Визуализация данных

Киреев В.С.,
к.т.н., доцент

Москва, 2023

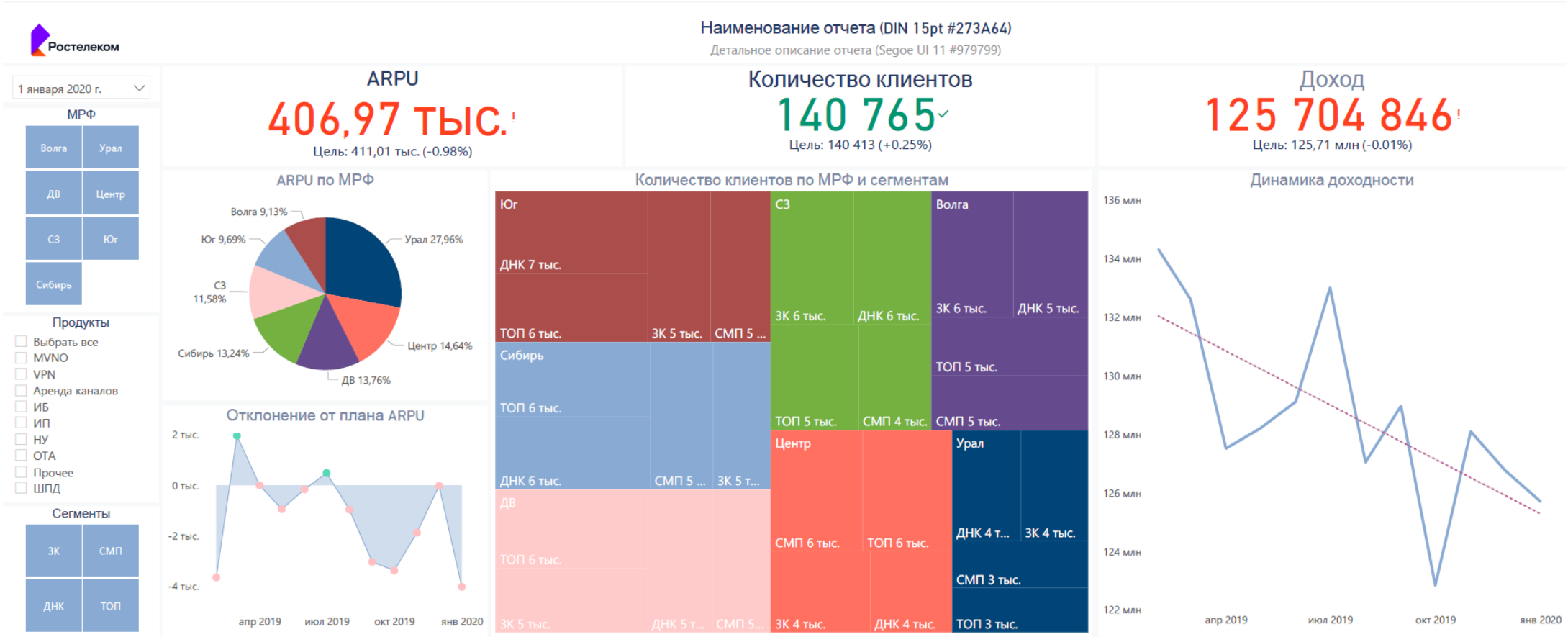
Инструменты сторителлинга на данных

Дэшборды. Понятие

Дашборд (с англ. Dashboard — «приборная панель», также иногда называется «панель индикаторов») — графический пользовательский интерфейс для демонстрации KPI определенного процесса, бизнес-цели, текущих процессах в компании.

Далее будем использовать термин «дэшборд» для обозначения комбинации нескольких визуализаций данных и текстовых дисплеев, которые часто взаимосвязаны/интерактивны и обычно располагаются в виде одностраничного макета.

Дэшборды. Пример



Дэшборды. Отличия от отчетов

- дэшборды - интерактивные и динамичные — данные обновляются автоматически, можно фильтровать и менять периоды;
- дэшборды используются регулярно, тогда как отчеты каждый раз нужно формировать заново;
- дэшборды нужны, чтобы контролировать ситуацию и принимать решения, отчеты — чтобы показать результаты работы руководству.

Дэшборды. История термина

Термин «приборная панель» (dashboard) происходит от доски (dash), которая была встроена в повозки, чтобы блокировать грязь, которую отбрасывала лошадь. Когда кареты стали автомобилями, эта доска оставалась актуальной, чтобы блокировать грязь, наносимую передним колесом.

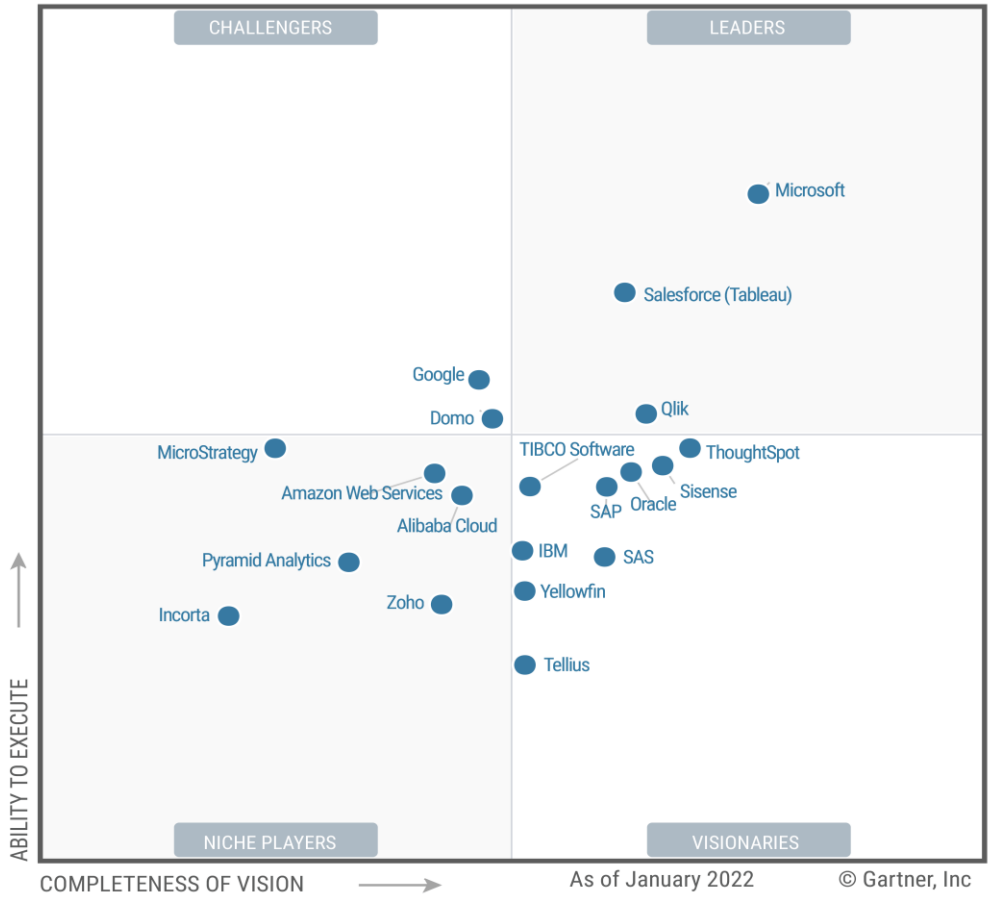
Когда дизайн автомобилей перешел к размещению двигателя спереди, назначением приборной панели стало защищать водителя от жары и масла. Приборная панель также стала удобным местом для размещения датчиков для контроля работы двигателя и других данных об автомобиле, таких как уровень топлива.

Приемы и инструменты создания дэшбордов

- группировка — способ объединения схожих данных (по какому-то общему признаку, например по первой букве слова или имени человека);
- агрегация (сумма, минимум, максимум, количество и т.д.) — способ отображения колонки фактов из исходной базы данных (например уникальное количество посетителей сайта, или сумма расходов на продукты);
- сортировка — упорядочивание уже сгруппированных данных по заданному признаку (кроме алфавита, можно отсортировать фамилии менеджеров по их наибольшим продажам за месяц и т.п.);
- фильтрация — исключение данных по заданному признаку или сложной формуле;
- вычисляемая колонка — способ получения новых данных и знаний с использованием методов работы с датами, строками, математических функций (например отображение имени и фамилии, вычисление возраста согласно дате рождения и текущей дате);
- топовые (лучшие) значения — способ отобразить указанное количество максимальных или минимальных значений данной группировки (например, возраст трёх самых молодых сотрудников крупной компании, или пять менеджеров, обеспечивающих максимальные продажи);
- виджеты (таблицы, диаграммы, карты и т.п.) — собственно способ визуализации вышеуказанных понятий.

Понятия и инструменты дэшбордов

Figure 1: Magic Quadrant for Analytics and Business Intelligence Platforms



Source: Gartner (March 2022)

Аналитические (ad hoc) дэшборды

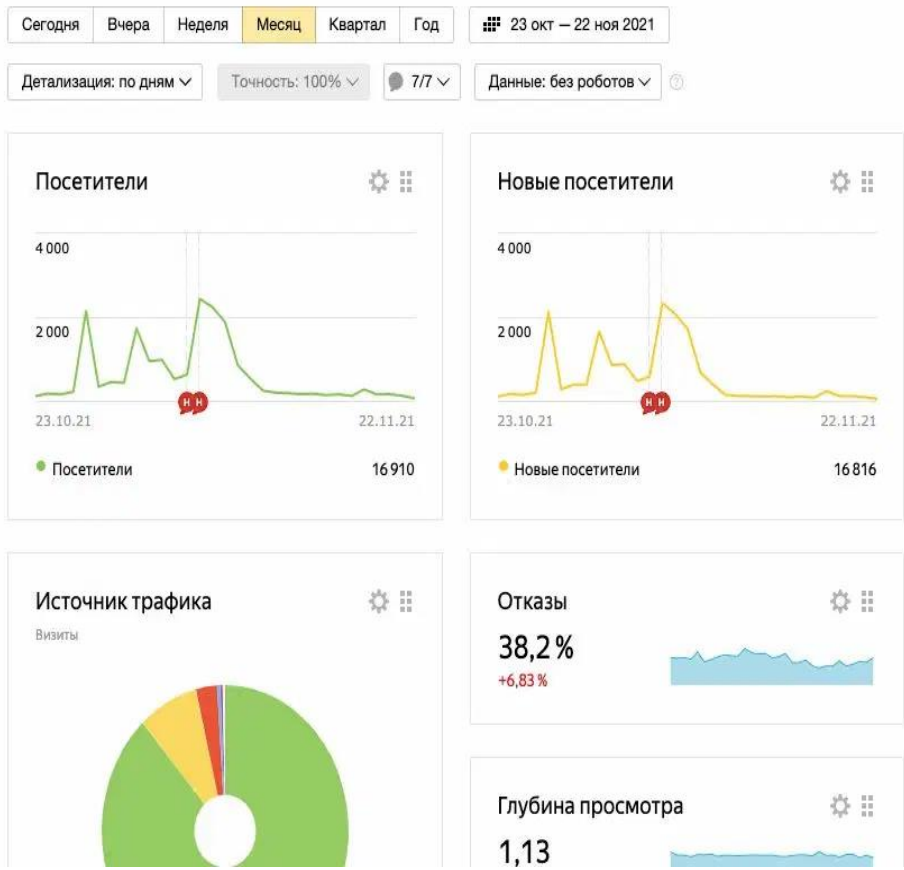
Ad hoc (по запросу) анализ полезен для прояснения собственного мышления или изучения аномалий в данных. Этот тип усилий редко требует какого-либо планирования или проектирования. Аналитики могут решить изучить вопрос несколькими способами, что означает, что на панель инструментов будут представлены несколько (неоптимизированных) визуализаций. Эта информационная панель обычно имеет аудиторию одного человека (вас) или, когда она предоставляется другим, сопровождается описанием или устным объяснением того, как ее интерпретировать.

Пример — разработка дашборда об изменении числа пользователей конкретного продукта за неделю. С его помощью можно увидеть средний чек, долю клиентов и процент товарооборота по каждому из конкурентов.

Операционные дэшборды. Поддержка принятия решений

Самый эффективный вариант использования дэшбордов— поддержка текущего принятия решений. Данные, лежащие в основе этих визуализаций, регулярно обновляются или практически обновляются, и это помогает людям принимать правильные решения несколько раз в день.

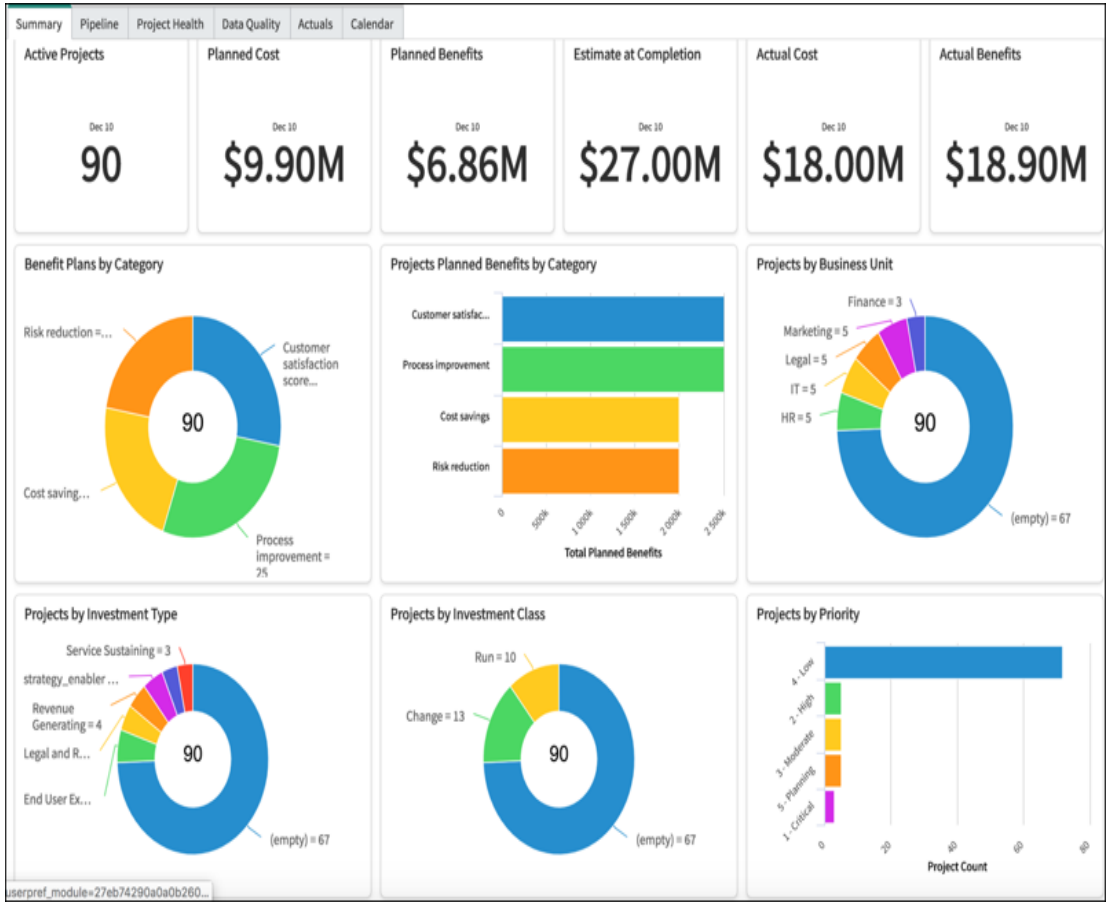
Пример операционного дэшборда для бизнеса — графики Яндекс Метрики, с помощью которых можно посмотреть, как менялась посещаемость сайта и что на неё влияло. Ведь за какой период смотреть график, пользователь выбирает сам.



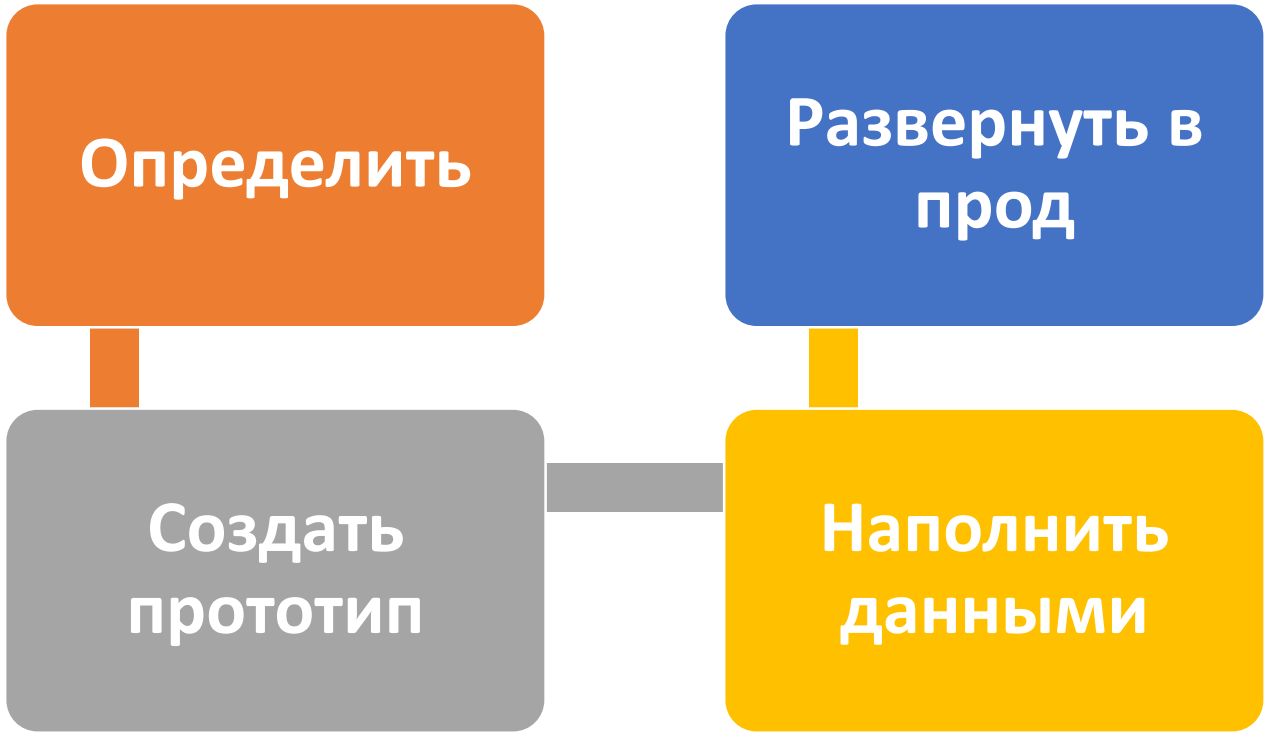
Стратегические дэшборды

Дэшборды стратегического планирования предназначены для крупных проектов или крупных решений, требующих большого количества статических исторических данных для принятия наилучших решений. Они объединяют несколько мнений о том, как работает бизнес, и сравнивают его с конкурентами. Хотя они чрезвычайно полезны, это не самый типичный вариант использования информационных панелей.

Например, создание дашборда о лояльности персонала поможет понять степень лояльности сотрудников и отследить её изменения среди разных групп.



Фреймворки создания дэшбордов. Стандартный процесс



Фреймворки создания дэшбордов. Стандартный процесс. Определить (Define)

Это первый и самый важный шаг. Четкое понимание того, для кого предназначена эта информационная панель и какие показатели для них важны, имеет решающее значение для создания информационной панели, которая будет использоваться.

Фреймворки создания дэшбордов. Стандартный процесс. Определить (Define). Заинтересованные стороны

Есть 4 основных заинтересованные стороны (Stakeholders):

- Дизайнер
- Аудитория (кто будет просматривать эту информационную панель)
- Ведущий (самый опытный член аудитории)
- Data Gatekeeper (член группы данных, который поможет с базой данных)

Фреймворки создания дэшбордов. Стандартный процесс. Определить (Define). Метрики

Вы будете работать с ответственным лицом, чтобы перейти от решений, которые необходимо принять, к метрикам, которые можно запрашивать и отслеживать.

Этот процесс включает в себя множество операций взад и вперед, чтобы отсеять интересные, но в конечном счете не относящиеся к делу метрики, чтобы получить критически важные данные для принятия решений.

Фреймворки создания дэшбордов. Стандартный процесс. Определить (Define). Метрики. Примеры

- количество и сумма продаж в разрезе каналов и источников трафика;
- переходы на сайт из соцсетей;
- поведенческие характеристики: отказы, глубина, время на сайте;
- заявки, звонки и коэффициент конверсии — из визитов в заявки и из лидов в продажи;
- показатели эффективности продвижения и рекламы в Инстаграмме* и во ВКонтакте;
- данные об аудитории — пол, возраст, география;
- расходы на SMM и т. д.

Фреймворки создания дэшбордов. Стандартный процесс. Создать прототипа. Визуализации

Используйте визуализации, которые четко и точно представляют метрики. Даже при создании эскизов и прототипов графиков принятие правильных решений по визуализации улучшит прототип и цикл обратной связи.

Фреймворки создания дэшбордов. Стандартный процесс. Создать прототипа. Дэшборды

Существуют рекомендации по использованию визуализаций и их объединению на дэшборде. На самом деле, некоторые варианты композиции могут фактически заставить вас изменить ту визуализацию, которую вы выбрали в качестве оптимальной ранее.

Фреймворки создания дэшбордов. Стандартный процесс. Создать прототипа. Эскиз и итерация.

На этом этапе рекомендуется набросать визуализации и информационные панели на бумаге или с помощью лоу-фай инструмента, не связанного с какими-либо реальными данными. Причина этого в том, что это позволяет вам быстро отбрасывать плохие идеи, не беспокоясь о временных затратах. Это также позволяет вам сосредоточиться на дизайне, а не на проверке правильности цифр.

Фреймворки создания дэшбордов. Стандартный процесс. Наполнить данными. Определение источников

На этом этапе может возникнуть множество проблем. Где хранятся данные? Данные беспорядочны? У нас вообще есть данные? Работа с группой данных и Data Gatekeeper имеет решающее значение для прохождения этого шага.

Фреймворки создания дэшбордов. Стандартный процесс. Наполнить данными. Рассчитать метрики

Нам нужно создавать запросы для включения метрик, создавать формулы и преобразовывать данные в диаграммы. Использование платформы для регистрации метрик, формул и источников данных значительно упрощает создание запросов.

Фреймворки создания дэшбордов. Стандартный процесс. Развернуть в прод. Совместное использование

Аудитория будет иметь различный уровень грамотности данных и контекст для данных, представленных на панели инструментов. Вам нужно убедиться, что у вас достаточно контекста на информационной панели и что вы проводите достаточное обучение, чтобы люди могли легко извлечь из нее информацию.

Фреймворки создания дэшбордов. Стандартный процесс. Развернуть в прод. Масштабирование

Если информационная панель полезна, количество просмотров и количество зрителей, вероятно, вырастут. Добавление ссылок, интерактивности и документации на информационную панель помогает использовать больше вариантов использования и вдохновляет других создателей информационных панелей.

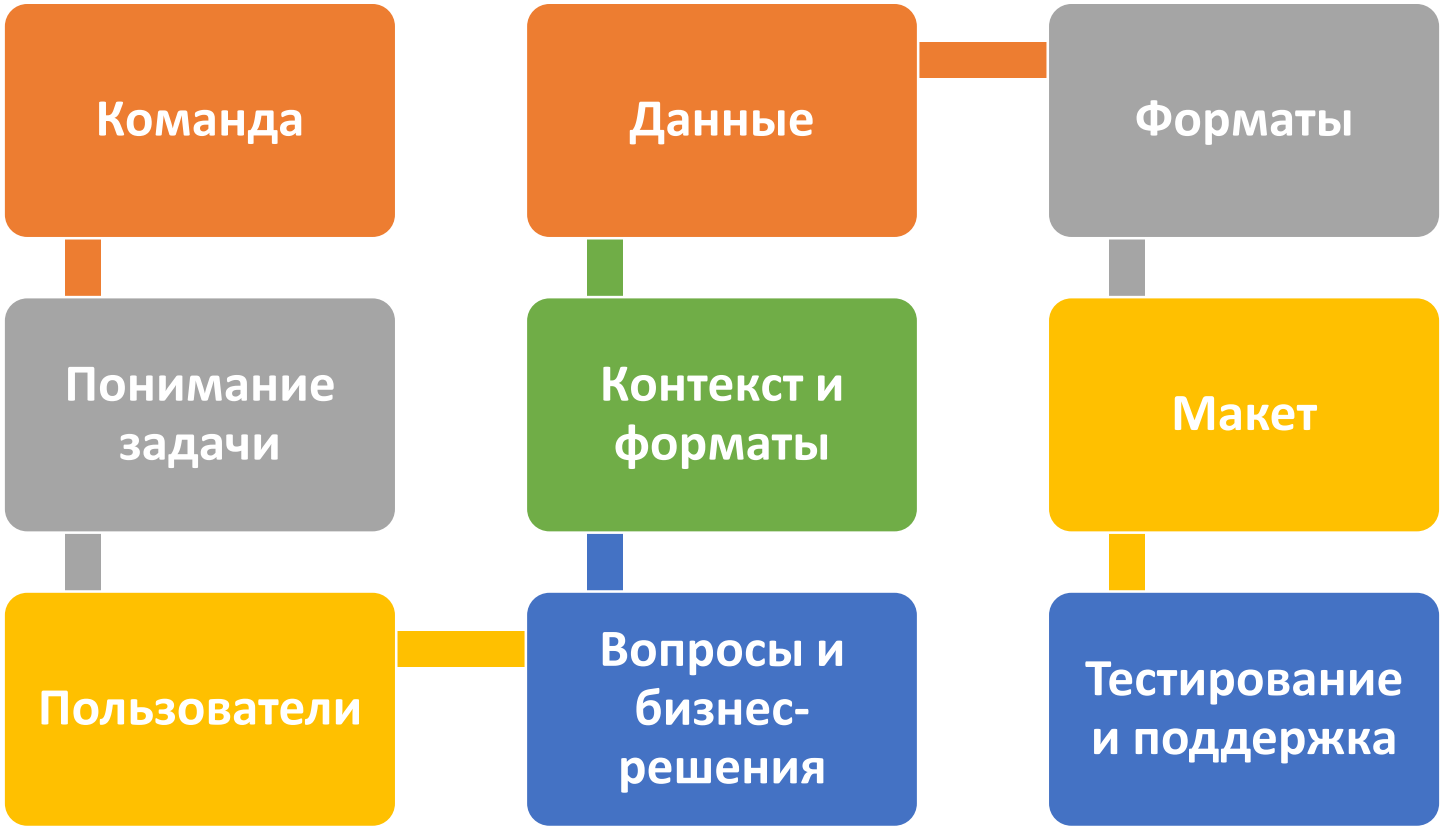
Кроме того, по мере увеличения количества просмотров и зрителей, затраты времени на оптимизацию запросов становятся важным способом поддержания полезности информационной панели.

Фреймворки создания дэшбордов. Стандартный процесс. Развернуть в прод. Обслуживание

Источники данных, таблицы и поля меняются, и информационные панели должны меняться вместе с ними. Настройка запланированного времени для просмотра информационных панелей имеет решающее значение для поддержания их актуальности и функциональности.

Предоставление аудитории возможности предупредить вас о проблемах позволит вам вносить осознанные улучшения в информационную панель.

Фреймворки создания дэшбордов. Dashboard Canvas



Фреймворки создания дэшбордов. Dashboard Canvas. Команда

Сначала мы разбираемся, кто участвует в решении задачи, кто за что отвечает и у кого какие ожидания. Потому что нередко от BI-аналитиков требуют не только визуализировать, но и подготовить данные. И если на проекте нет выделенного BI- или Data-инженера, это может стать проблемой.

Фреймворки создания дэшбордов. Dashboard Canvas. Понимание задачи

На этом этапе необходимо описать, как работает наш бизнес и конкретное подразделение, для которого мы будем создавать дашборд. Цель — синхронизировать видение ситуации и задачи. Удобнее всего делать описание задачи на основе интервью: сначала аналитик общается с менеджером продуктов, а потом формирует понимание задачи.

Пример

Компания оказывает услуги по шести направлениям: консалтинг, интеграция, разработка интернет-магазинов, мобильных приложений и т. д.

Работа ведется с B2B клиентами.

Процесс продажи состоит из следующих этапов:

Получение лида → Встреча (очная или онлайн) → Выставление КП → Согласование, дожим клиента → Подписание договора, выставление счета → Получение аванса, старт работ по проекту.

Фреймворки создания дэшбордов. Dashboard Canvas. Пользователи

Теперь необходимо выяснить, кто будет пользоваться дашбордом, какие роли пользователей в нем будут.

Пример

Руководитель отдела продаж

Смотрит дашборд на ежедневной основе, хочет видеть общую картинку, иметь возможность анализировать успешность каналов и разных типов продукта. Понимать на какие сделки и менеджеров нужно обратить внимание. Если внезапно позвонит владелец компании, должен сориентировать по плану продаж и когда возможно будут закрыты новые сделки.

Менеджеры по продажам

Смотрят дашборд каждый день, хотят видеть на какие сделки стоит обратить внимание в первую очередь и их статус. Анализируют свой портфель и планируют, что нужно исправить, чтобы закрыть план.

Фреймворки создания дэшбордов. Dashboard Canvas. Вопросы и бизнес-решения

Здесь необходимо разобраться, какие бизнес-решения пользователи будут принимать на основе визуализированных данных и на какие вопросы искать ответы. И это, пожалуй, самая сложная задача в Dashboard Canvas.

Пример

Какова общая обстановка: сколько сделок в работе, какие средние сроки закрытия сделки и конверсия, как выполняется план.

Какова динамика количества сделок

Какова динамика конверсии

Каков портрет клиента (размер сделок, новый или постоянный клиент)

Фреймворки создания дэшбордов. Dashboard Canvas. Контекст и форматы

В этом блоке необходимо описать, когда и в каких условиях пользователи смотрят дашборд, какими устройствами пользуются.

Пример

Дашборд смотрят ежедневно по утрам при планировании рабочего дня или при проведении стенд-апа на мониторах компьютера (менеджеры работают удалённо).

Руководителю отдела продаж иногда нужно заходить в отчет с мобильного, чтобы понять общую ситуацию и сообщить её руководителю фирмы.

Фреймворки создания дэшбордов. Dashboard Canvas. Данные

После этого мы определяем структуры данных, выясняем, какие данные уже есть, а какие предстоит собрать.

Пример

Менеджеры ведут сделки в CRM и ежедневно заполняют данные. План утверждается раз в квартал в разрезе менеджер-продукт и хранится в эксель.

Интеграция с CRM:

schema.table_crm id | Название сделки | Компания | Продукт | Менеджер | Этап сделки | Сумма | Дата создания | Количество дней в работе | Источник

План утверждается раз в квартал, файл в эксель состоит из следующих колонок:

Продукт | Менеджер | План

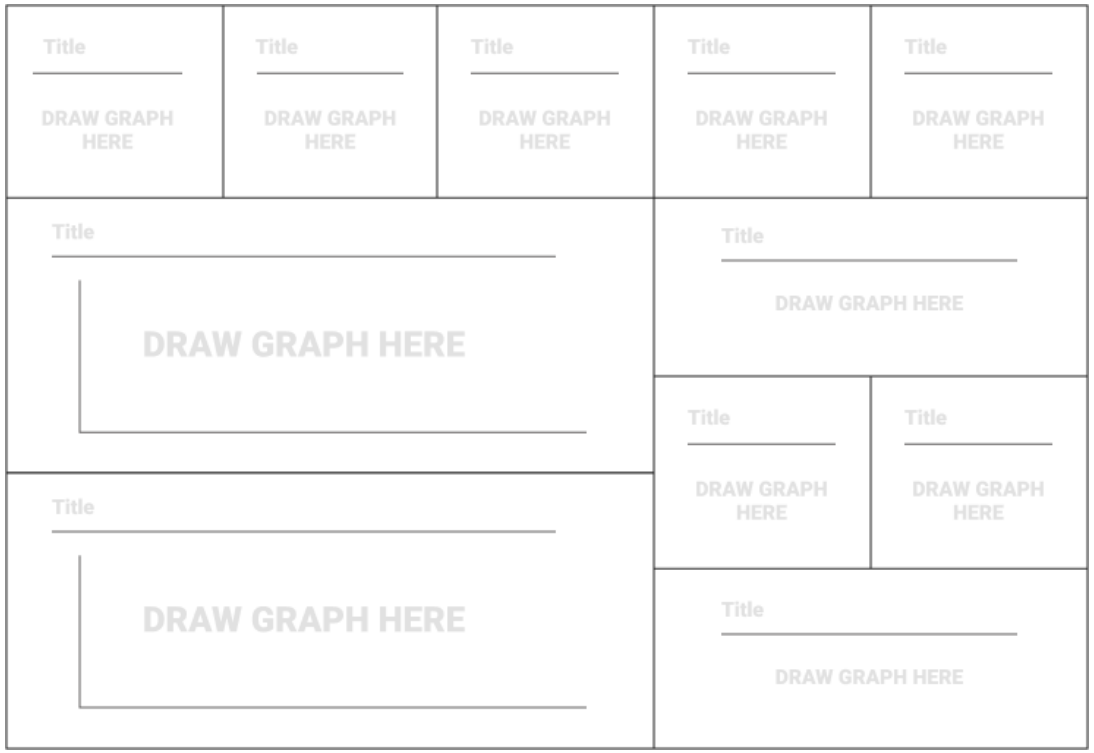
Фреймворки создания дэшбордов. Dashboard Canvas. Форматы

Когда у нас есть структура данных и контекст использования дашбордов, можно заполнять седьмой блок и продумать форматы: будет ли мобильная версия, какие графики и показатели будут располагаться выше и правее, а какие займут менее приоритетные позиции.

На каждый вопрос из блока 5 необходимо предложить как минимум один график для ответа.

Фреймворки создания дэшбордов. Dashboard Canvas. Макеты

На этом этапе мы верхнеуровнево отрисовываем макет будущего дэшборда, располагаем блоки и графики относительно друг друга, а потом заверстываем



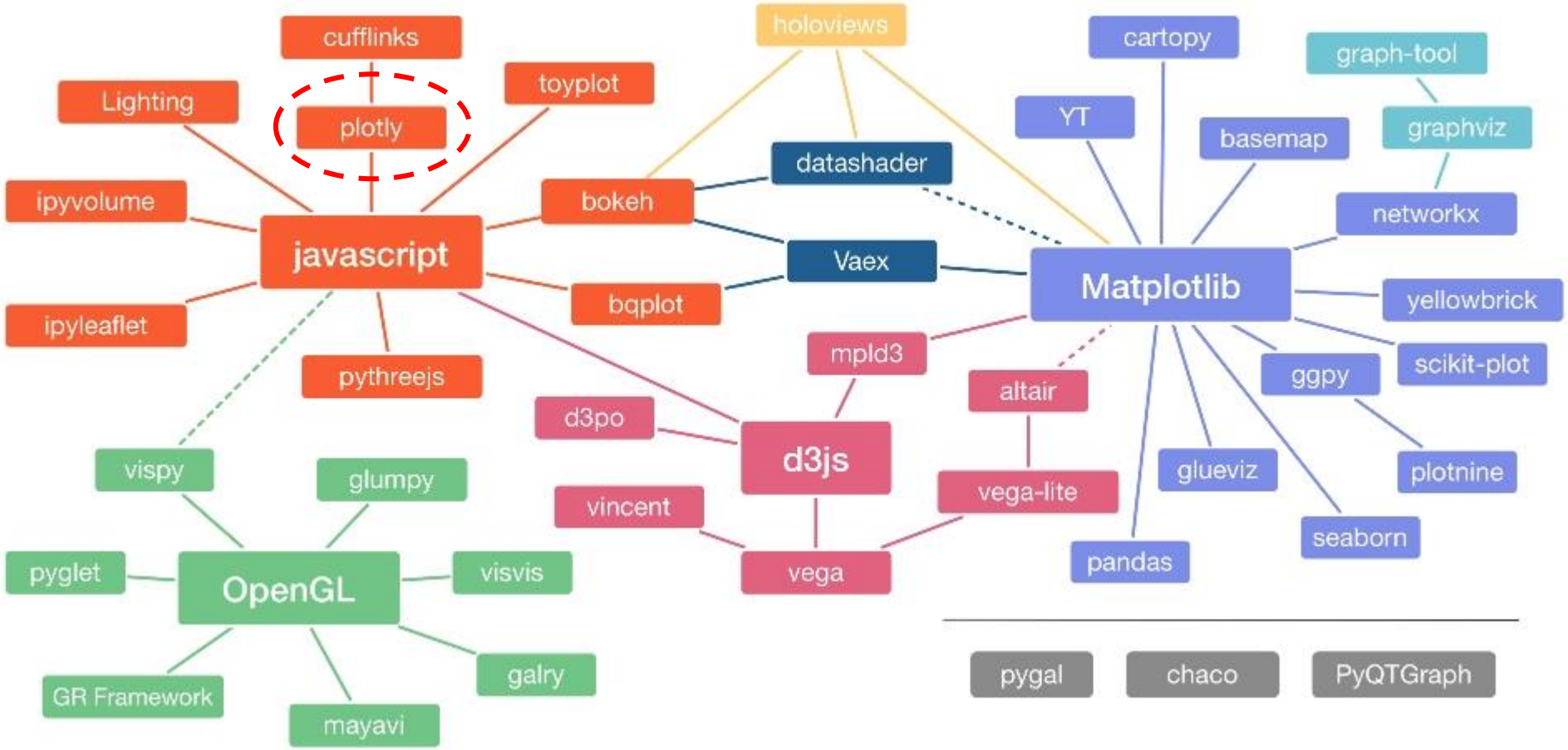
Фреймворки создания дэшбордов. Dashboard Canvas. Тестирование и поддержка

- Демо-сессия и форма сбора обратной связи через чат и опросы.
- Тестирование в реальных условиях, наблюдение за использованием дашборда.
  - Статистика по использованию дашборда.
  - Success Metric — все менеджеры каждый день используют отчет.
  - Средний балл удовлетворенности дашбордом 4,5 из 5.

Фреймворки создания дэшбордов. Dashboard Canvas. Пример опросника

- Расскажи, как работает ваше подразделение
- Как вы понимаете, хорошо или плохо вы работаете? Какие основные KPI у вашего подразделения?
- От чего зависит изменения этих метрик? Что вы делаете чтобы их менять?
- Какой дашборд вы хотите? Зачем он вам нужен? Как решаете задачу сейчас?
- Кто будет им пользоваться? Кого можно включить в тестовую группу?
- Как будете пользоваться дашбордом? В каких случаях?
- Какие основные сценарии и контексты при использовании?
- Что делаете после того как посмотрели дашборд? Какие и как решения применяете?
- Когда хотели бы получить результат? Почему именно в это время?
- Какие данные можно посмотреть?
- Что я забыл спросить?

Plotly. Dash. Место фреймворка



Plotly. Dash

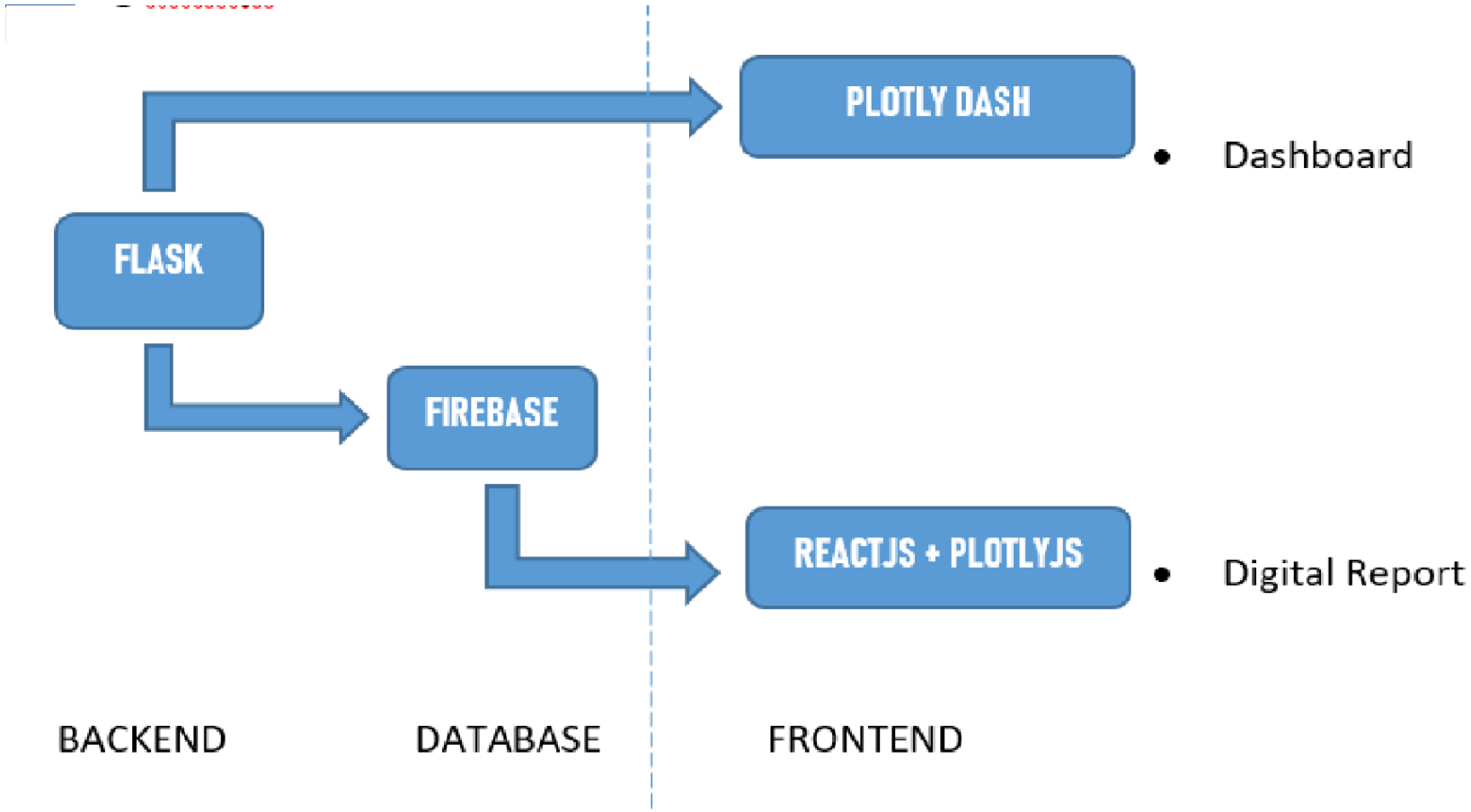
Plotly Dash — это фреймворк с открытым исходным кодом для создания интерфейсов визуализации данных. Выпущенная в 2017 году как библиотека Python, она расширилась и теперь включает реализации для R, Julia и F#. Dash помогает специалистам по обработке и анализу данных создавать аналитические веб-приложения, не требуя передовых знаний в области веб-разработки.

Plotly. Dash. Технологии

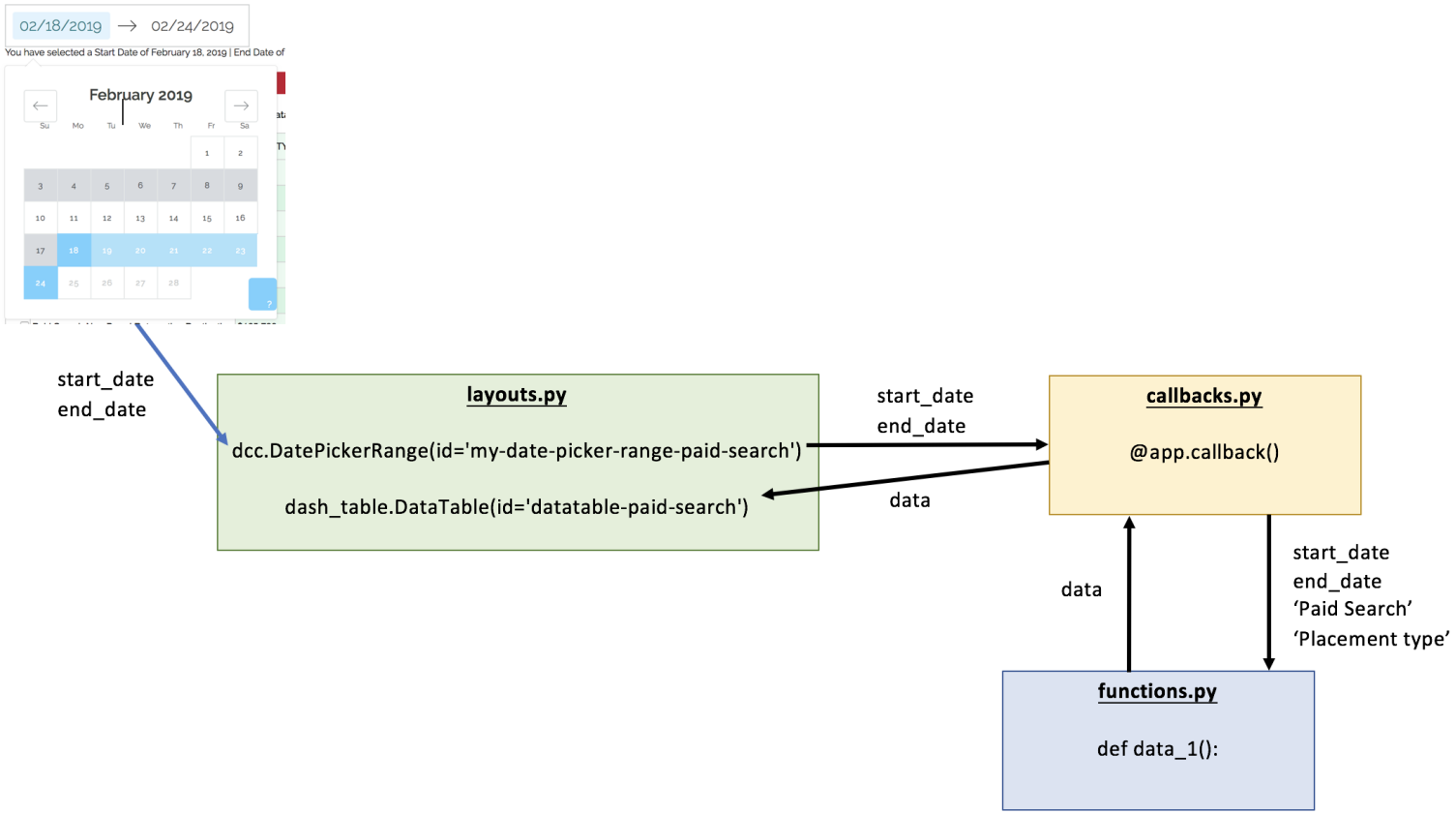
Работа Plotly Dash обеспечивается применением следующих технологий:

- Flask обеспечивает функциональность веб-сервера
- React.js отображает пользовательский интерфейс веб-страницы
- Plotly.js генерирует диаграммы, используемые в целевом приложении

Plotly. Dash. Структура приложения



Plotly. Dash. Структура объектов



Модули Dash. DCC, HTML

В dash_core_components содержатся различные динамические формы такие как, например, выпадающие списки, графики и чек-боксы.

В dash_html_components содержатся html конструкции, которыми можно обернуть формы. Например, Div блоки или теги заголовков H1, H2, и т.п.

Dash. DCC. Слайдер

Dash is running on http://127.0.0.1:8050/

INFO:dash.dash:Dash is running on http://127.0.0.1:8050/

Dash. DCC. Выпадающий список

```
Dash is running on http://127.0.0.1:8050/

INFO:dash.dash:Dash is running on http://127.0.0.1:8050/
```

Plotly. Dash. Макет. Особенности

- Макет состоит из дерева «компонентов», таких как html.Div и dcc.Graph.
- Модуль HTML-компонентов Dash (dash.html) имеет компонент для каждого тега HTML. Компонент html.H1(children='Hello Dash') создает HTML-элемент <h1>Hello Dash</h1> в вашем приложении.
- Не все компоненты представляют собой чистый HTML. Модуль Dash Core Components (dash.dcc) содержит компоненты более высокого уровня, которые являются интерактивными и генерируются с помощью JavaScript, HTML и CSS через библиотеку React.js.
- Каждый компонент полностью описывается с помощью ключевых слов. Dash является декларативным: вы в первую очередь будете описывать свое приложение с помощью этих атрибутов.

Модули Dash. Inputs, Outputs

В Dash входы (Inputs) и выходы (Outputs) приложения — это свойства конкретного компонента. Всякий раз, когда изменяется входное свойство, функция, которую обертывает декоратор обратного вызова, будет вызываться автоматически.

Dash предоставляет этой функции обратного вызова новое значение входного свойства в качестве аргумента, а Dash обновляет свойство выходного компонента тем, что было возвращено функцией.

Plotly. Dash. Обратные вызовы (callbacks)

Главной особенностью Dash являются обратные вызовы (callbacks). Данные могут быть самыми разнообразными: показатели могут меняются в зависимости от атрибутов или данные могут обновляться с каждым днем и т.д. С таким видом данных в Dash можно работать, используя лишь специальный декоратор @app.callback, где app — название переменной приложения.

JupyterDash

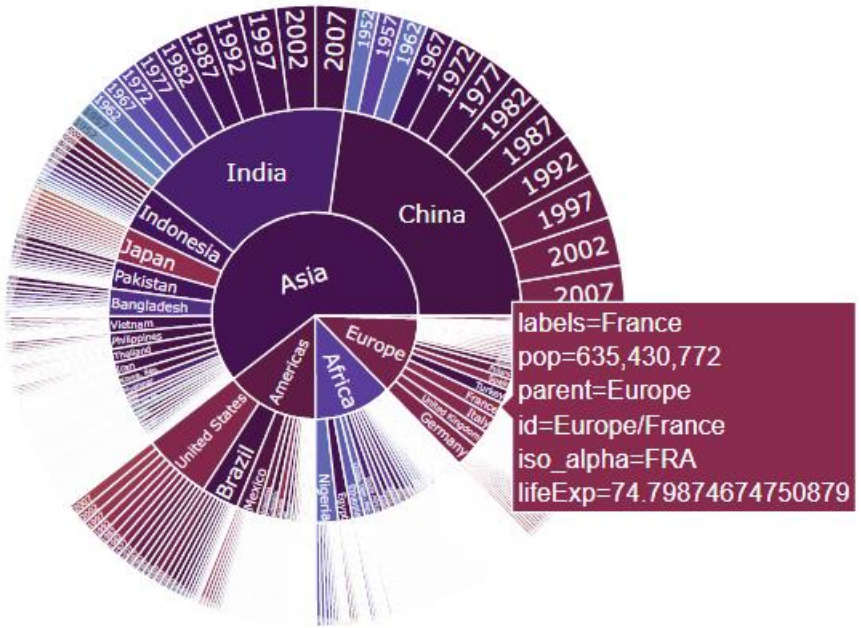
JupyterDash разработан на основе платформы Dash, что делает его полностью подходящим для сред ноутбуков, таких как Colab. Библиотека JupyterDash с открытым исходным кодом делает графики в Colab интерактивными в реальном времени с помощью наведения курсора, маркеров и других удобных элементов управления.

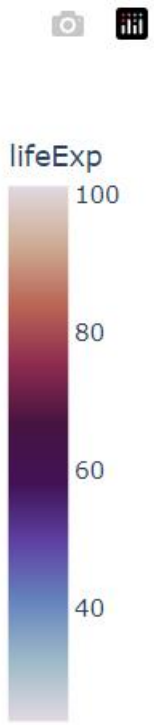
Изменения в данных или коде немедленно влияют на визуализацию, что делает Plotly удобным решением для потоковой передачи данных. Кроме того, визуализация Colab отображается на отдельной веб-странице с горячей перезагрузкой и взаимодействием ввода/вывода.

Plotly. Dash. Построение дэшборда с фильтрами. Пример солнечной вспышки

```
import plotly.graph_objects as go
import plotly.express as px
from jupyter_dash import JupyterDash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import numpy as np

df = px.data.gapminder()

fig = px.sunburst(df, path=['continent', 'country', 'year'], values='pop',
                  color='lifeExp', hover_data=['iso_alpha'],
                  color_continuous_scale='twilight',
                  color_continuous_midpoint=np.average(df['lifeExp'], weights=df['pop']))

app = JupyterDash(__name__)

app.layout = html.Div([dcc.Graph(figure=fig)])

app.run_server(mode='inline')
```

Plotly. Dash. Построение дэшборда с фильтрами. Пример солнечной вспышки. Продолжение

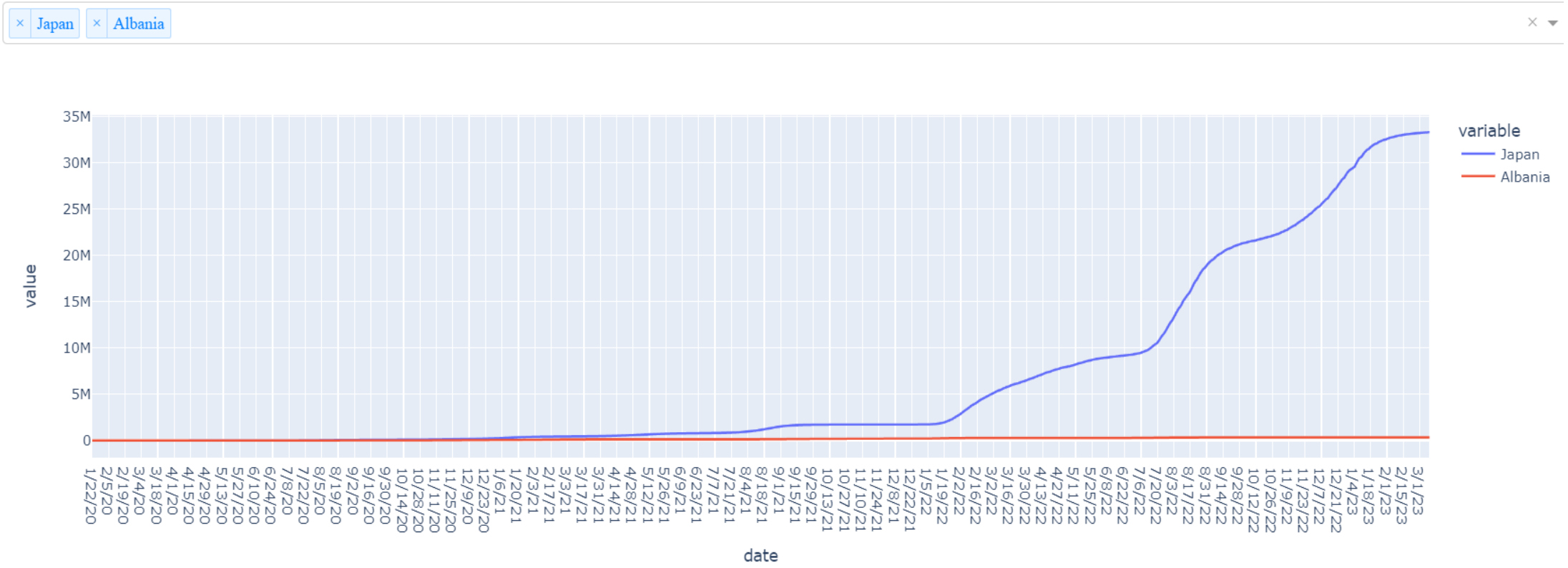




Plotly. Dash. Построение дэшборда с фильтрами. Пример линейного графика

```
app = JupyterDash(__name__)

app.layout = html.Div([
  dcc.Dropdown(id="my_dropdown",
    options=[{"value": country, "label": country} for country in df.columns.unique()],
    value=["Japan"],
    multi=True
    ),
  dcc.Graph(id="my_graph")
])

@app.callback(Output("my_graph", "figure"), Input("my_dropdown", "value"))
def update_graph(selected_country):
  return px.line(df, x="date", y=selected_country)

app.run_server(mode="inline")
```

Plotly. Dash. Построение дэшборда с фильтрами. Пример линейного графика. Продолжение



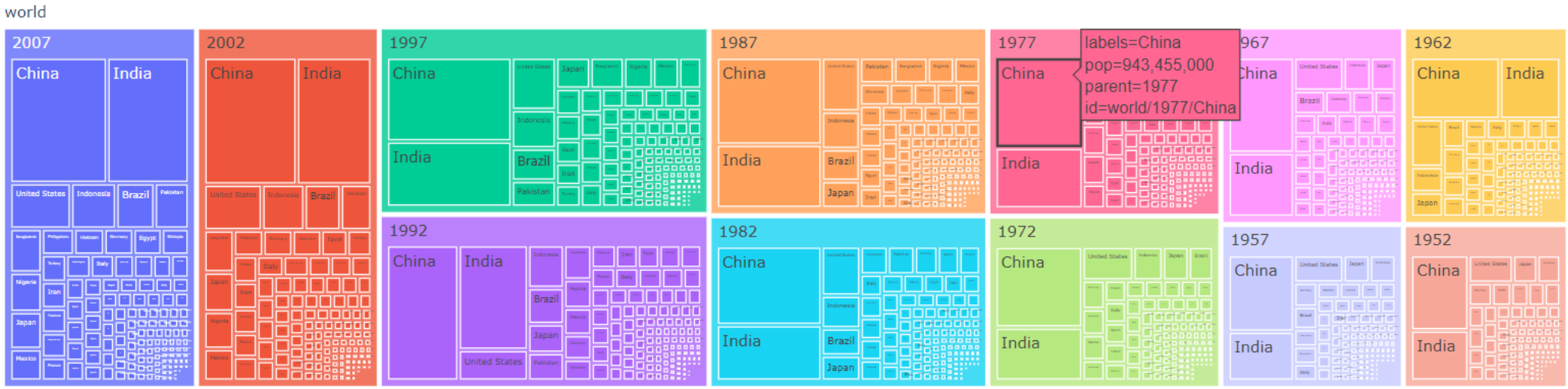
Plotly. Dash. Построение дэшборда с фильтрами. Пример пузырьковой диаграммы

```
import plotly.graph_objects as go
import plotly.express as px
from jupyter_dash import JupyterDash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import numpy as np

df = px.data.gapminder()
data = df.query('year==2002')
fig = px.scatter(data, x="gdpPercap", y="lifeExp", size="pop", color='continent', hover_name="country", size_max=60)

app = JupyterDash(__name__)

app.layout = html.Div([dcc.Graph(figure=fig)])

app.run_server(mode='inline')
```

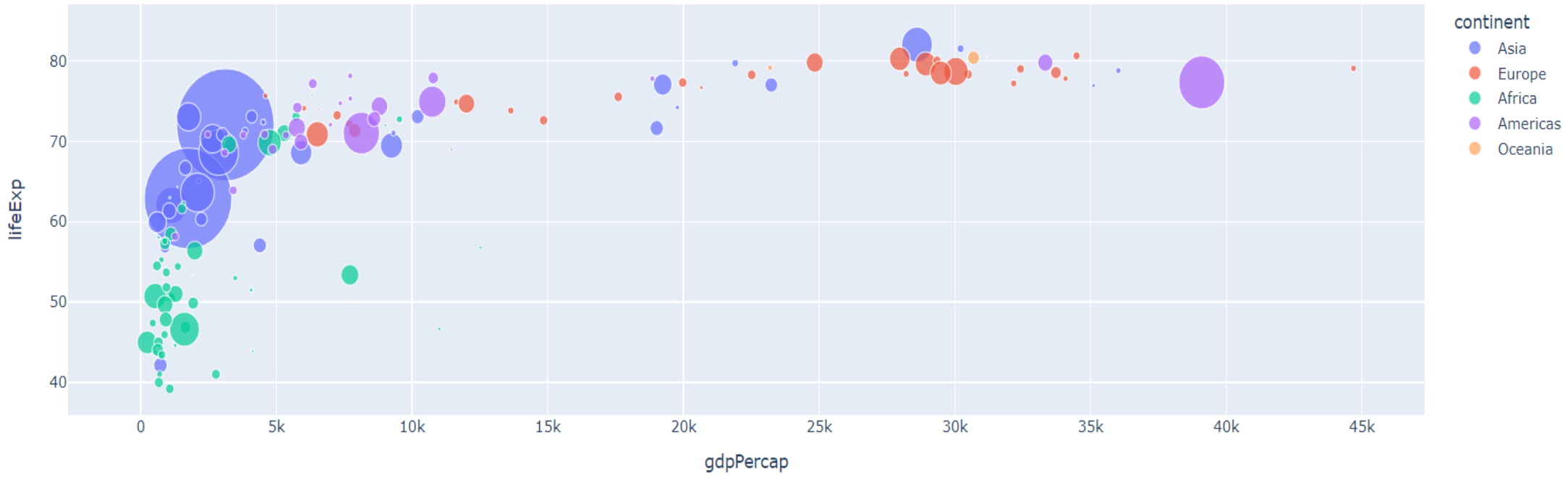
Plotly. Dash. Построение дэшборда с фильтрами. Пример древесной диаграммы.

```
import dash
from jupyter_dash import JupyterDash
import dash_core_components as dcc
import dash_html_components as html
import plotly.express as px
from dash.dependencies import Input, Output

gapminder = px.data.gapminder()
gapminder.head()

gapminder['board'] = 'world'
fig=px.treemap(gapminder, path=['board', 'year', 'country'], values='pop')

app = JupyterDash(__name__)

app.layout = html.Div([dcc.Graph(figure=fig)])

app.run_server(mode='inline')
```

Plotly. Dash. Построение дэшборда с фильтрами. Пример древесной диаграммы. Продолжение



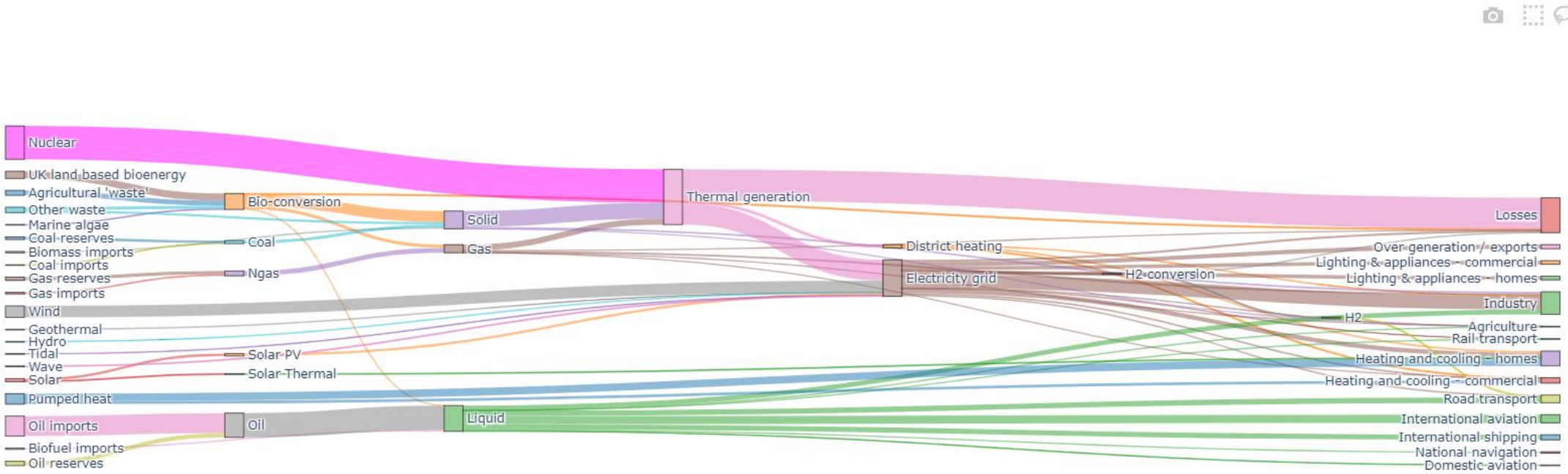
Plotly. Dash. Построение дэшборда с фильтрами. Пример пузырьковой диаграммы. Продолжение



Plotly. Dash. Построение дэшборда с фильтрами. Пример диаграммы Санки. Callbacks

```
import json
import urllib
url = 'https://raw.githubusercontent.com/plotly/plotly.js/master/test/image/mocks/sankey_energy.json'
response = urllib.request.urlopen(url)
data = json.loads(response.read())
app = JupyterDash(__name__)
app.layout = html.Div([
    dcc.Graph(id="graph"),
    html.P("Opacity"),
    dcc.Slider(id='opacity', min=0, max=1,
               value=0.5, step=0.1)
])
@app.callback(
    Output("graph", "figure"),
    [Input("opacity", "value")])
def display_sankey(opacity):
    opacity = str(opacity)
    node = data['data'][0]['node']
    link = data['data'][0]['link']
    node['color'] = [
        'rgba(255,0,255,{})'.format(opacity)
        if c == "magenta" else c.replace('0.8', opacity)
        for c in node['color']]
    link['color'] = [
        node['color'][src] for src in link['source']]
    fig = go.Figure(go.Sankey(link=link, node=node))
    fig.update_layout(font_size=10)
    return fig
app.run_server( mode='inline')
```

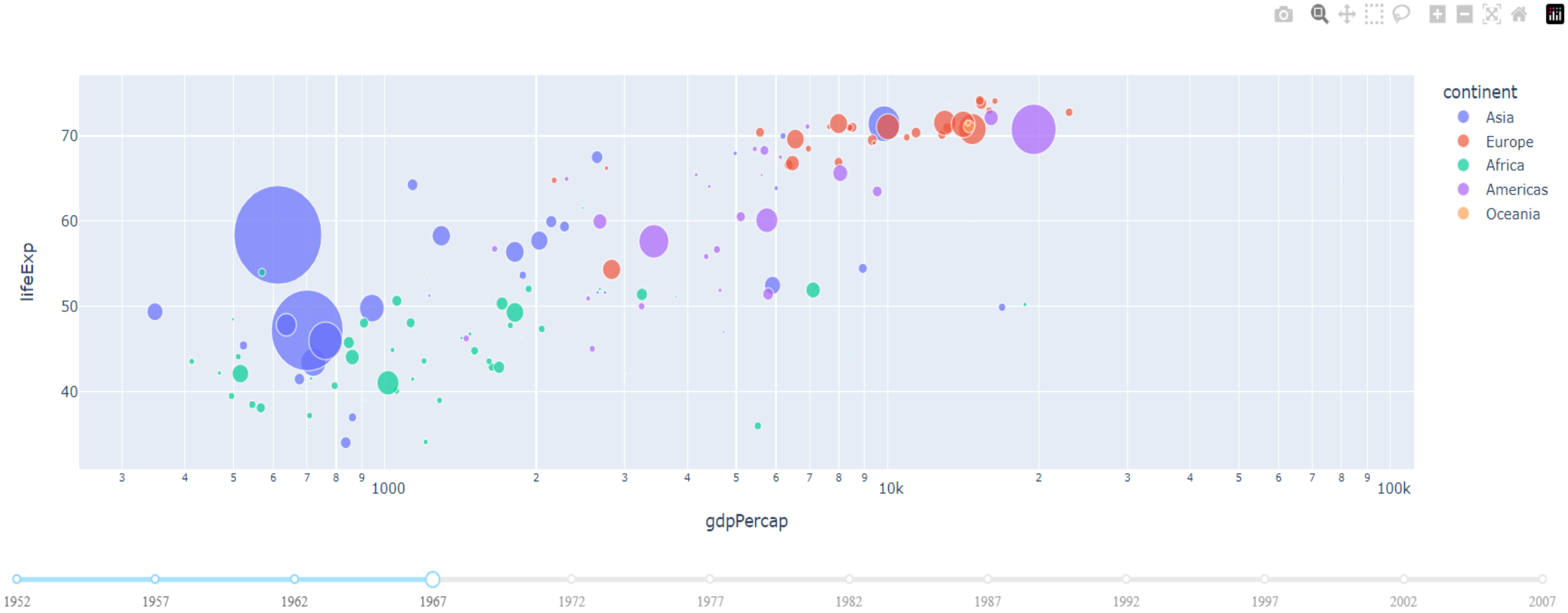
Plotly. Dash. Построение дэшборда с фильтрами. Пример диаграммы Санки. Callbacks. Продолжение



Plotly. Dash. Построение дэшборда с фильтрами. Пример пузырьковой диаграммы. Callbacks

```
from dash.dependencies import Input, Output
import plotly.express as px

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/gapminderDataFiveYear.csv')

app = JupyterDash('app')
app.layout = html.Div([
    dcc.Graph(id='my-graph'),
    dcc.Slider(
        id='year-slider',
        min=df['year'].min(),
        max=df['year'].max(),
        value=df['year'].min(),
        marks={str(year): str(year) for year in df['year'].unique()},
        step=None
    )
])

@app.callback(
    Output('my-graph', 'figure'),
    [Input('year-slider', 'value')])
def update_figure(selected_year):
    filtered_df = df[df.year == selected_year]

    fig = px.scatter(filtered_df, x="gdpPercap", y="lifeExp",
                     size="pop", color="continent", hover_name="country",
                     log_x=True, size_max=55)
    return fig
app.run_server(mode='inline')
```

Plotly. Dash. Построение дэшборда с фильтрами. Пример диаграммы Санки. Продолжение



Спасибо за внимание!