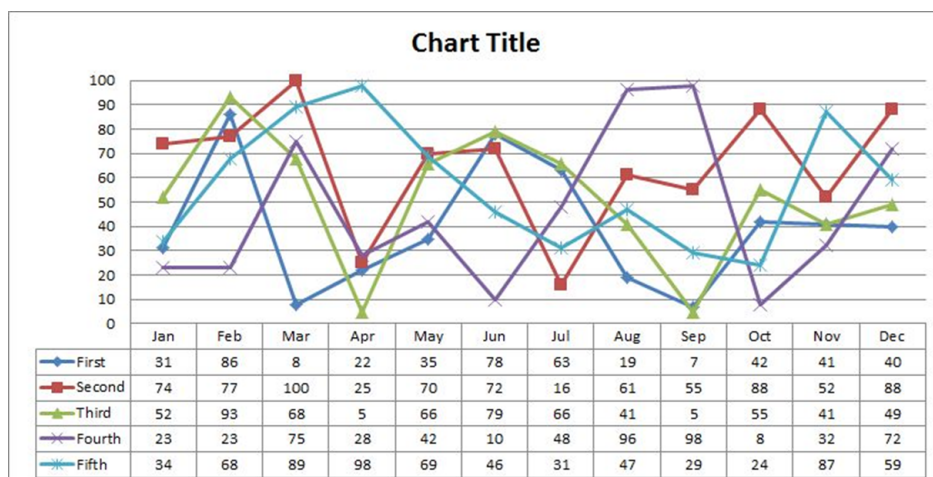
Паттерны организации пар диаграмм

Существует три стандартных паттерна организации пар диаграмм внутри дашбордов:

- диаграмма + таблица;
- обзор + детали;
- «что-где-когда».

Диаграмма + таблица. Часто бывает, что пользователю недостаточно иметь данные, представленные диаграммой, ему необходимо видеть конкретные цифры.

На графике не всегда удобно подписать все числовые значения. В этом случае пользователю можно дать возможность одновременно видеть исходную базу данных и диаграмму, построенную по этим данным. Максимальная польза от такого дашборда достигается, когда мы наделяем его интерактивом — например, можно подсвечивать данные в таблице и соответствующую им кривую на графике. Тогда диаграмму и таблицу мы можем называть связанным представлением.

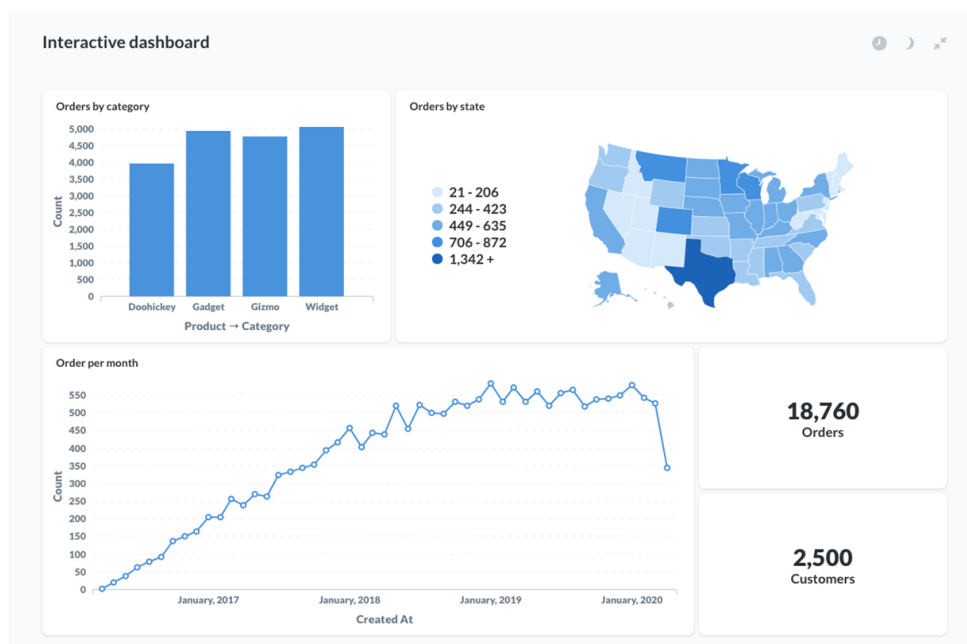


Обзор + детали. Очень часто в диаграмме недостаточно одного фиксированного масштаба, чтобы изучить карту или временной ряд. В этом случае можно предоставить пользователю возможность видеть одновременно данные на разных частях диаграммы или дашборда в разном масштабе.



«Что-где-когда». В этом паттерне мы включаем на дашборд несколько разных диаграмм, которые описывают данные с разных сторон.

Часто в базах данных слишком много информации о бизнесе компании. Данные могут содержать в себе одновременно и количественные показатели, и геопозиционную информацию, и темпоральные атрибуты. Это все не покажешь на диаграмме одного и того же типа. В этом случае делают несколько разных видов диаграмм, каждая из которых отвечает на свой вопрос про данные.



Для таких связанных между собой диаграмм можно добавить интерактив. Например, при нажатии на столбик на столбчатой диаграмме (выбор категории товаров в примере на рисунке выше) можно получить фильтрацию данных, на которых строятся другие графики.

2. Презентации на основе диаграмм. Общие практики

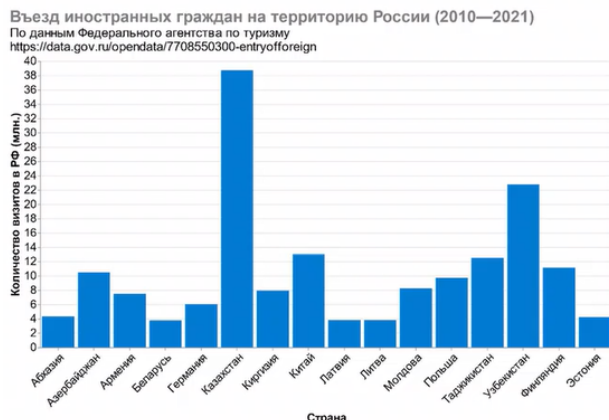
Рассмотрим, как можно делать презентации на основе диаграмм.

Правила создания презентации

- **Соответствие целям.** Каждый слайд делается с конкретной целью — донести какое-то сообщение. Редко когда, особенно в корпоративном контексте, делают презентацию, чтобы о чем-то поговорить.
- **Соответствие аудитории.** Сообщение на слайде должно быть сформировано для конкретной аудитории. Поэтому перед созданием презентации нужно продумать не только цели, но и для какой они аудитории.
- **Презентация = слайд + голос + ...** Презентация — это не только слайды, но и спикер. Бессмысленно пытаться поместить весь текст на слайд.
- **Использование стандартных советов по дизайну.**

Теперь рассмотрим антипример того, как не надо делать слайд для презентации.

Разбор неверного слайда презентации



- На слайде представлена статистика по въезду **иностранных граждан** в РФ по данным **Федерального агентства по туризму**.
- Согласно этим данным, лидирует **Казахстан** с более чем **38 млн.** пересечений гражданами границы за 12 лет. Некоторые другие страны тоже не отстают. Например, подтягивается Узбекистан. Китай входит в тройку лидеров.

Ошибки слайда:

1. На картинке в тексте несколько сообщений, которые следуют из столбчатой диаграммы. Сообщения даны так, будто это не презентация, а письменный отчет.

Как исправить. Нужно оставлять минимальный объем текста (ключевые слова, сокращенные тезисы), остальное (вводные слова, вспомогательные факты) проговаривать голосом. Согласно исследованиям, во время презентации люди не воспринимают одновременно голос докладчика и текст на экране.

2. Излишнее количество цветов, шрифтов, начертаний на слайде.

Как исправить. В рамках одного слайда и даже всей презентации рекомендуется использовать не больше 2–3 шрифтов, включая различные их начертания. Выделение слов не понадобится, если поместить на слайд только ключевые слова или тезисы.

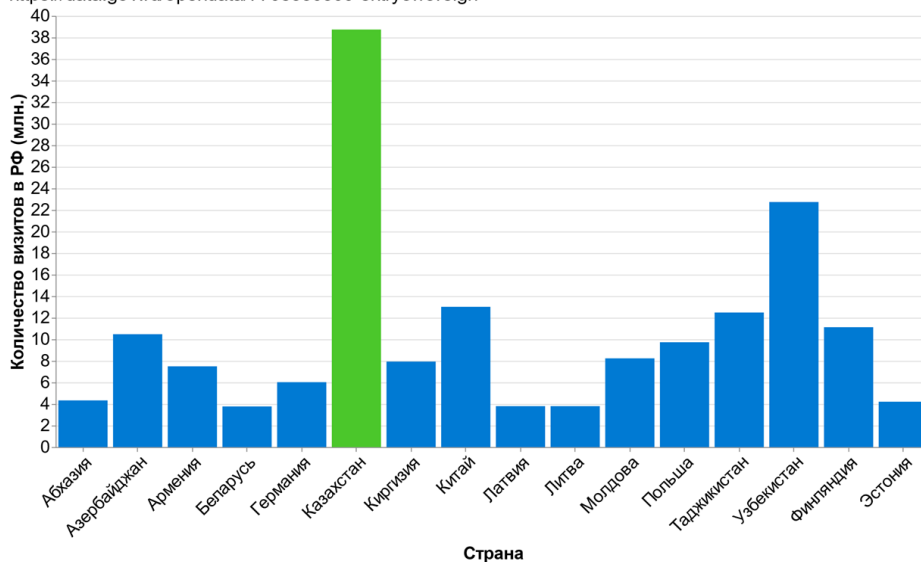
3. Не должно быть сразу нескольких сообщений. Один слайд — одно сообщение или часть этого сообщения.

Чтобы исправить приведенный антипример, нужно показать одну диаграмму и убрать на ней лишние детали — данные об источнике информации. Во время презентации проговорить голосом, откуда мы эти данные получили, или сделать пояснение на отдельном слайде.

Можно работать с диаграммой на последовательных слайдах. Если мы хотим заявить что-то про диаграмму, можно подсветить определенный элемент, о котором идет речь. По сути, мы должны построить несколько вариантов одной и той же диаграммы.

Въезд иностранных граждан на территорию России (2010—2021)

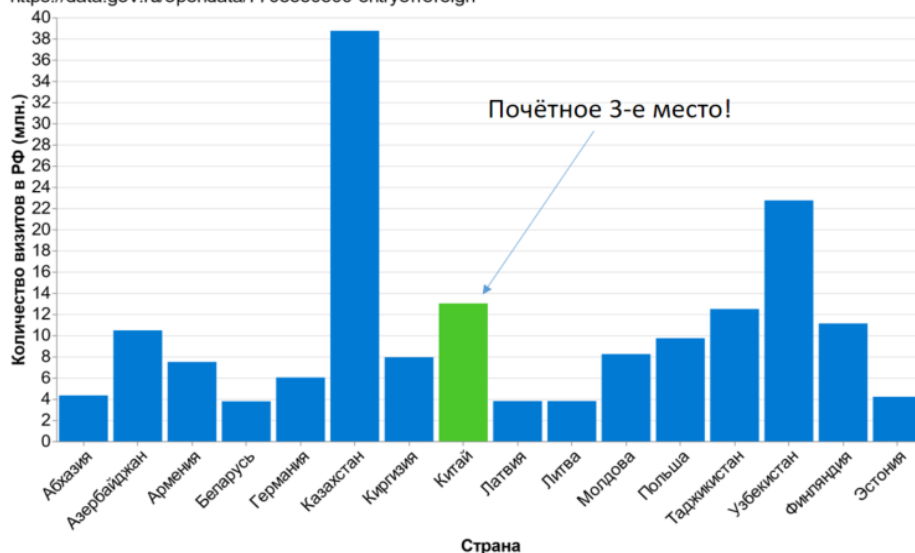
По данным Федерального агентства по туризму
<https://data.gov.ru/opendata/7708550300-entryofforeign>



Необязательно описывать данные на диаграмме текстом рядом, достаточно это отметить прямо на ней:

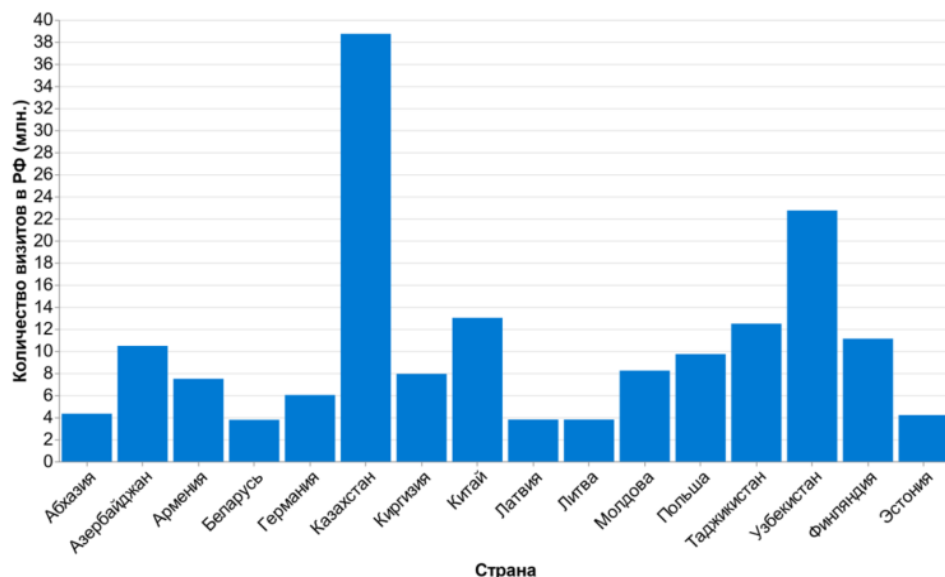
Въезд иностранных граждан на территорию России (2010—2021)

По данным Федерального агентства по туризму
<https://data.gov.ru/opendata/7708550300-entryofforeign>



Сообщение про диаграмму можно поместить в заголовок диаграммы или слайда. Теперь зритель может сразу считать все нужные текстовые сообщения:

Граждане Казахстана — самые активные!



Таким образом, мы избавляемся от второстепенных объектов и текста на диаграмме. Мы даем возможность зрителям быстро пробежаться по диаграмме, считать все текстовые сообщения и больше не отвлекаться на нее и слушать докладчика.

3. Подготовка инфопанелей и презентаций с помощью Yandex DataLens

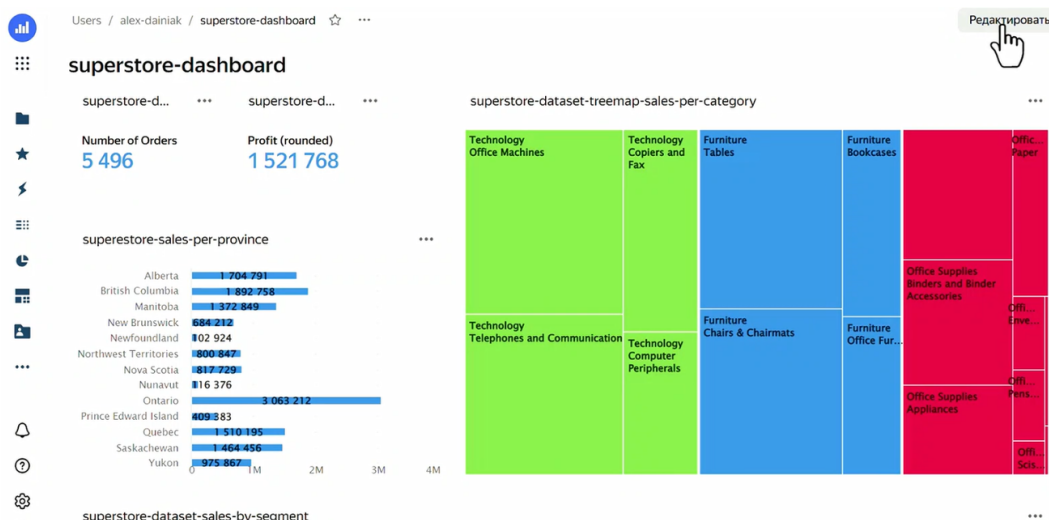
Познакомимся с техническими возможностями объединения отдельных диаграмм в дашборды и презентации.

Ранее мы разбирали визуальный инструмент [Yandex DataLens](#), который не требует программирования для построения графиков, а также изучили [библиотеку Altair](#). В этот раз мы исследуем эти два инструмента с точки зрения возможностей создания графиков для презентации дашбордов.

Для примера будем использовать датасет с заказами в крупном магазине офисной техники *superstore-dataset*. На основе этого датасета мы построим графики, они будут доступны в разделе «Чарты». Отдельная ячейка дашборда может демонстрировать не только диаграммы, но и одно число, которое может быть очень важным показателем для приведенных данных.

Возможности построения дашбордов в Yandex DataLens

При создании дашборда с нуля необходимо войти в режим редактирования:



Чтобы подгрузить новый график, нужно нажать «Добавить» и выбрать график. Если график был построен заранее, то можно вставить на него ссылку и добавить на дашборд. В настройках можно изменить название графика или скрыть его, а также поменять название всего дашборда.

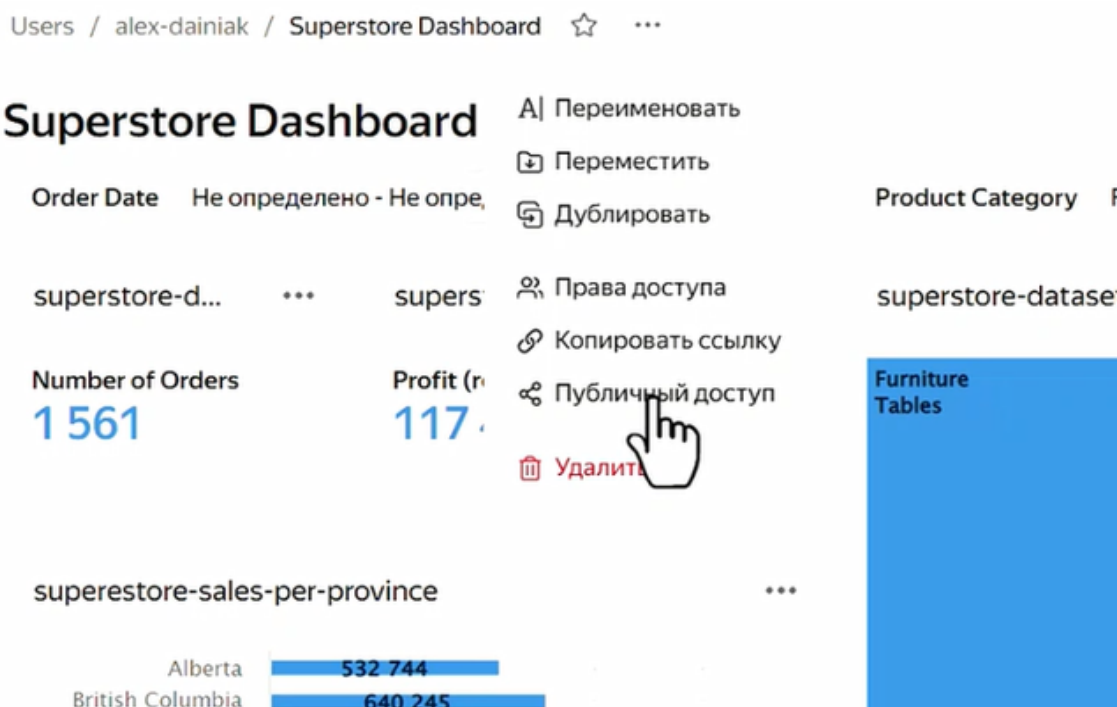
В дашборде также можно менять расположение графиков, их размер, добавлять фильтры. Фильтры, как правило, размещают в верхней части дашборда, чтобы пользователь сразу имел представление, как работать с информацией.

В рассматриваемом примере можно добавить на дашборд фильтры по категории продукта и по дате. Таким образом, появится возможность взаимодействовать с графиком, выбрать отдельные покупательские сегменты и получить общую картину.

Чтобы добавить фильтры по разным категориям, нужно добавить на дашборд селектор, выбрать датасет, по которому строили все графики, и установить, как взаимодействовать с селектором.

Если мы не хотим, чтобы все графики перестраивались под фильтр, а селектор относился только к определенному графику, нужно зайти в раздел «Связи», в котором перечислены все связи этого селектора с графиками. Если мы не хотим, чтобы селектор взаимодействовал с определенным графиком, нужно нажать «Игнорировать» для данной связи.

Как и для отдельного графика после построения дашборда можно скопировать ссылку на дашборд, либо открыть публичный доступ по ссылке. При этом все элементы дашборда, включая датасет, на основе которого он был построен, также будут доступны.



Для построения графиков для презентации есть два варианта:

- экспортировать графики (как рисунки) и вставить в презентацию;
- использовать более продвинутую технологию и сохранить интерактивность графиков.

Для сохранения интерактивности графиков воспользуемся Java-HTML-скриптом **Reveal.js**.

Работа с Java-HTML-скрипт в Reveal.js

Движок Reveal.js имеет свою [страницу](#) с документацией. Здесь же показаны примеры того, что он умеет: анимации смены слайдов, автоанимации, работа с программным кодом с подсветкой и др.

Воспользуемся движком Reveal.js для построения презентации. Мы создали специальный [шаблон](#) в [Replit](#).

Любая презентация в Reveal.js — это файл HTML, который содержит теги — ключевые слова языка в угловых скобках. Файл в Reveal.js состоит из нескольких частей, начинающихся открывающимся тегом (например, тег заголовка `<head>`) и заканчивающихся закрывающимся тегом (в данном примере — `</head>`). Внутри тега могут содержаться другие теги.

Строка, которая позволяет выбрать тему оформления презентации:

```
<link rel="stylesheet" href="dist/theme/black.css">
```

Основное содержимое слайдов — текстовый и другой контент — находится в разделе `<body>`.

Внутри каждого тега `<section>Slide 1</section>` прописывается содержимое слайда. Слайды содержат не слишком много элементов. Это может быть заголовок, текстовые списки, пункты, графики.

Заголовки на слайдах пишутся так: `<h1></h1>` — заголовок всей презентации, `<h3></h3>` — подзаголовок третьего уровня.

А так создаются список и пункты на слайде:

```
<h3>Слайд 1</h3>
<ul>
  <li>Тезис 1</li>
  <li>Тезис 2</li>
  <li>Тезис 3</li>
</ul>
```

Результат выполнения написанного кода:

СЛАЙД 1

- Тезис 1
- Тезис 2
- Тезис 3

Текст по умолчанию центрируется, но это можно поменять — выставить соответствующие стили.

Если мы хотим, чтобы тезисы появлялись по клику мышки, добавим класс `fragment` к каждому пункту списка:

```
<h3>Слайд 1</h3>
<ul>
  <li class="fragment">Тезис 1</li>
  <li class="fragment">Тезис 2</li>
  <li class="fragment">Тезис 3</li>
</ul>
```

Существуют разные виды `fragment`, анимацию можно настраивать. Поместим во второй слайд графики из Yandex DataLens:

```
<section>
  <h3>Слайд 2</h3>
  <iframe class="r-stretch"
    src="https://datalens.yandex/ultbrkcsga0k">
  </iframe>
</section>
```

Чтобы получить ссылку для вставки графика:

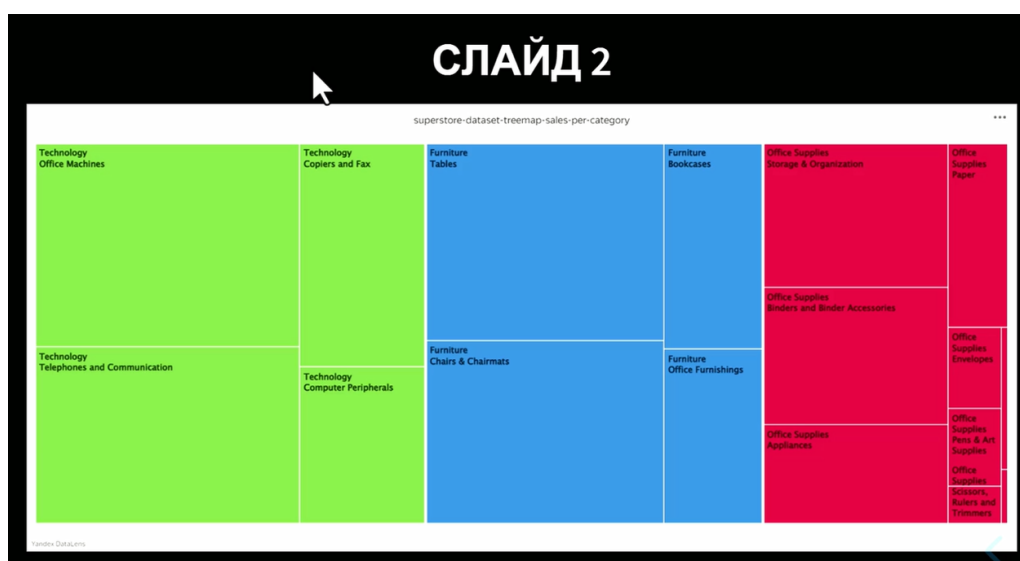
1. Заходим в «Чарты» на Yandex DataLens.
2. Находим нужный график и экспортируем его в публичный доступ.
3. Выставляем доступ по ссылке.
4. Копируем полученную ссылку.

5. Переходим в презентацию Reveal.js и копируем ссылку в элемент `<iframe>`.

Дополнительно у графика установлен стилевой класс `class="r-stretch"`, который заставляет элемент занять все отведенное ему пространство на слайде:

```
<iframe class="r-stretch" src="https://ссылка_на_график">
```

Результат:



Интерактивность графика при этом сохраняется, по нему можно щелкать мышкой и смотреть категории.

У презентации HTML есть преимущество — если меняется датасет или стиль графика, а также его параметры в Yandex DataLens, то этот график обновится по ссылке в презентации.

4. Подготовка инфопанелей и презентаций с помощью библиотеки Altair

Посмотрим теперь, как в [библиотеке Altair](#) строить графики для презентаций и объединять их в инфопанели.

Построение графиков

Подгрузим ранее использованный датасет *superstore-dataset* из Excel-файла, проверим, что датасет подгрузился корректно:

```
import pandas as pd
import altair as alt

alt.themes.enable('dark')
alt.data_transformers.disable_max_rows();
alt.warnings.simplefilter(action='ignore')

superstore_data = pd.read_excel('superstore.xlsx')

superstore_data = superstore_data.set_index('Row ID')
superstore_data['Order Date'] =
pd.to_datetime(superstore_data['Order Date'])
superstore_data['Ship Date'] =
pd.to_datetime(superstore_data['Ship Date'])
superstore_data.head(1)
```

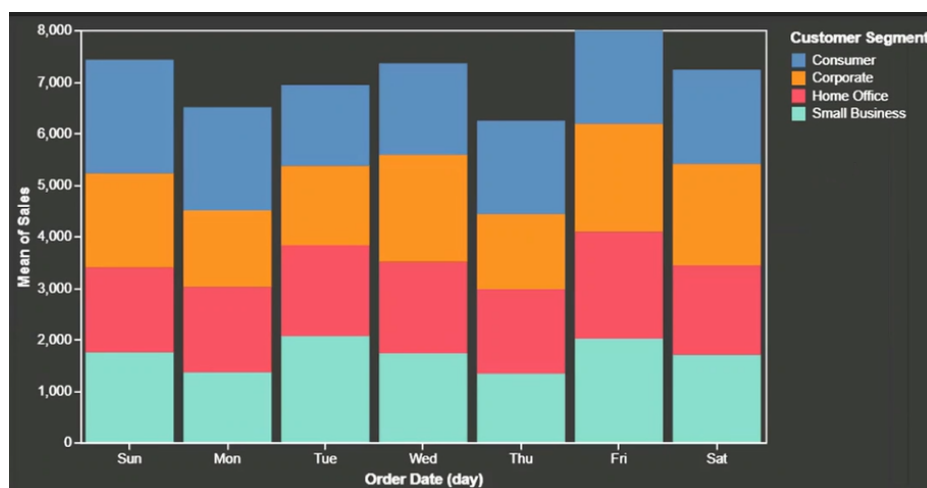
Далее переходим к построению графиков. В примере мы построим столбчатую диаграмму, показывающую продажи по дням недели:

```
selection = alt.selection_multi(fields=['Customer Segment'],
bind='legend')

chart = alt.Chart(
    superstore_data
).mark_bar(
).add_selection(
    selection
).encode(
    x='day(Order Date):O',
    y=alt.Y('mean(Sales)', axis=alt.Axis(grid=False)),
    color=alt.condition(selection, 'Customer Segment',
alt.ColorValue('darkgrey')),
    opacity=alt.condition(selection, alt.value(1), alt.value(0.3))
).properties(
    width=500
)

chart
```


Дополнительно на график добавлена возможность выбирать по легенде покупательский сегмент. Результат:



Рассмотрим, как вставить график в презентацию с сохранением интерактива.

Графики можно сохранять в json-виде:

```
with open('bar-chart.json', 'w') as f:
    f.write(chart.to_json())
```

Теперь вставим график в имеющуюся презентацию в Reveal.js. Подгружаем файл bar-chart.json в папку charts. Для вставки графика используем библиотеку Vega-Embed. Полностью код для презентации будет следующим:

```
<html>
<body>
    <div class="reveal">
        <div class="slides">
            <section>
                <h1>Main Title</h1>
            </section>
            <section>
                <div data-chart="charts/bar-chart.json"
                    class="r-stretch">
                </div>
            </section>
            <section>
                Slide 2
            </section>
        </div>
    </div>
</body>
```

```
        </section>
      </div>
    </div>
    <script src="dist/reveal.js"></script>
    <script>
      Reveal.initialize({
        // height: 720,
        // width: 1280,
        // minScale: 1,
        // maxScale: 1,
        plugins: []
      });

      Reveal.addEventListener('ready', function(event) {
        let options = {
          renderer: 'svg',
          theme: 'dark',
          actions: false
        };
        let vegaDivs =
document.querySelectorAll('div[data-chart]');
        for(let div of vegaDivs){
          vegaEmbed(div, div.dataset.chart, options)
        }
      });
    </script>
  </body>
</html>
```

Для вставки графика на конкретный слайд внутри соответствующей секции необходимо прописать аналогичный код:

```
<div data-chart="charts/bar-chart.json" class="r-stretch">
```

В результате получаем ту же самую столбчатую диаграмму на слайде презентации. Весь интерактив на диаграмме также сохраняется. При изменении графика в ноутбуке, его нужно пересохранить в json-формат и заново подгрузить в Reveal.js.

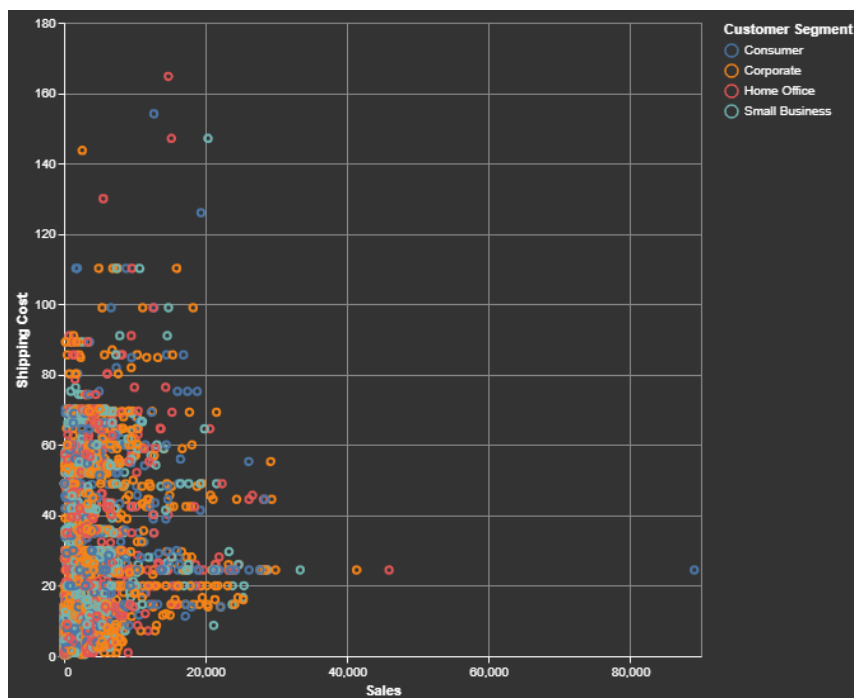
Объединение графиков и добавление элементов

Возьмем еще один график — диаграмму рассеяния, показывающую корреляцию продаж со стоимостью доставки. Диаграмма имеет кликабельную легенду:

```
segment_selection = alt.selection_multi(fields=['Customer  
Segment'], bind='legend')
```

```
alt.Chart(
    superstore_data
).mark_point(
).add_selection(
    segment_selection
).encode(
    alt.X('Sales', axis=alt.Axis(tickCount=6)),
    alt.Y('Shipping Cost', axis=alt.Axis(tickCount=10)),
    color='Customer Segment',
    opacity=alt.condition(segment_selection, alt.value(1),
alt.value(0.1))
).properties(
    width=300,
    height=300
)
```

Построенная диаграмма:



На диаграмме есть точки с очень большим уровнем продаж (правая синяя) и, чтобы поместить все точки на одно общее полотно, приходится брать очень большой масштаб. Все остальные точки оказались в куче слева.

Чтобы отфильтровать в данных выбросы, можно добавить «отсечку». То есть пользователь сможет выбирать диапазон, по которому он захочет посмотреть продажи.

Мы создадим элемент — слайдер, к которому привяжем выбор «отсечки» для данных (`selector`). Также введем элемент `brush`, который позволяет выбирать на графике прямоугольные области, и который нам пригодится в дальнейшем:

```
segment_selection = alt.selection_multi(fields=['Customer
Segment'], bind='legend')
brush = alt.selection(type='interval', resolve='global')
slider = alt.binding_range(min=0, max=100000, step=1000,
name='Cutoff: ')
selector = alt.selection_single(
    name='SalesCutoff',
    fields=['cutoff'],
    bind=slider,
    init={'cutoff': 30000}
)
```

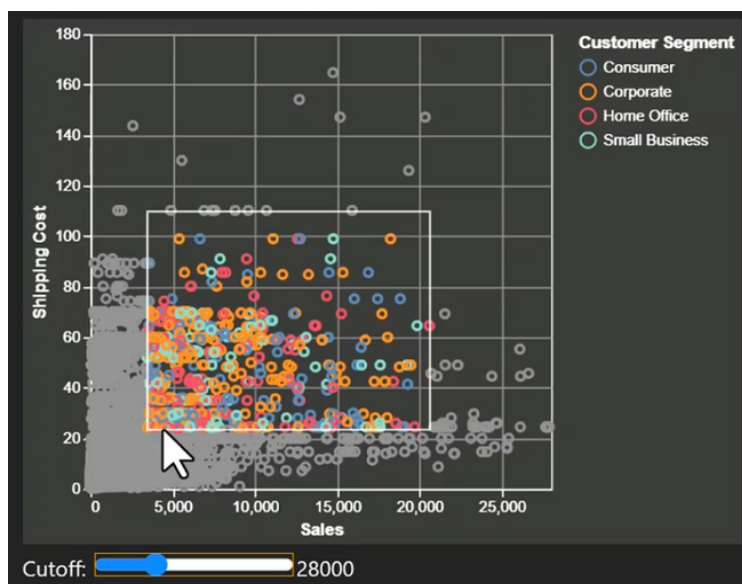
После чего `selector` добавляем на график и указываем, как он действует. Нам необходимо убрать точки из данных для того, чтобы шкала была перестроена. Для этого добавим преобразование фильтрации `transform_filter`:

```
chart = alt.Chart(
    superstore_data
).mark_point(
).add_selection(
    selector
).transform_filter(
    alt.datum['Sales'] <= selector.cutoff
).add_selection(
    segment_selection
).add_selection(
    brush
).encode(
    alt.X('Sales', axis=alt.Axis(tickCount=6)),
    alt.Y('Shipping Cost', axis=alt.Axis(tickCount=10)),
    color=alt.condition(brush, 'Customer Segment',
alt.ColorValue('gray')),
    opacity=alt.condition(segment_selection, alt.value(1),
alt.value(0.3))
).properties(
    width=300,
    height=300
)
chart
```

На графике появился элемент `Cutoff`, с помощью которого можно отобразить лишь требуемые данные:



И можно выбирать прямоугольные интервалы:

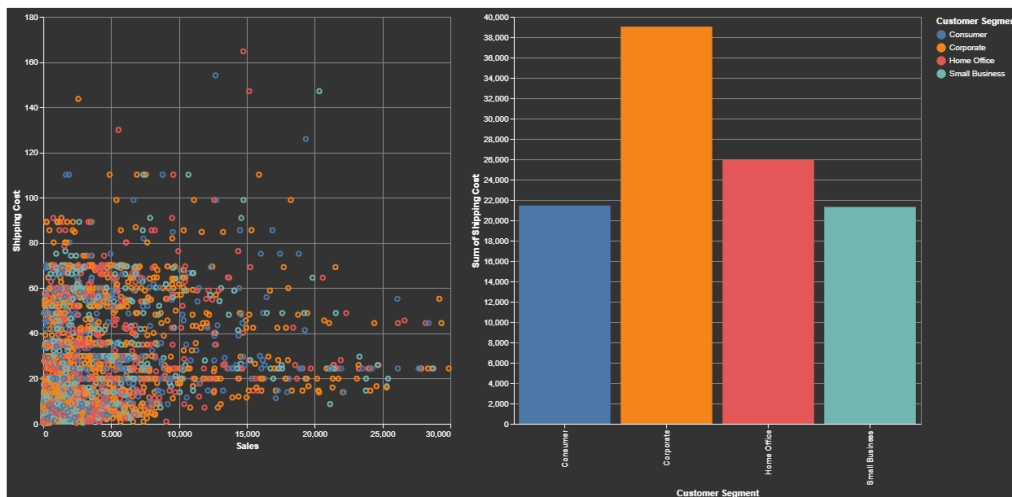


Посмотрим, как объединять графики. Создадим еще один график — столбчатую диаграмму, данные будут отсекаются по тому же селектору. Кроме того, на столбчатую диаграмму будут попадать данные, выбранные с помощью селектора `brush`. На графике мы покажем общие расходы на доставку по покупательским сегментам.

```
chart2 = alt.Chart(
    superstore_data
).mark_bar(
).transform_filter(
    alt.datum['Sales'] <= selector.cutoff
).transform_filter(
    brush
).encode(
    x='Customer Segment',
    y='sum(Shipping Cost)',
    color='Customer Segment',
    opacity=alt.condition(segment_selection, alt.value(1),
alt.value(0.3))
).properties(
    width=300,
    height=300
)
```

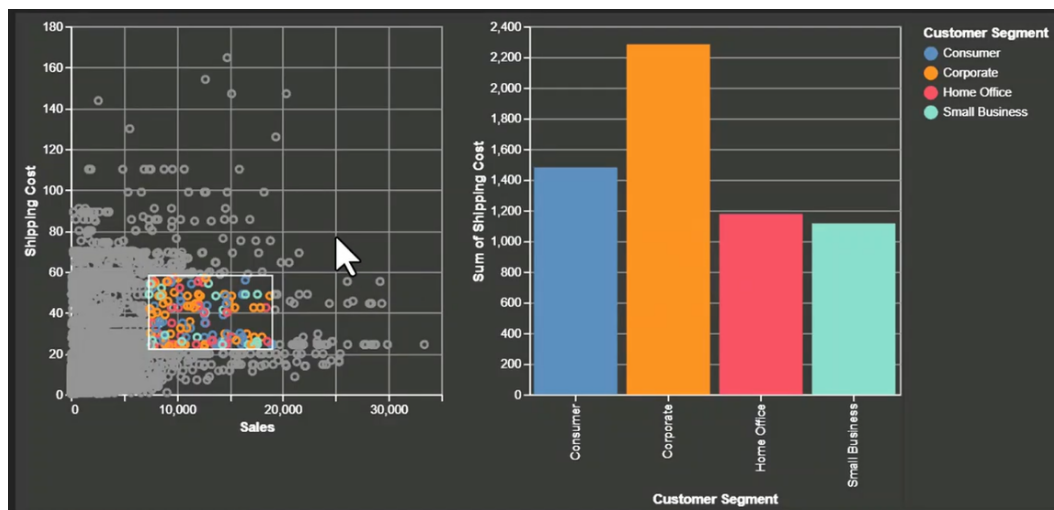
Если записать графики через горизонтальную черту «|», они встанут рядом горизонтально друг с другом:

```
chart | chart2
```



Аналогично можно сформировать дашборд из четырех графиков — объединить их по парам горизонтально, а затем вертикально. В рассматриваемом примере мы получаем небольшой дашборд:

Сохраняется элемент `Cutoff`, и он распространяется на все графики. Сохраняется кликабельность легенды, она отражается на обоих графиках. И самое важное: взаимодействие с одним графиком отражается на втором. То есть теперь выделяя область данных на левом графике, меняются значения столбчатой диаграммы справа.



Таким образом, в Altair можно создавать полноценные панели из нескольких графиков. Всю такую информационную панель при желании можно встроить в презентацию на Reveal.js, но так делать не рекомендуется, поскольку в презентациях действует стандартное правило: один слайд — один график.

Дополнительные материалы для самостоятельного изучения

1. [Визуализация данных — Yandex DataLens | Yandex Cloud — Сервисы](#)
2. [Vega-Altair: Declarative Visualization in Python — Vega-Altair 5.0.0dev documentation \(altair-viz.github.io\)](#)
3. [The HTML presentation framework | reveal.js \(revealjs.com\)](#)