

# Apache Superset für Fortgeschrittene

Ein Informationsblatt für Interessierte.

## Übersicht

Dieses Informationsblatt bezieht sich auf Release 2.0.1 von Apache Superset. Es enthält folgende Themen:

- Dokumentation und Hilfe
- Upload von Daten (Tabellen im CSV-Format)
- Einen Chart erstellen
- Ein Dashboard erstellen, bearbeiten und publizieren
- Eine Tabelle mit einer anderen verknüpfen und die Datenbanksprache SQL nutzen
- Datentypen, Datentyp-Rollen und Datentyp-Encodings
- Der Visualisierungsprozess und die Superset Map Charts
- Charts und Dashboards anpassen, u.a. mit Tooltips und Annotations

Im Anhang sind alle Map Charts mit allen notwendigen Parametern nochmals zusammengestellt.

Man beachte auch das Arbeitsblatt "Einführung in Apache Superset".



Apache Superset ist auf Daten-Visualisierungen ausgerichtet, die einen erheblichen Bedarf an Ressourcen auf Server- wie auch auf Client-Seite erfordert. Superset ist darum vor allem auf Web-Browser auf dem Desktop oder auf Tablets ausgerichtet und kann auf mobilen Geräten mit kleinen Bildschirmen nur beschränkt genutzt werden.

## Dokumentation und Hilfe

Eine [offizielle, englischsprachige Dokumentation](#) zu Apache-Superset gibt es auf der Website zu finden. Ebenfalls auf der selben Seite zu finden, ist ein [FAQ](#) in welchem einige wichtige Fragen geklärt werden.

Wer Fragen hat, dem empfehlen wir, diese auf [Stackoverflow](#) zu stellen (immer zusammen mit dem Tag 'apache-superset').

# Upload von Daten (Tabellen im CSV-Format)

Dieses Kapitel beschreibt, wie man Daten vom eigenen Computer in Apache Superset importiert. Superset erlaubt das Hochladen von Dateien im Format CSV (Comma Separated Values).

Als Beispiel nehmen wir für den Upload die Daten des "CIA World Factbook". Die entsprechenden Daten heissen "Apache Superset-Daten (CSV)" auf [OpenSchoolMaps](#) und können dort unter **Unterrichtsmaterialien > Apache Superset das Datenvisualisierungs- und Publikations-Werkzeug > Daten** heruntergeladen und ausgepackt werden.

Die CSV-Dateien `cia_world_factbook_2019.csv` und `country_codes.csv` sollten nun auf deinem Computer vorhanden sein. Überprüfe zuerst die Kodierung und das Format der Datei: Die Codierung muss UTF-8 sein und das Format CSV.



MS Excel verwaltet normalerweise Dateien im .xlsx-Format. Diese lassen sich als CSV UTF-8 exportieren. Der von MS Excel bevorzugte Delimiter ist das Semikolon.

CSV-Datei hochladen:

- Im Menü **+ > Data > Upload CSV to Dataset**. Es erscheint das Formular "CSV upload".



Wenn diese Option ausgegraut ist, muss diese Option zunächst aktiviert werden. Um die Option zu aktivieren, muss zunächst zu **Data > Databases** navigiert werden, wo man anschliessend die Datenbank, welche verwendet werden soll, auswählen kann. Mit einem Klick auf den Editierstift kann im Tab **Advanced > Security** die Checkbox **Allow Data upload** aktiviert werden.

- Im Formular "CSV upload" den Tabellennamen auswählen, z.B. `factbook_data`. (Der Einfachheit halber wird empfohlen, keine Grossbuchstaben in Tabellennamen zu verwenden.)
- Die CSV-Datei `cia_world_factbook_2019.csv` auswählen
- Die gewünschten "Database" und "Schema" wählen, hier `examples` und `main` (Datenbank muss durch den Superset-Administrator vorbereitet sein.)
- Falls nötig, den Delimiter anpassen. (Für die Aufgabe ;)
- Unter **File Settings > If Table Already Exists** auswählen, was Superset tun soll, wenn schon eine Tabelle mit demselben Namen existiert.
- Auf "Upload" klicken.
- Diesen Upload falls nötig wiederholen, beispielsweise mit der CSV-Datei `country_codes.csv`



**Rows > Header row** definiert, welche Row aus dem CSV die Header definiert.



Die Zuordnung und Konversion von Datentypen benötigt Kenntnisse der zugrundeliegenden Daten und know-how. Falls man betreffend den möglichen Datentypen unsicher ist, kann man unten im Kapitel [Datentypen](#) nachschauen.

Damit ist das Hochladen der Dateien abgeschlossen und man möchte typischerweise die Daten publizieren. Dies geschieht mit einem Chart und einem Dashboard.

Das nächste Kapitel erläutert, wie man einen Chart erstellt.

**CSV upload** [X]

**General information**  
Upload a file to a database. [^]

CSV file \*  
[Select]

Preview uploaded file

Columns:  
Upload file to preview columns

Database \*  
examples [v]

Schema  
main [v]

Table name \*  
factbook\_data

Delimiter ⓘ  
Semicolon ";" [v]

**File settings**  
Adjust how spaces, blank lines, null values are handled and other file wide settings. [v]

**Columns**  
Adjust column settings such as specifying the columns to read, how duplicates are handled, column data types, and more. [v]

[Cancel] [Upload]

Abbildung 1. Formular "CSV to Database configuration".

# Einen Chart erstellen

Ein neuer Chart wird wie folgt aus einer Tabelle erstellt:

- Im Menübalken oben **Datasets** auswählen.
- Die gewünschte Tabelle suchen (z.B. die zuvor hochgeladene Tabelle **factbook\_data**) und auf den Namen klicken.
- Es erscheint ein Chart-Fenster mit "Visualization Type" (Chart) "Table" (dem Default Chart-Typ).
- Es kann überprüft werden, ob alle Zeilen übernommen wurden, indem die Anzeige auf **[ Samples ]** geklickt wird. Anschliessend sollte oben rechts im Fenster **261 rows** erscheinen.
- **Dimensions** kann auf **country\_name** gestellt werden.
- Unter **Metrics > Simple** kann die Zeile **population** zusammen mit dem "aggregate" **SUM** eingetragen werden.
- Mithilfe von **Filters > Simple** kann die **population** mit "Is not null" herausgefiltert werden.
- Anschliessend soll dem Chart ein Name gegeben werden (unter: "Add the name of the chart"), in diesem Beispiel **CIA World Factbook Table**.
- Rechts oben auf **[ Save ]** klicken. Es erscheint ein Dialog "Save Chart".
- Wenn dem Chart noch keinen Namen gegeben wurde muss nun, unter **Chart name > Name**, einen Namen zugeteilt werden.
- Auf **[ Save ]** klicken.

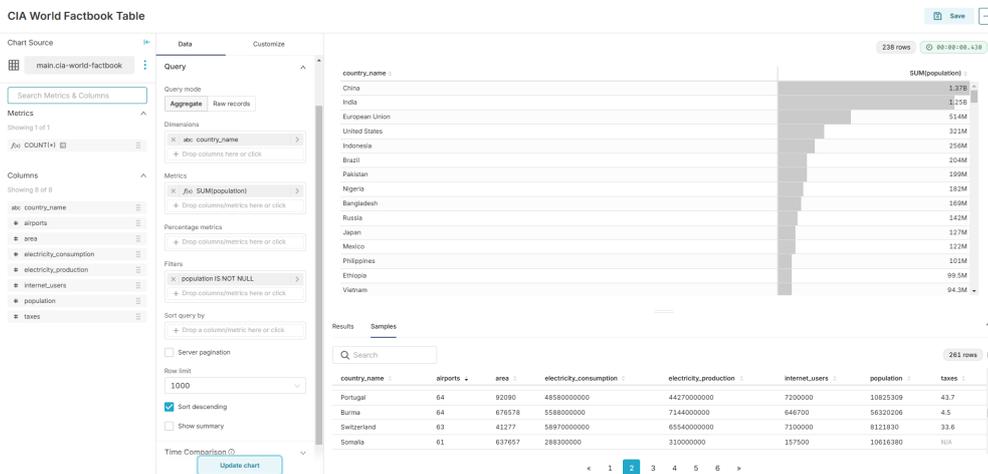


Abbildung 2. Einen Chart erstellen: hier ein Chart vom Typ "Table" mit der Tabelle **factbook\_data** als Quelle.

# Ein Dashboard erstellen, bearbeiten und publizieren

Dieses Kapitel zeigt, wie man ein Dashboard aus einem Chart erstellen, bearbeiten und publizieren kann.

## Ein Dashboard erstellen

Hat man einen Chart erstellt — beispielsweise den Chart "CIA World Factbook Table" im Kapitel "Upload von Daten" (siehe auch das Arbeitsblatt "Einführung in Apache Superset") — so will man typischerweise den Chart zu einem Dashboard hinzufügen und publizieren.

Neues Dashboard erstellen:

- Ein neues Dashboard kann entweder im Menü **Dashboards** über die Schaltfläche **" + Dashboard "** rechts erstellt werden oder aber direkt rechts oben mit der Schaltfläche **+ > Dashboard**. Es öffnet sich das Formular "Add Dashboard".
- Im Formular "Add Dashboard" einen Titel angeben, z.B. "CIA World Factbook Test".
- Mit Klick auf die Schaltfläche **[ Save ]** das Dashboard speichern.

Damit ist das Dashboard zwar erstellt, doch wir müssen es noch layouten und Charts zuordnen.

## Ein Dashboard bearbeiten

Dieses Kapitel zeigt, wie man ein neu erstelltes Dashboard bearbeitet (Layout, d.h. Positionierung der Charts).

Dashboard bearbeiten:

- Auf Menü **Dashboards** klicken. Eine Liste mit Dashboards erscheint.
- In der Liste auf das gewünschte Dashboard klicken, z.B. "CIA World Factbook Test". Das Dashboard öffnet sich.
- Im Dashboard-Fenster rechts oben auf die Schaltfläche **[ Edit Dashboard ]** klicken. Das Dashboard ist nun im "Edit Mode" und es erscheint unter anderem der Dialog "Insert components" rechts.
- Das Dashboard kann nun mit Charts und Layout Elementen ge-layoutet werden. Änderungen werden direkt gespeichert.
- Mein einem Klick auf **[ SAVE ]** kann zurück in die User-Ansicht gewechselt werden.



Das Layout, d.h. die Positionierung der Charts erfordert einige Übung. In den Kapiteln weiter unten gibt es einige Hinweise, wie man Charts und Dashboards speziell anpasst.

# Ein Dashboard publizieren

Dieses Kapitel zeigt, wie man ein Dashboard publiziert und mit anderen teilt.

Dashboard publizieren:

- Ins Menü **Dashboards** wechseln. Die Dashboards-Liste erscheint.
- Bei dem gewünschten Dashboard kann auf das kleine Edit-Symbol geklickt werden. Dann können in dem neuen Menu die berechtigten Rollen eingestellt werden.
- Im Feld **Access** kann die Rolle eingetragen werden, welche Zugriff erhalten soll.
- Das Dashboard, welches publiziert werden soll, auswählen und unter dem Menü **share > Copy permalink to clipboard** den Link kopieren. Über diesen kann das Dashboard nun aufgerufen werden.

Nun ist dein Dashboard publiziert und kann anderen Benutzer\*innen zur Verfügung gestellt werden!



Falls der/die Benutzer\*in zwar das publizierte Dashboard sehen und aufmachen kann, aber keine Daten in den Charts sieht, sollte man die Benutzer-Rolle kontrollieren. Sie muss so konfiguriert sein, dass sie die zugrundeliegende Datenbank lesen darf (permission "datasource access on ..." oder "all\_datasource\_access"). Dies sollte vom Superset-Administrator bereits eingerichtet sein.

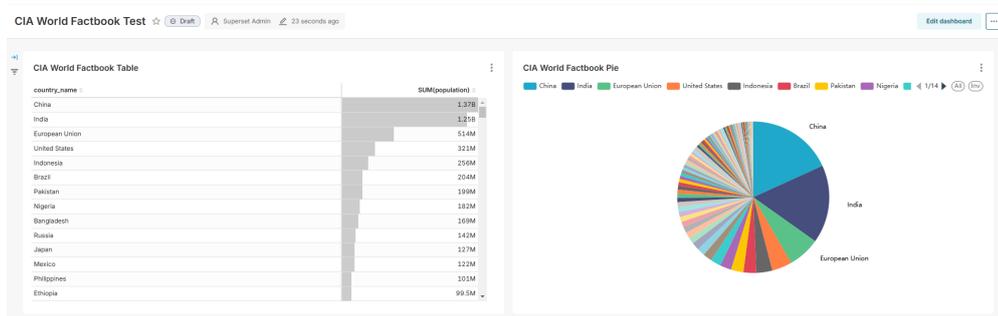


Abbildung 3. Ein Dashboard bearbeiten: Ein Dashboard im View-Mode.

# Eine Tabelle mit einer anderen verknüpfen

Dieses Kapitel zeigt, wie man eine Tabelle mit einer anderen Tabelle verknüpfen kann.

Durch diese Verknüpfung werden der ersten Tabelle weitere Attribute (Felder, Spalten) der weiteren Tabelle(n) in horizontaler Richtung hinzugefügt. Die Verknüpfung geschieht mit der Datenbanksprache SQL über ein bestimmtes Attribut der einen Tabelle (ein Identifikator, ein Primärschlüssel), das mit einem Attribut (einem Fremdschlüssel) einer weiteren Tabelle verbunden wird. Im Hintergrund muss das Datenbanksystem typischerweise ein sogenannter JOIN ausführen.

Zwei Tabellen können mit SQL vom Prinzip her wie folgt verknüpft werden, hier gegeben die fiktiven Tabellen mytable (als die eigene Tabelle) und othertable (als die andere Tabelle) mit PostgreSQL als das zugrundeliegende Datenbankmanagement-System wie bei der Apache Superset Cloud:

```
SELECT *  
FROM mytable  
JOIN othertable ON mytable.id = othertable.id;
```



Beim Verknüpfen von zwei Tabellen müssen die Attribute, die verknüpft werden (d.h. Primär- und Fremdschlüssel), denselben — oder aber einen kompatiblen — Datentyp haben.

Im nachfolgenden Kapitel "Mit Superset eine Tabelle mit einer anderen verknüpfen" gibt es ein konkretes Beispiel dazu.

Wenn die andere Tabelle in einer anderen Database ist, kann diese zurzeit nur durch jemanden mit Datenbank-Administrator-Rechten und guten Datenbank-Kenntnissen verfügbar gemacht werden (siehe [PostgreSQL "Foreign Data Wrappers"](#)).

## Mit Superset eine Tabelle mit einer anderen verknüpfen

Dank der Datenbanksprache SQL gibt es die Möglichkeit, zwei oder mehr Tabellen mit einander zu verknüpfen und vereint als Datenquelle zu verwenden.

Nachfolgend ein Beispiel: Dafür benötigen wir die Tabellen `factbook_data` und `country_codes`: `factbook_data` enthält für jedes Land Kennzahlen wie Population und Fläche (in km<sup>2</sup>). Die beiden Tabellen musst du zuerst als Dateien `country_codes.csv` und `cia_world_factbook_2019.csv` herunterladen: siehe "[Apache Superset-Daten \(CSV\)](#)" [openschoolmaps.ch](#) > [Apache Superset](#) > [Daten](#). Die heruntergeladene Datei ist eine zip-Datei, die du zuerst in ein eigenes Verzeichnis kopierst und dort auspackst ("unzip"). Nun sind die zwei Dateien bereit, um auf Superset hochgeladen zu werden. Das Hochladen von CSV-Dateien ist im Kapitel "Upload von Daten" beschrieben.

Angenommen wir wollen die Bevölkerungsdichte aller Länder als Karte darstellen. Die Bevölkerungsdichte eines Landes ist die Population bezogen auf die Fläche dieses Landes. Wir

verknüpfen dazu die Tabelle `country_codes` mit der Tabelle `factbook_data`, und zwar über den `country_name`, der bei beiden das Land identifiziert:

- Öffnen des Dialogs **SQL > SQL Lab**.
- Eingeben des folgenden SQL-Skripts (dieses sinngemäss anpassen. Zur besseren Lesbarkeit sind die SQL-Schlüsselworte in Grossbuchstaben):

```
SELECT
  cc.*,
  fb.population,
  fb.area,
  CASE
    WHEN fb.area = 0
    THEN 0
    ELSE (fb.population / fb.area * 100)
  END AS density
FROM "country_codes" AS cc
JOIN "factbook_data" AS fb
ON cc.country_name = fb.country_name
;
```



Das Attribut `country_codes` beinhaltet die Namen der Länder und deren ISO 3166-1 alpha-3 Codes, `factbook_data` die Namen und sonstige länderspezifische Daten (Zahlenwerte). Diese Tabellen werden verknüpft damit jene Zahlen auf der *World Map* visualisiert werden können. In diesem Beispiel sind es die Bevölkerungszahl, die Fläche und die Bevölkerungsdichte.

- Im *SQL Editor* die Schaltfläche **[ Run ]** anklicken.
- Drücke neben **[ save ]** auf <sup>□</sup> und wähle *Save dataset*
- Im *Save or Overwrite Dataset* Dialog wähle *Save as new*, gib dem Dataset einen treffenden Namen und drücke **[ Save & Explore ]**
- Im *Create Chart* Dialog muss unter **Query > Query mode** "Aggregate" gewählt werden.
- Schliesslich können unter **Query > Columns > Dimensions** die Spalten eingefügt werden.

Nun lässt sich das Dataset zum erstellen von Charts verwenden.



Falls bei der Ausführung des SQL-Skripts ein Fehler ausgegeben wird, kann dies mehrere Ursachen haben:

1. Du hast das SQL-Skript syntaktisch falsch geschrieben
2. Du hast die falsche Database gewählt (Lösung: SQL und/oder Database anpassen).

Die Bevölkerungsdichte ist wie gesagt das Verhältnis zwischen Population und Fläche, also ein Attributs-Merkmal 'Ratio' (Verhältniszahl) pro Geometrie-Datentyp 'Area'. Gemäss Kapitel "Zuordnung

von Geodaten zu Kartentypen" eignet sich dafür die Choroplethenkarte. In Superset sind das die *World Map* und die *Country Map*.

## Mit Superset die Datenbanksprache SQL nutzen

Der im vorhergehenden Kapitel gezeigte Arbeitsablauf, bei dem es um die Verknüpfung von Tabellen ging, eröffnet die umfangreichen Möglichkeiten der Datenbanksprache SQL. Der Arbeitsablauf kann nämlich auch nur auf eine Tabelle verwendet werden, beispielsweise, um Daten vorzuarbeiten.



Wer SQL näher kennenlernen will, für den gibt es eine Fülle von Quellen. Eine unterhaltsame Einführung für Einsteiger ist [SQL Island](#). Gerne nehmen wir Feedback entgegen (vgl. [OpenSchoolMaps](#) | [Kontakt & Rechtliches](#)) und geben weitere Tipps.

Hier ein Beispiel-Skript mit einer SQL-Window-Funktion, welches den Gebrauch und die Produktion von Elektrizität eines Landes vergleicht und Durchschnittswerte berechnet:

```
WITH temporary AS (  
  SELECT  
    country_name,  
    electricity_production, electricity_consumption,  
    CASE  
      WHEN electricity_production = 0  
      THEN 0  
      ELSE (electricity_consumption / electricity_production * 100)  
    END AS percentage  
  FROM factbook_data  
)  
SELECT  
  *,  
  AVG(electricity_production) OVER () AS avg_production,  
  AVG(electricity_consumption) OVER () AS avg_consumption  
FROM temporary  
WHERE percentage IS NOT NULL  
;
```

# Datentypen

Ein Datentyp bestimmt den erlaubten Wertebereich eines Attributs. Beispiele von Datentypen sind Text oder Ganzzahl.

Superset kennt folgende **Basis-Datentypen**:

## Zeichenkette (en: Text)

In Superset mit dem Icon "abc" dargestellt. Das können z.B. die Typen **TEXT** oder **VARCHAR(255)** sein, wobei der Wert in Klammern die maximale Länge der Zeichenkette bestimmt.

## Datum & Zeit (en: Timestamp)

In Superset mit dem Icon "Uhr" dargestellt. Das ist ein "temporaler" d.h. ein zeitlicher Typ mit Datum- und Zeit. Ein zeitlicher Datentyp kann mit oder ohne Zeitzone abgespeichert sein. Diese Datentypen heissen technisch **TIMESTAMP WITH TIMEZONE** und **TIMESTAMP WITHOUT TIMEZONE**. Eine genormte Darstellung eines **TIMESTAMP WITH TIMEZONE** in MEZ sieht z.B. so aus "2019-11-10 12:21:03.000000+01".

## Zahl (en: Number)

In Superset mit dem Icon "#" dargestellt. Das sind in Apache Superset Ganzzahlen. Technisch heissen diese **INTEGER** und **SMALLINT**.

## Fließkommazahl (en: Real) und Festkommazahl (en: Numeric)

Keine Icons in Superset. Technisch heissen diese **REAL** und **NUMERIC**. Bei **NUMERIC** gibt der erste Wert an, wie viele Stellen die Zahl insgesamt hat und der Zweite gibt an, wie viele Nachkommastellen es gibt. Z.B. kann **Numeric(5,2)** alle reellen Zahlen von -999.99 bis 999.99 speichern.

## Metrik (en: Metrics)

In Superset als "f(x)" dargestellt. Aggregations-Typ, z.B. **COUNT(\*)**. Superset fügt bei Tabellen immer dieses abgeleitete Attribut hinzu.

Zeitliche Daten sind wichtig in jeder Datenanalyse. Sie werden u.a. zum Filtern der Daten verwendet. Die Tatsache, dass ein Attribut vom Typ Timestamp ist, wird durch dessen Attributtyp direkt in der Datenbank-Tabelle bestimmt, zusätzlich zu der Kennzeichnung als temporaler Typ, die in Superset noch manuell gesetzt werden muss. (Bei der Tabellenansicht, Checkbox *is temporal* unter *Columns*).

Es existieren mehr Basis-Datentypen (beispielsweise **CHAR(3)** und **BOOLEAN**) als diejenigen in Apache Superset, die oben erwähnt wurden.



Die Schweiz liegt in der Zeitzone "Mittleuropäische Zeit" (MEZ; englisch Central

European Time, CET). Diese bezieht sich auf den 15. Längengrad Ost. Die Differenz der MEZ zur Weltzeit (Coordinated Universal Time, UTC) beträgt eine Stunde, was mit +1 zum Ausdruck kommt. Während des Sommerhalbjahrs in Europa/Schweiz beträgt die Differenz zur UTC effektiv +2 Stunden.

## Datentyp-Rollen und raumbezogene Datentypen

Die oben erwähnten Basis-Datentypen genügen alleine nicht den vielfältigen Anforderungen von Daten. Wie die zeitlichen (en: temporal) sind auch die raumbezogenen (en: spatial) Daten interessant für die Analyse und Visualisierung. Raumbezogene/räumliche Datentypen - oft auch Geodaten genannt - haben besondere Eigenschaften. Superset verlangt darum mit Blick auf bestimmte Charts zusätzlich die Angabe einer "**Datentyp-Rolle**" (en: Data type role) wie beispielsweise "Point" (de: Punkt). Für die Datenbank sind diese Daten weiterhin beispielsweise vom Basis-Datentyp Text, doch Superset kann diese dann speziell interpretieren. Indem der Benutzer die Datentyp-Rolle angibt, ist es Superset möglich, beispielsweise Daten als Karten (Map Charts) zu visualisieren.

Die Geometrie ist ein bekannter raumbezogener Datentyp. Die vier wichtigsten **Geometrie-Datentypen** heißen genormt **Point** (de: Punkt; Synonyme Koordinate, Punktkoordinate, Ort, Lage), **LineString** (de: Linie; Synonyme Line, Polyline, Way) und **Polygon** (de: Fläche; Synonym: Area) sowie "Volumen" (en: Volume). Ein Punkt bestimmt die Position/Lage eines Objekts und besteht aus einem Paar von **Koordinaten**. Für die internationale Angabe von Positionen auf der Erde — beispielsweise mit dem GPS — werden **Längengrade** und **Breitengrade** verwendet (en: Longitude/Latitude, abgekürzt: Lon/Lat).

Übersicht über raumbezogene Datentyp-Rollen in Apache Superset:

<b>Datentyp-Rolle "Point"</b>	Gibt es vor allem als Rolle "Longitude & Latitude columns". Die entsprechenden Datentyp-Encodings sind "Delimited" oder "Geohash" (unten mehr dazu).
<b>Datentyp-Rolle "Longitude &amp; Latitude columns"</b>	Zwei untrennbar miteinander verbundene Attribute in der Tabelle, Längengrad und Breitengrad, die vom Datentyp Fließkommazahl (z.B. Fließkommazahl float8) sind. Diese Rolle ist eine Variation der Datentyp-Rolle "Point".
<b>Datentyp-Rolle "LineString"</b>	Attribute vom Basis-Datentyp Text (abc in Superset), welche die Datentyp-Encodings "JSON" oder "Polyline" (WKT) haben.
<b>Datentyp-Rolle "Polygon"</b>	Attribute vom Basis-Datentyp Text (abc in Superset), welche das Datentyp-Encoding "JSON" hat. Dazu kommen das Datentyp-Encoding "Polyline" (WKT) sowie "ISO 3166-2 Codes".



Längengrade und Breitengrade basieren auf einem weltweit einheitlichen,

geodätischen Koordinatensystem, dem "World Geodetic System 1984" (WGS 84). Lat/Lon werden oft als Fließkommazahlen behandelt. Eigentlich genügt für die meisten Anwendungen eine Festkommazahl mit maximal 6 Nachkommastellen, also **NUMERIC(9,6)** (Beispiel von Lat/Lon für das Hauptgebäude der OST Rapperswil, welches man z.B. auf [www.osm.org](http://www.osm.org) prüfen kann: 47.223316, 8.8175570).

Im folgenden Kapitel wird erklärt, was mit dem Datentyp-Encoding gemeint ist.

## Datentyp-Encodings

Mit der Angabe eines Datentyps und einer "Datentyp-Rolle" ist es noch nicht getan: Es muss zusätzlich noch festgelegt sein, wie der Inhalt eines Datentyps kodiert ist. Man nennt dies Datentyp-Encoding (de: Datentyp-Kodierung).



Sicherlich vielen bekannt sind Probleme mit Umlauten äöü. Diese Probleme haben mit dem Datentyp-Encoding vom Datentyp Text (Zeichenkette) zu tun. Es gibt bei Text (Zeichenkette) unter anderem Kodierungen wie UTF-8, ANSI oder ASCII. Wir empfehlen in jedem Fall **UTF-8!**

Koordinaten sind nicht die einzige Möglichkeit zur Positionierung und Lagebestimmung von Objekten. Es gibt auch andere Georeferenzierungs-Systeme, wie die sogenannten **ISO 3166-2 Codes** und die **Geohashes**. Alle Georeferenzierungs-Systeme - Koordinate, ISO 3166-2 Code und Geohash - haben eigene Kodierungen.

In Superset kommen folgende **Datentyp-Encodings** vor:

- Bei der Datentyp-Rolle "Point" ist es "Delimited" oder "Geohash".
- Bei der Datentyp-Rolle "LineString" (Line) sind es "Polyline" oder "JSON".
- Bei der Datentyp-Rolle "Polygon" sind es (geschlossene) "Polyline" oder "ISO 3166-2 Codes".

Nachfolgend sind einige Informationen zu den Datentyp-Encodings zusammengestellt:

**Delimited**            Zum Beispiel "47.22311, 8.81636" sind die Koordinaten des Gebäudes 4 der OST Rapperswil. Siehe <https://www.openstreetmap.org/search?query=47.22311,8.81636> (man beachte die Link-Adresse).

## Geohash

Geohashing ist ein Georeferenzierungs-System, das einen geografischen Standort in eine kurze Folge von Buchstaben und Ziffern kodiert. Es ist eine hierarchische räumliche Datenstruktur, bei der die Erde in Kacheln aufgeteilt wird, die weiter unterteilt werden können. Den Kacheln werden Buchstaben und Ziffern zugeordnet. D.h. je länger ein Geohash-Code ist, desto genauer die Ortsangabe. Geohash-Codes sind auf [Wikipedia](#) frei dokumentiert und haben im Vergleich zu Koordinaten unter anderem die Eigenschaft, dass sie kurze Zeichenketten sind. Geohashes werden unter anderem auch von OpenStreetMap und Geocaching verwendet. Nimmt man z.B. die Koordinaten des Gebäudes 4 der OST Rapperswil (47.22311, 8.81636) als Beispiel, ergibt sich der Geohash "u0qk8seys1d". Man kann dies mit der [Webapp Geohash.org](#) nachprüfen.

## Polyline (WKT)

Damit ist eigentlich ein Encoding gemäss "Well Known Text" (Abk. WKT) gemeint. Ein Beispiel ist "LineString(47.22311 8.81636, 47.22469, 8.8173)" für die Luftlinie zwischen der OST und dem Bahnhof Rapperswil. Siehe z.B. [WKT auf Wikipedia](#).

## JSON

JSON als Abkürzung von "JavaScript Object Notation" ist ein Datenformat in einer menschenlesbaren Textform zum Zweck des Datenaustauschs zwischen Web-Anwendungen. Hier im Zusammenhang mit Map Charts ist mit JSON eigentlich GeoJSON gemeint. GeoJSON ist ein genormtes Datenformat, um geografische Daten nach der Simple-Feature-Access-Spezifikation zu repräsentieren. GeoJSON kann u.a. mit der Website [geojson.io](#) dargestellt und editiert werden. Und mit dem Web-Tool [geojsonlint.com](#) können GeoJSON-Daten geprüft werden.

## ISO 3166-2-Code

Map Charts wie *Country Map* und *World Map* können u.a. auch ISO 3166-2-Codes von Ländern und Untergliederungen davon (Schweizer Kantone, französische Departemente, kanadische Provinzen) verarbeiten. ISO 3166-2 ist Teil der von der internationalen Organisation für Normung (ISO) veröffentlichten Norm ISO 3166. Siehe auch [Apache Superset-Dokumentation](#) und [Wikipedia](#).

# Geokodierung

Auch geografische Namen wie "Rapperswil" oder Adressen wie "Oberseestrasse 10, Rapperswil-Jona" bilden eine Art textuelles Georeferenzierungs-System, wenn auch ein mehrdeutiges. Die Interpretation geografischer Namen ist ein aufwändiger Prozess und nennt sich **Geokodierung**.

Die Geokodierung ist ein Algorithmus und ein Prozess, der eine Beschreibung einer Position (Lage, Ort) — beispielsweise ein Geohash, eine Postadresse oder einen Ortsnamen — in eine Koordinate auf der Erdoberfläche wandelt.

Hier als Beispiel einer Geokodierung der Gebäudeadresse "Oberseestrasse 10, Rapperswil-Jona" ein [Geokodierungs-Aufruf mit OpenStreetMap](#).



In Apache Superset ist zurzeit - nebst den ISO 3166-2 Codes und den Geohashes -

keine Geokodierung von **Gebäude- und Postadressen** eingebaut. Wobei es durch die Verknüpfung mit einer anderen Tabelle immer möglich ist, geografische Namen(-skürzel) mit Codes zu verknüpfen: Siehe dazu das Beispiel im Kapitel oben "Mit Superset eine Tabelle mit einer anderen verknüpfen".

# Der Visualisierungsprozess und die Superset Map Charts

Visualisieren bedeutet Daten auf grafische Symbole oder Diagramme abzubilden. Auch Apache Superset ist so aufgebaut: Der Visualisierungs-Prozess beginnt mit der Wahl der Datenquelle (Tabelle), dann wird ein passender Chart ausgewählt, und schliesslich wird der Chart einem Dashboard zugeordnet.

Datentypen helfen, die Entscheidung zur Auswahl eines Charts oder Kartentyps zu vereinfachen. Zu Beginn dieses Visualisierungs-Prozesses lohnt es sich, sich mit den Daten vertraut zu machen. Das ist die Domäne der beschreibenden Statistik und des Data Engineerings. Wer die Merkmale eines Attributs kennt, findet die in Frage kommenden Charts schneller.

## *Qualitativ und quantitativ*

Man unterscheidet zwischen qualitativen und quantitativen Attribut-Merkmalen. Qualitative Merkmale sind solche, die sich nicht durch Zahlen direkt erfassen lassen; deren Datentyp ist meist Text. Beispiele dazu sind Vornamen, Firmennamen, etc. Quantitative Merkmale dagegen sind durch Zahlen bestimmbar. Diese Merkmale lassen sich durch wiegen, messen, abzählen usw. bestimmen.

## *Attributs-Merkmals-Levels*

Die quantitativen Merkmale lassen sich ihrerseits differenzieren in vier "Attributs-Merkmals-Levels" (Quelle: Stanley Stevens, 1946, "One the theory of Scales of Measurement"):

<b>Nominal</b>	Ein Name, Klassenname ("etwas Benanntes"). Meist Datentyp Text.
<b>Ordinal</b>	Eine Ordnungszahl, eine Rangierung, ("etwas zum Vergleichen"). Meist Datentyp Ganzzahl.
<b>Intervall</b>	Ein massstabsloser Zahlenwert, Verhältniszahl ("etwas Relatives"). Meist Datentyp Fließ- oder Festkommazahl.
<b>Ratio</b>	Ein Zahlenwert mit absolutem Nullpunkt; meist Datentyp Fließ- oder Festkommazahl.

## Kartentypen

In der Geovisualisierung wird der oben beschriebene Visualisierungs-Prozess auf Geodaten und Karten angewendet. Dabei werden die Geodaten den Kartensymbolen zugeordnet; dies nennt man Symbolisierung (en: symbolization, styling).

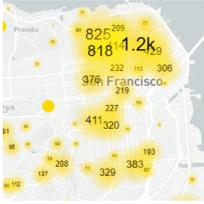
Es gibt theoretisch folgende Kartentypen:

- Standortkarten
- Netzkarten
- Choroplethenkarten
- Ortsdiagrammkarten

Die Kartentypen aus der Kartografie helfen, die Superset Map Charts zu gruppieren. Es werden hier nur die bekanntesten Kartentypen vorgestellt. Superset deckt nicht alle Kartentypen ab, stellt aber zurzeit schon zwölf verschiedene Map Chart-Typen zur Verfügung (vgl. auch der Anhang):

**Standortkarten** (en: Dot Map) werden symbolisiert mit Punktsignaturen (beispielsweise rote Symbole/Marker) bezogen auf Punkte/Orte. Zu diesem Kartentyp gehören folgende Superset Charts:

- Mapbox** Standortkarte mit Point Clustering. Erwartet die Datentyp-Rolle "Longitude & Latitude columns".
- Deck.gl Scatterplot** Standortkarte ohne Point Clustering. Erwartet die Datentyp-Rolle "Longitude & Latitude columns".

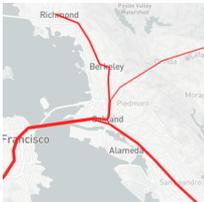
Mapbox	Deck.gl Scatterplot
	



Unter Point Clustering versteht man das "Verschmelzen" von Punkten beim Herauszoomen. Siehe Abbildung 1 und 2 im Anhang.

**Netzarten** (en: Network map) (deutsches Synonym: Vektorkarten), werden symbolisiert mit Liniensignaturen (beispielsweise rote Linien) bezogen auf Linien. Zu diesem Kartentyp gehören folgende Superset Charts:

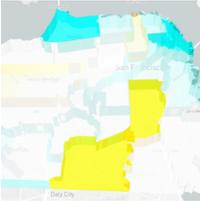
- Deck.gl Path** Netzkarte. Erwartet die Datentyp-Rolle "LineString".
- Deck.gl Arc** Netzkarte mit Linien, die aus 2 Endpunkten bestehen und die Kurven kürzester Distanz auf der Erdoberfläche (Orthodrome) darstellen, beispielsweise zur Visualisierung von Flügen oder Schiffswegen. Erwartet die Datentyp-Rolle "Longitude & Latitude columns".

Deck.gl Path	Deck.gl Arc
	

**Choroplethenkarten** (en. Choropleth Map) (deutsche Synonyme: Dichtemosaiikkarten; Flächenkartogramme) werden symbolisiert mit Flächensignaturen bezogen auf Flächen. Die Werte sind hier meist nicht absolut, sondern relativ. Zu diesem Kartentyp gehören folgende Superset Charts:

- Country Map** Choroplethenkarte mit Ländergrenzen. Erwartet ISO 3166-2-Codes von Ländern und Untergliederungen davon als Zeichenkette.

**Deck.gl Polygon** Choroplethenkarte mit Flächen. Erwartet die Datentyp-Rolle "Polygon".

Country Map	Deck.gl Polygon
	

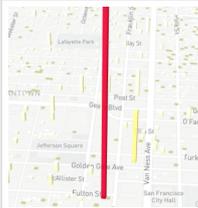
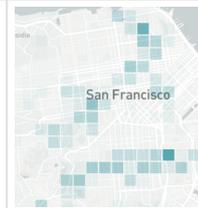
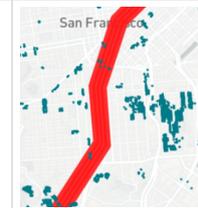


Ein häufiger Fehler in Choroplethenkarten ist, wenn die dargestellten Werte absolut sind, beispielsweise "Summe der Traktoren pro Land". Richtig wäre eine relative Angabe wie "Traktoren pro km<sup>2</sup>", d.h. die absolute Summe durch die Landesfläche geteilt werden.

**Ortsdiagrammkarten** (en. Symbol Map) werden symbolisiert mit lokalen Business-Diagrammen bezogen auf Orte bzw. Flächen. Man beachte den Unterschied von Symbol/Marker zu Diagramm. Zu diesem Kartentyp gehören folgende Superset Charts:

- World Map** Ortsdiagrammkarte 2D. Erwartet ISO 3166-2-Codes von Ländern und Untergliederungen davon als Zeichenkette.
- Deck.gl 3D Hexagon** Ortsdiagrammkarte 3D. Erwartet die Datentyp-Rolle "Longitude & Latitude columns".
- Deck.gl Grid** Variante von Deck.gl 3D Hexagon, jedoch mit rechteckigen Säulen. Erwartet die Datentyp-Rolle "Longitude & Latitude columns".
- Deck.gl Screen Grid** Ähnlich wie Deck.gL Grid und Hexagon. Erwartet die Datentyp-Rolle "Longitude & Latitude columns".
- Deck.gl GeoJSON** (Bemerkung: Funktioniert in Superset momentan nicht!).

Ein weiterer Map Chart ist **Deck.gl Multiple Layers**. Dieser "Chart" erlaubt die Kombination von mehreren Deck.gl-Charts, zusammen mit einer Basiskarte.

World Map	Deck.gl 3D Hexagon	Deck.gl Grid	Deck.gl Screen Grid	Deck.gl Multiple Layers
				



Dies ist nur eine Auswahl der wichtigsten Kartentypen. Sie ist unvollständig. Eine mögliche Quelle zum Nachlesen ist [GITTA.info](https://gitta.info).

# Zuordnung von Geodaten zu Kartentypen

Die Tabelle unten zeigt die Zuordnung von Geometrie-Datentypen auf der horizontalen Achse und Attributs-Merkmal-Levels auf der vertikalen Achse zu Kartentypen:

Tabelle 1. Kartentypen eingeteilt nach Geometrie-Datentypen und Attributs-Merkmal-Levels (Quelle: David Unwin, 1981, "Introductory Spatial Analysis", London Methuen)

	<b>Point</b>	<b>Line</b>	<b>Area</b>
<b>Nominal</b>	Dot map	Network map	Colored area map
<b>Ordinal</b>	Symbol map	Ordered network map	Ordered colored map
<b>Intervall</b>	Graduated symbol map	Flow map	Choropleth map
<b>Ratio</b>			

Aus Erfahrung heraus sind die am häufigsten verwendeten Map Charts einerseits *Mapbox* (Kartentyp Dot Map), wenn der Geometrie-Datentyp punkthaft (Point) ist, und andererseits *Country Map* (Choropleth Map), wenn der Geometrie-Datentyp flächenhaft (Area) ist.

# Charts und Dashboards anpassen

## Dashboard-Optionen

Ein Dashboard kann die Darstellungs-Konfiguration (z.B. Chart-Farben) der Charts überschreiben. Dies ist möglich, indem man die Metadaten eines Dashboards editiert. Die Einstellungen findet man unter **Dashboards** > **Entsprechendes Dashboard** > **Edit** > **Advanced** > **JSON metadata**. Hier wirst du ein Textfeld namens "JSON Metadata" finden. Dies könnte z.B. so aussehen (es kann auch sein, dass es leer ist):

```
{
  "filter_immune_slices": [],
  "timed_refresh_immune_slices": [],
  "filter_immune_slice_fields": {},
  "expanded_slices": {},
  "refresh_frequency": 0,
  "default_filters": "{}"
}
```

Die Parameter gelten allgemein für alle Dashboards:

- `filter_immune_slices`: Array von `slice_ids` (int) von Charts, die nicht gefiltert werden sollen.

```
"filter_immune_slices": [324, 65, 92],
```

- `timed_refresh_immune_slices`: Array von `slice_ids`, die vom Browser nicht automatisch neu geladen werden sollen.

```
"timed_refresh_immune_slices": [324]
```

- `filter_immune_slice_fields`: Hier kannst du einstellen, ob bei gewissen Charts spezifische Felder nicht filterbar sein sollen.

```
{
  "filter_immune_slice_fields": {
    "177": ["country_name", "__time_range"],
    "32": ["__time_range"]
  }
}
```

- `refresh_frequency`: Anzahl Sekunden bis der Browser die Daten vom Server automatisch neu lädt (Ganzzahl in Sekunden).

```
"refresh_frequency": 5
```

- `default_filters`: Setzt Default-Werte in den Filter. Die `slice_id` des Filters muss gegeben sein.

```
"default_filters": "{\"95\": {\"country_name\": [\"Kenya\"]}}"
```



Eine Liste aller Parameter ist ebenfalls in der [Doku](#) zu finden.

## Chart-Farben im Dashboard definieren



Damit die gewünschten Farben im Dashboard angewendet werden, muss bei den einzelnen Charts unter *Customize* das Farbschema **Airbnb Colors** gesetzt sein.

Um die Farbe von einem Chart abzuändern, musst du `"label_colors": {"key": "color"}`, hinzufügen. Hier ein Beispiel wie das aussehen könnte:

```
{
  "label_colors": {
    "girl": "#0200ff",
    "boy": "#00ff00"
  },
  "filter_immune_slices": [],
  "timed_refresh_immune_slices": [],
  "filter_immune_slice_fields": {},
  "expanded_slices": {},
  "refresh_frequency": 0,
  "default_filters": "{}"
}
```

Die "Keys" wie z.B. "girl" kannst du in der Legende des jeweiligen Charts finden.



Alle Optionen müssen gültiges JSON sein. Der ["JSON Editor Online"](#) kann bei der Bearbeitung helfen.

## Tooltips mit JavaScript anpassen

Jeder Punkt bei z.B. Scatterplot zeigt dessen Längengrad und Breitengrad als Tooltip-Text an. Dieser ist beim Chart unter **Advanced** > **Javascript tooltip generator** anpassbar.



Für diese Anpassung muss das Schreiben von JavaScript aktiviert sein (ENABLE-JAVASCRIPT-CONTROL). Kontaktiere den Sys-Admin falls nötig.

Im Textfeld muss man eine Funktion schreiben, die einen String zurückgibt, der dann als Tooltip angezeigt wird. Hier ein einfaches Beispiel, welches den Tooltip zu "test" setzt:

```
function myFunction(punkt) {  
  return 'test'  
}
```

Um einen sinnvolleren Tooltip zu erhalten, muss man zuerst die Daten auswählen, die man nutzen will. Das geht durch das Hinzufügen von den Spalten unter *Extra data for JS*.

Wenn du dir jetzt für jeden Punkt einen Tooltip anzeigen lassen willst, mit Informationen von einer Spalte mit dem Namen "Name", musst du die Funktion so schreiben:

```
function myFunction(punkt) {  
  return punkt.object.extraProps.Name  
}
```



Die Funktion muss ein gültiger JavaScript-Code sein. JavaScript ist eine Computersprache, die entsprechende Kenntnisse und Tools verlangt.

# Charts mit Annotations ergänzen

*Annotations* sind Text- und Grafik-Notizen, die als Anmerkung über Charts gelegt werden können. Je nach Annotations-Typ ist es entweder eine Linie oder ein gefärbtes Rechteck im Hintergrund. Sie können dazu benutzt werden, allgemeine Informationen (z.B. weltweites Bevölkerungswachstum) darzustellen, ohne diese als Daten in einer Tabelle ablegen zu müssen.

*Annotation Layers* dienen dazu, Annotations zu bündeln. Ein Layer kann mehrere Annotations beinhalten und man bindet in Diagrammen jeweils den Annotation-Layer ein, der dann alle Annotations darstellt.

Es gibt vier Annotations-Typen:

<b>Ereignis (en: Event)</b>	Basierend auf Tabellendaten, wird als Linie dargestellt.
<b>Intervall (en: Interval)</b>	Wie Event, allerdings mit Start- und Enddatum, wird als "Range" gerendert.
<b>Zeit (Time Series)</b>	Linie basierend auf beliebiger Time Series.
<b>Formel (en: Formula)</b>	Linie basierend auf einer Formel (z.B. $2x$ oder $0.005x$ ).

## Erstellen von Annotations

Ereignis- und Intervall-Annotations sind unter **Settings** > **Annotation Layers** der Superset-Menüleiste erstellbar. Die *Description* einer *Annotation* wird im Chart als Beschreibungstext vom Ereignis oder Intervall dargestellt. Das Start- und End-Datum dient dem Intervall als Zeitspanne, dem Ereignis nur das Start-Datum als fixer Zeitpunkt. Diese Art von *Annotation* muss zu einem *Annotation-Layer* hinzugefügt werden.

Ereignis- und Intervall-Annotations können auch mit einer *Table* erstellt werden. In der Datenbank-Tabelle soll es für die Annotations eine Kolonne für die Beschreibung (Text), eine für das Start-Datum (Date) und evtl. für das End-Datum (Date) vorhanden sein.



Die Datenbank-Tabelle kann nicht direkt in den *Annotations and Layers* referenziert werden. Eine *Table* mit jener Tabelle, die alle benötigten Kolonnen visualisiert, muss zuerst erstellt werden. Diese *Table* dient dann als *Annotation Source* des anderen Diagramms.

Die Daten für Zeit-Annotations werden von einem beliebigen *Line Chart* geholt. Dessen "Verlauf" ist in anderen Diagrammen mit einer Zeit-Achse visualisierbar.

## Ereignis-Annotations

Ereignisse (en: Events) sind eine Liste von Zeitpunkten (Datum) mit Beschreibungen. Sie werden als vertikale Linien dargestellt. Ihre Beschreibung wird in den Tooltips angezeigt, wenn man mit der Maus über sie fährt.

## Intervall-Annotationen

Genau wie Events, aber mit einem Start- und Enddatum, werden sie als Bereich dargestellt.

## Zeitreihen-Annotationen

Zeitreihen werden als zusätzliche Linie auf dem Liniendiagramm hinzugefügt. Der Name und die Anzeigeeigenschaften können konfiguriert werden.

## Formel-Annotationen

Formeln werden als zusätzliche Linie in das Diagramm eingefügt. Sie können einen mathematischen Ausdruck eingeben, der auf der Client-Seite mit "mathjs" ausgewertet wird. Man kann die Anzeigeeigenschaften festlegen und beliebige mathematische Formel angeben.

# ANHANG: Die elf Map Charts von Apache Superset

Hier siehst du die elf Map Charts von Apache Superset und deren Auswahlmöglichkeiten.



Die folgenden Screenshots basieren auf einer älteren Version von Apache Superset. Die grösste Änderung gegenüber der Version von Apache Superset Stand Februar 2025, ist das Verschieben von **Data > Time > Time Column** und Time Range, die nun unter **Data > Query > Filters** einzustellen wären.

The screenshot displays the Apache Superset interface. On the left is the 'DATA' and 'CUSTOMIZE' sidebar, and on the right is the 'RESULTS' and 'SAMPLES' panel.

**DATA CUSTOMIZE:**

- datetime
- TIME RANGE: No filter
- Query
- LONGITUDE: # LON
- LATITUDE: # LAT
- CLUSTERING RADIUS: 60
- ROW LIMIT: 10000
- FILTERS: + Drop columns/metrics here or click
- DIMENSIONS: + Drop columns here or click
- Points: v
- Labelling: v
- UPDATE CHART

**RESULTS SAMPLES:**

LAT	LON
37.7079905	-122.4512898
37.7080478	-122.4512067
37.708131	-122.451149
37.7081463	-122.4617814
37.7081627	-122.4453473

Abbildung 4. Visualization Type Mapbox, rechts am Beispiel Punkte in San Francisco, Kalifornien, links der dazugehörige Dialog

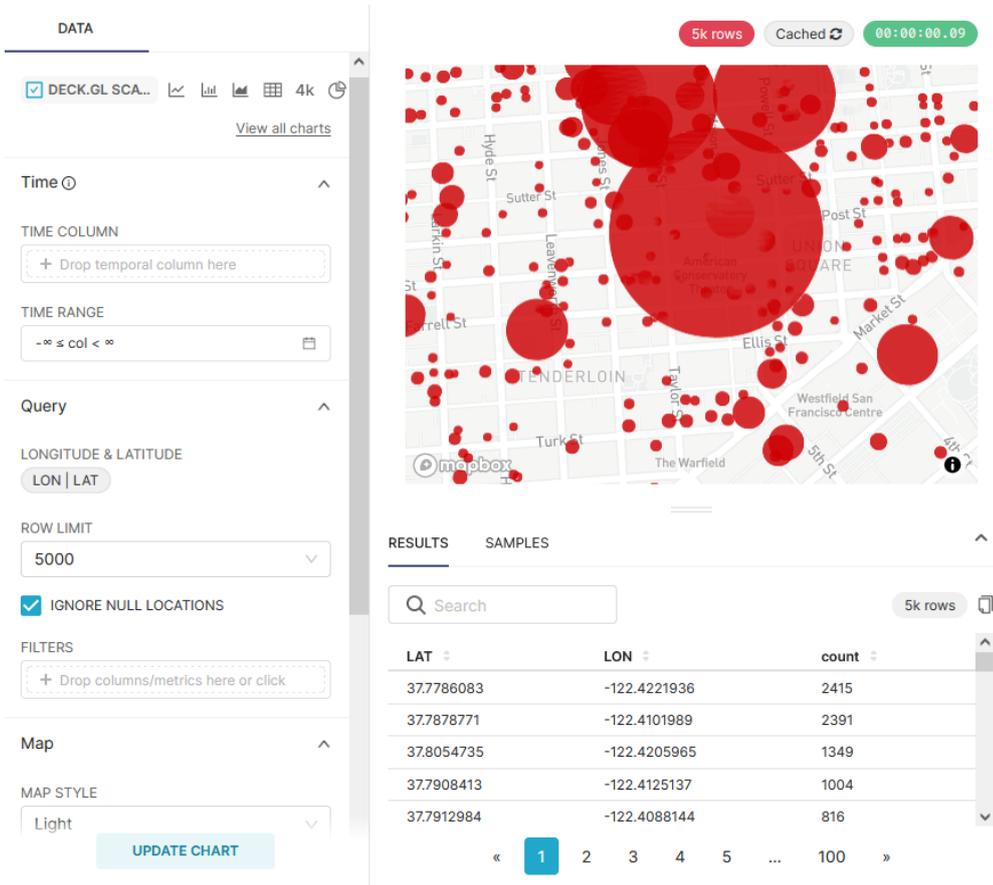


Abbildung 5. Visualization Type Deck.gl Scatterplot (entspricht scatter plot in Excel), rechts am Beispiel Punkte in San Francisco, Kalifornien, links der dazugehörige Dialog

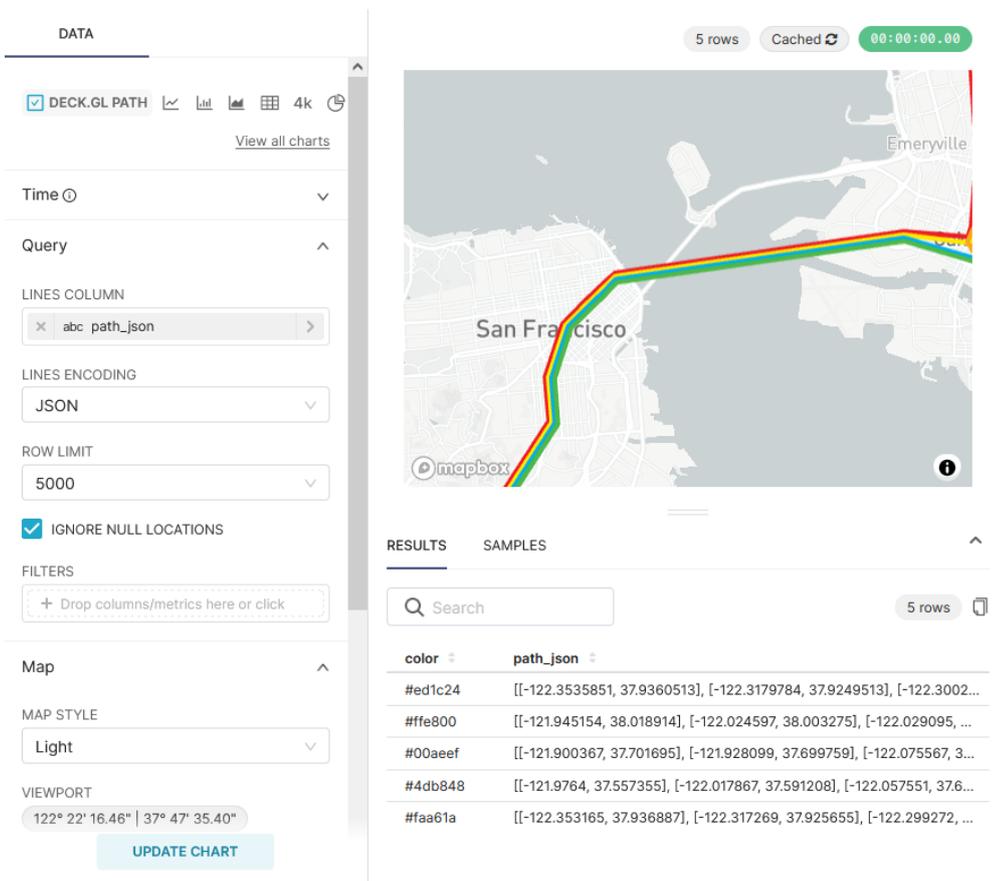


Abbildung 6. Visualization Type Deck.gl Path, rechts am Beispiel Routen in Kalifornien, links der dazugehörige Dialog

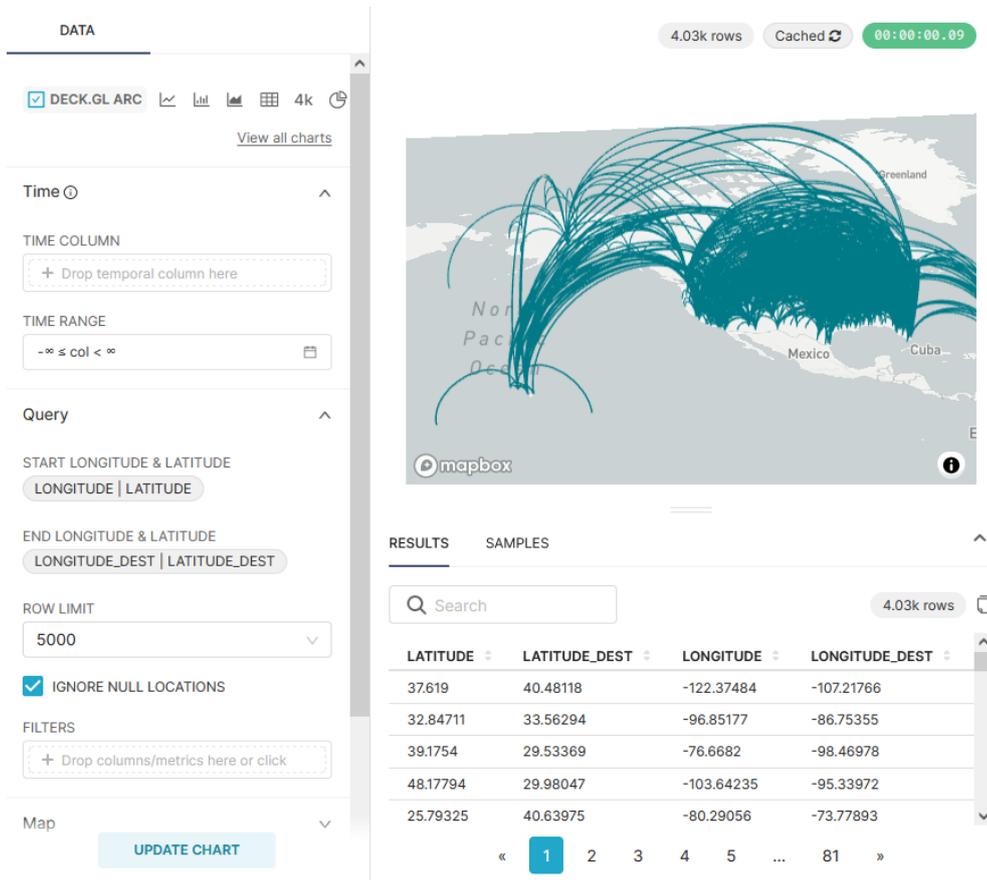


Abbildung 7. Visualization Type Deck.gl Arc, rechts am Beispiel Inland-Flüge in den USA, links der dazugehörige Dialog

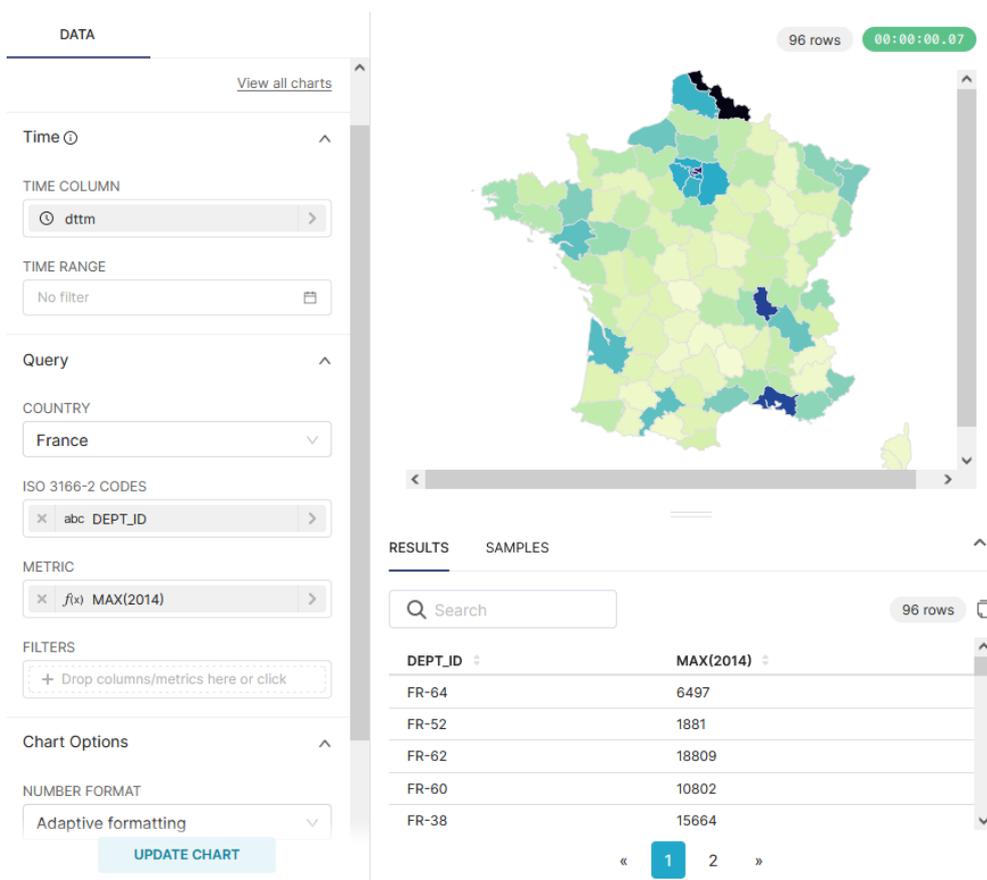


Abbildung 8. Visualization Type Country Map, rechts am Beispiel die Geburtenrate 2014 in den französischen Departements, links der dazugehörige Dialog

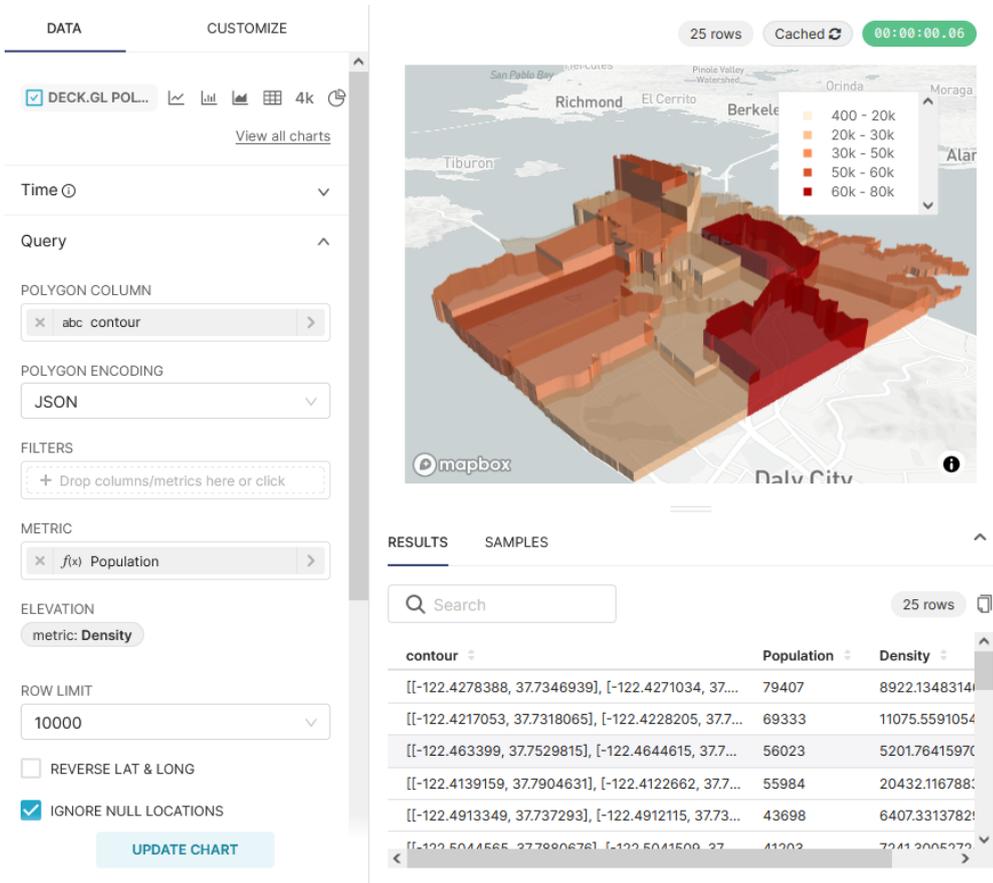


Abbildung 9. Visualization Type Deck.gl Polygon, rechts am Beispiel Bezirke in San Francisco und deren Bevölkerungsdichte, links der dazugehörige Dialog

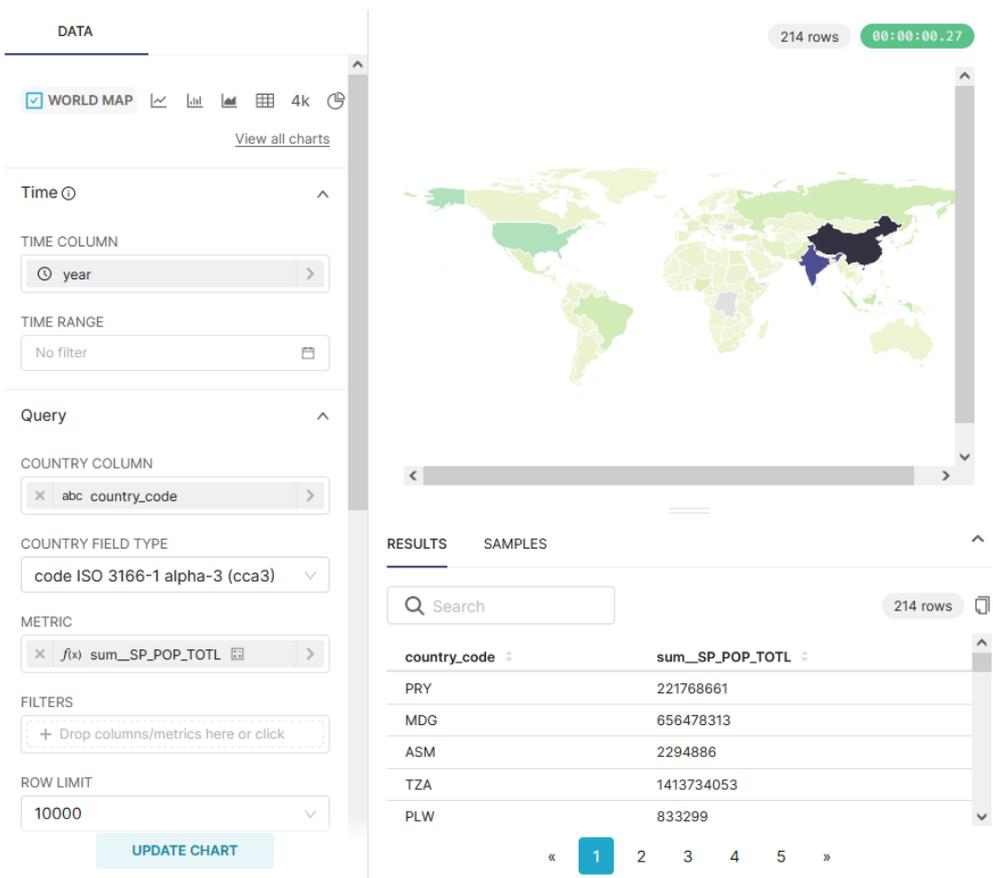


Abbildung 10. Visualization Type World Map, rechts die Bevölkerungsdichte der Länder, links der dazugehörige Dialog

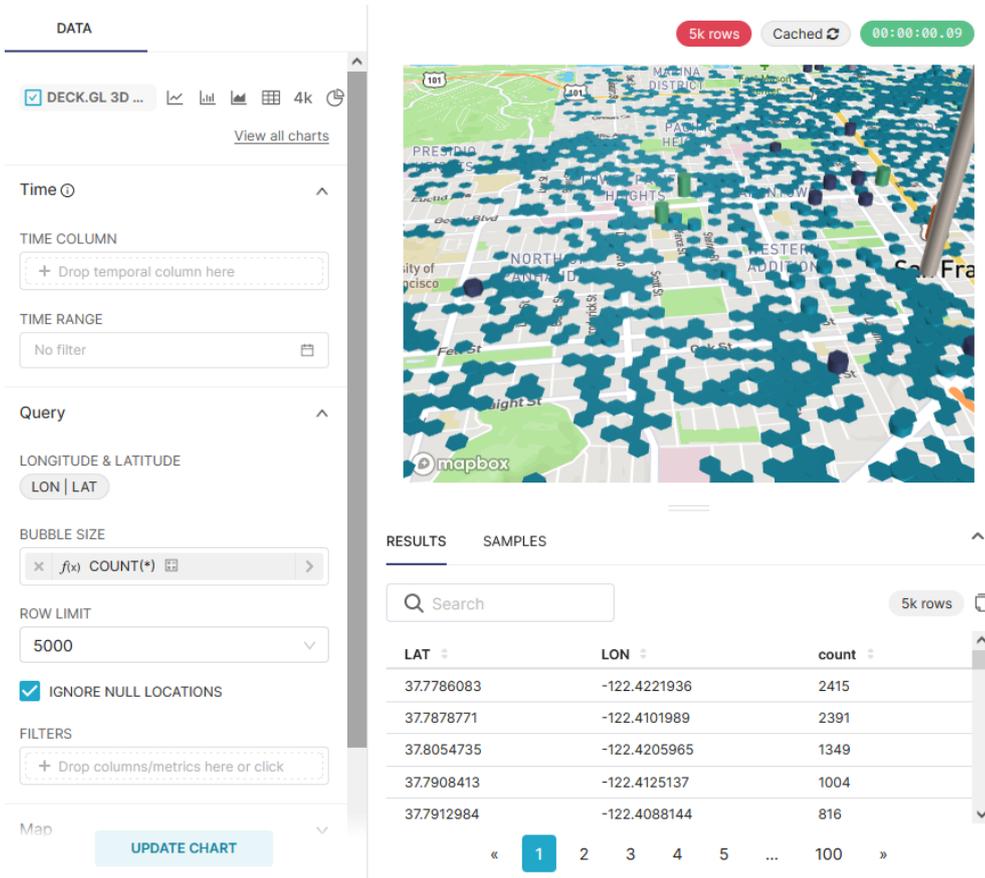


Abbildung 11. Visualization Type Deck.gl 3D Hexagon, rechts am Beispiel Punkte in San Francisco, Kalifornien, links der dazugehörige Dialog

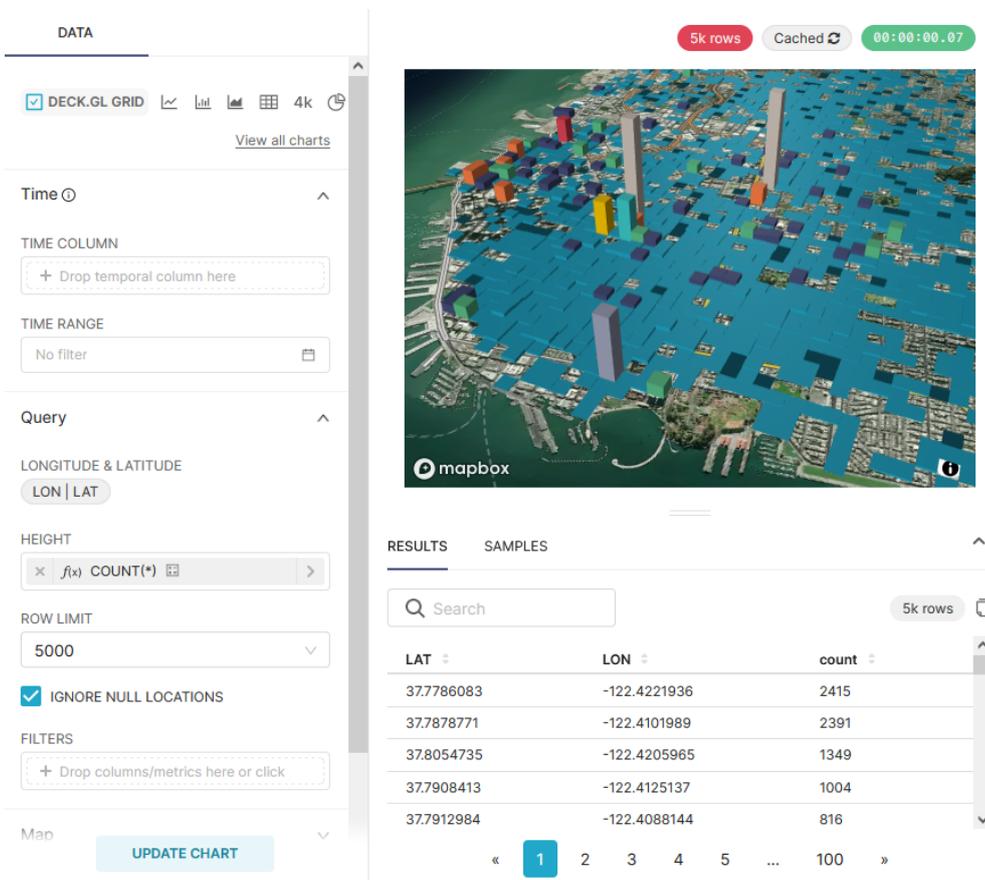


Abbildung 12. Visualization Type Deck.gl Grid, rechts am Beispiel Punkte in San Francisco, Kalifornien, links der dazugehörige Dialog

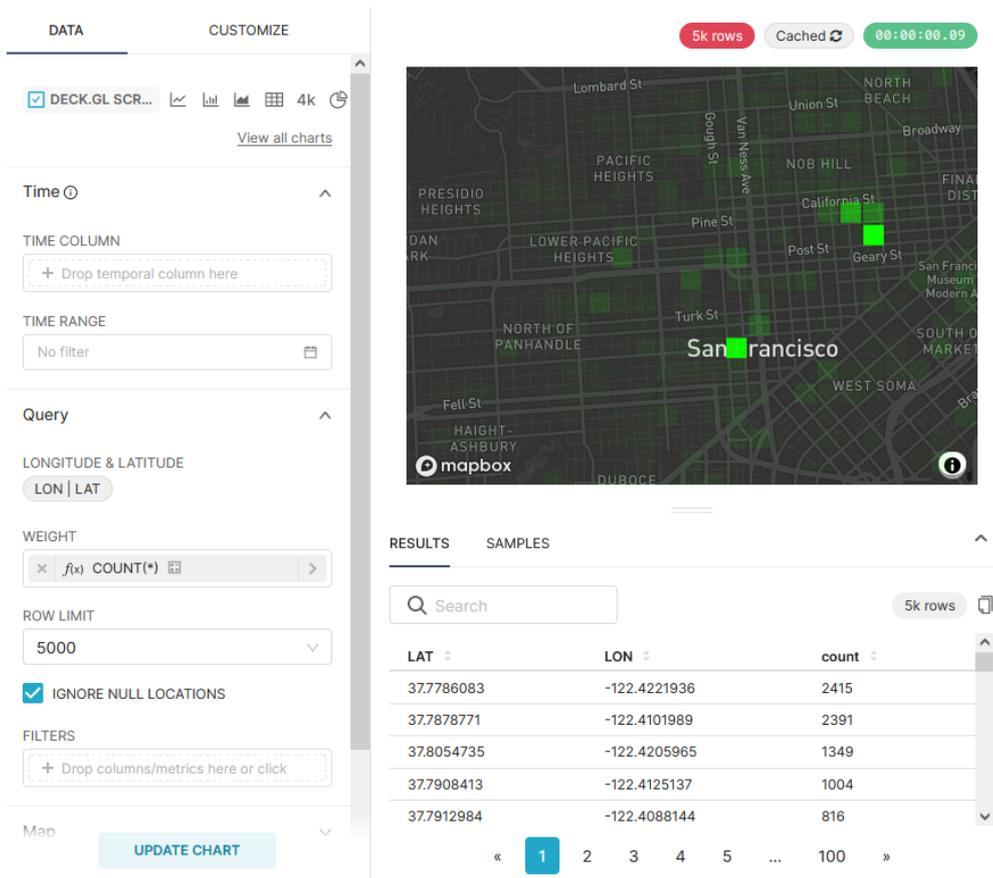


Abbildung 13. Visualization Type Deck.gl Screen Grid, rechts am Beispiel Punkte in San Francisco, Kalifornien, links der dazugehörige Dialog

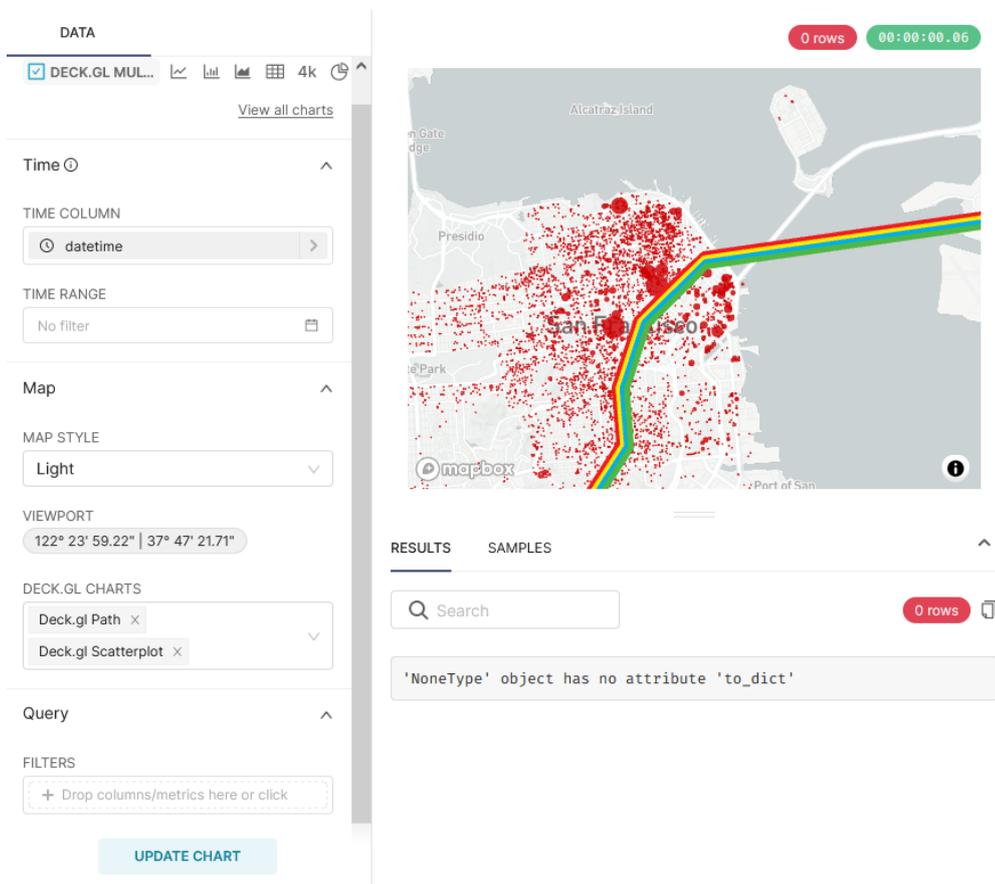


Abbildung 14. Visualization Type Deck.gl Multiple Layers, rechts am Beispiel die obigen Charts deck.gl Scatterplot und deck.gl Path zusammengeführt, links der dazugehörige Dialog

Noch Fragen? Siehe "Kontakt" auf OpenSchoolMaps!



Frei verwendbar unter [CC0 1.0](#)