Datos Abiertos en tiempos modernos

Problemas, soluciones, y algunas ideas.

@davidgasquez

@davidgasquez

kaggle

Search

Sign In

Register

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

View Active Events

Competitions

Grow your data science skills by competing in our exciting competitions. Find help in the documentation or learn about Community Competitions.

Host a Competition

Search competitions

Filters

All Competitions
Everything, past & present

Featured
Premier challenges with prizes

Getting Started
Approachable ML fundamentals

Research
Scientific and scholarly challenges

Community
Created by fellow Kagglers

Playground
Fun practice problems

Get Started

See all

New to Kaggle?

These competitions are perfect for newcomers.

Titanic - Machine Learning from Disaster

Start here! Predict survival on the Titani...

Getting Started

13633 Teams

Knowledge Ongoing

House Prices - Advanced Regression Techniques

Predict sales prices and practice feature...

Getting Started

4399 Teams

Knowledge Ongoing

Spaceship Titanic

Predict which passengers are transporte...

Getting Started

1936 Teams

Knowledge Ongoing

train.csv (61.19 kB)

Detail | Compact | Column | 11 of 12 columns

About this file

contains data

# Survived	# Pclass	Name	Sex	# Age	# SibSp
0 1	1 3	891 unique values	male 65% female 35%	0.42 80	0 0.00 - 0.80 Count: 608
0	3	Braund, Mr. Owen Harris	male	22	1
1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1
1	3	Heikkinen, Miss. Laina	female	26	0
1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1
0	3	Allen, Mr. William Henry	male	35	0
0	3	Moran, Mr. James	male		0
0	1	McCarthy, Mr. Timothy J	male	54	0
0	3	Palsson, Master. Gosta Leonard	male	2	3
1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0
1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1
1	3	Sandstrom, Miss. Marguerite Rut	female	4	1
1	1	Bonnell, Miss. Elizabeth	female	58	0
0	3	Saundercock, Mr.	male	20	0

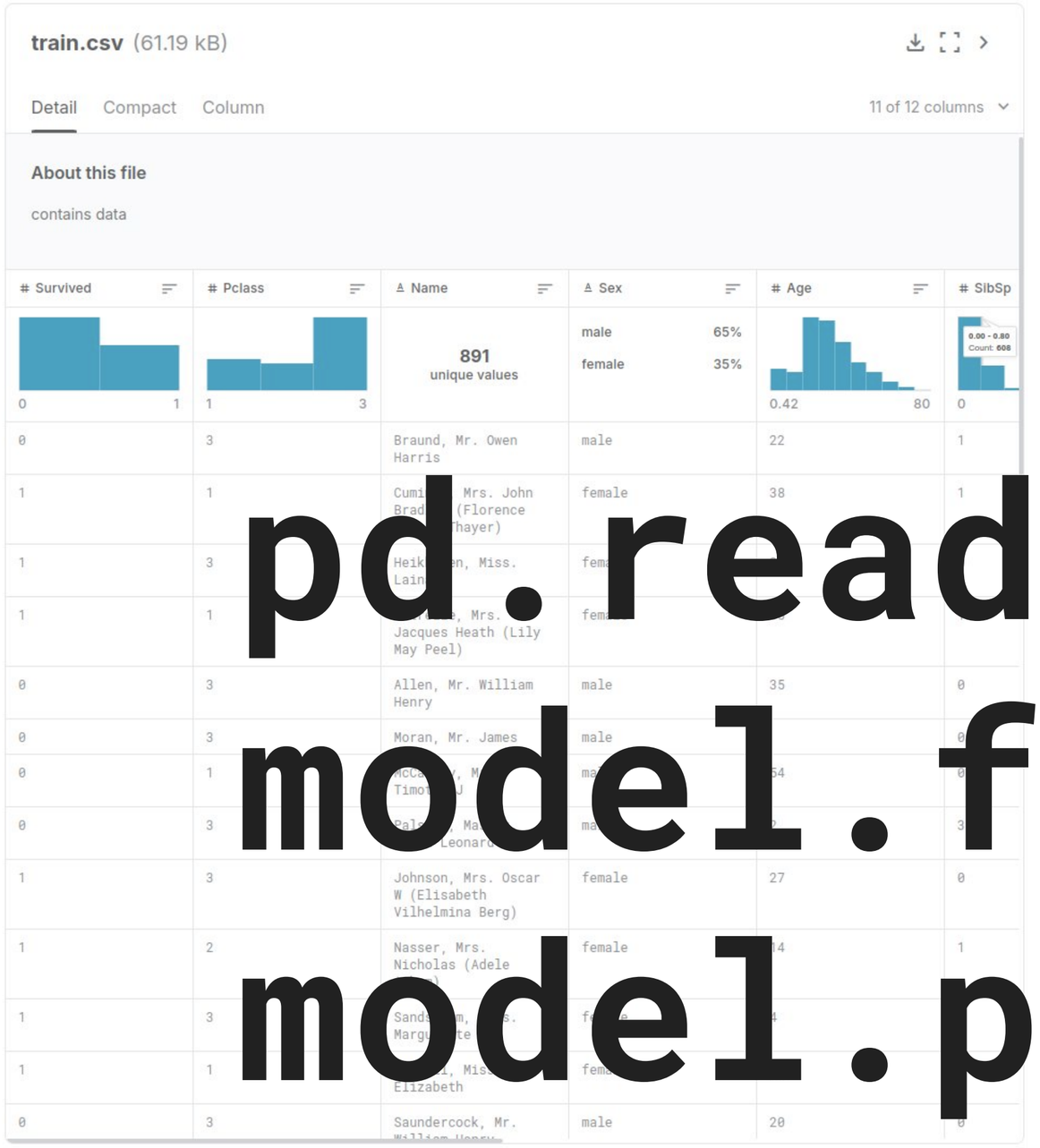
test.csv (28.63 kB)

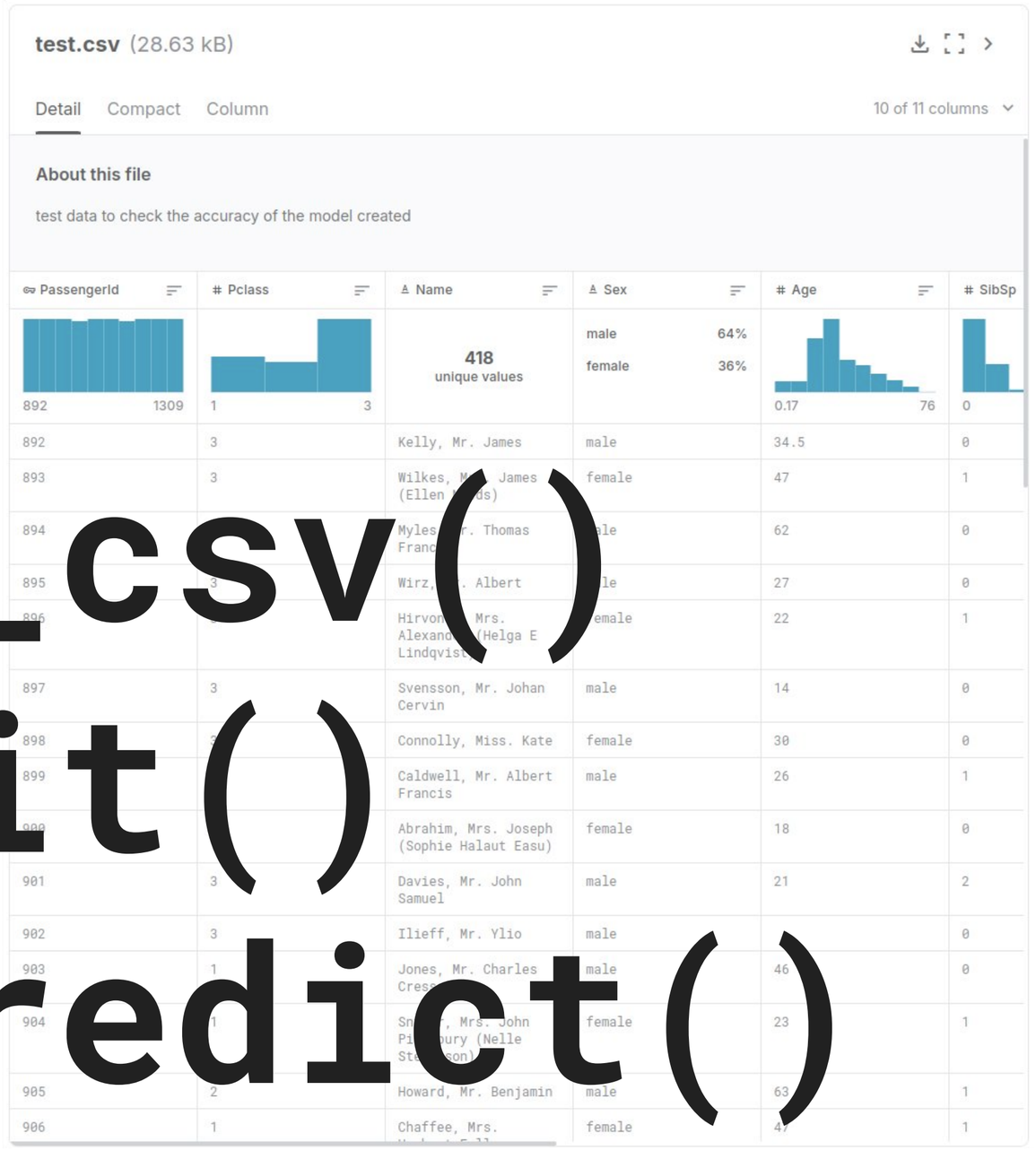
Detail | Compact | Column | 10 of 11 columns

About this file

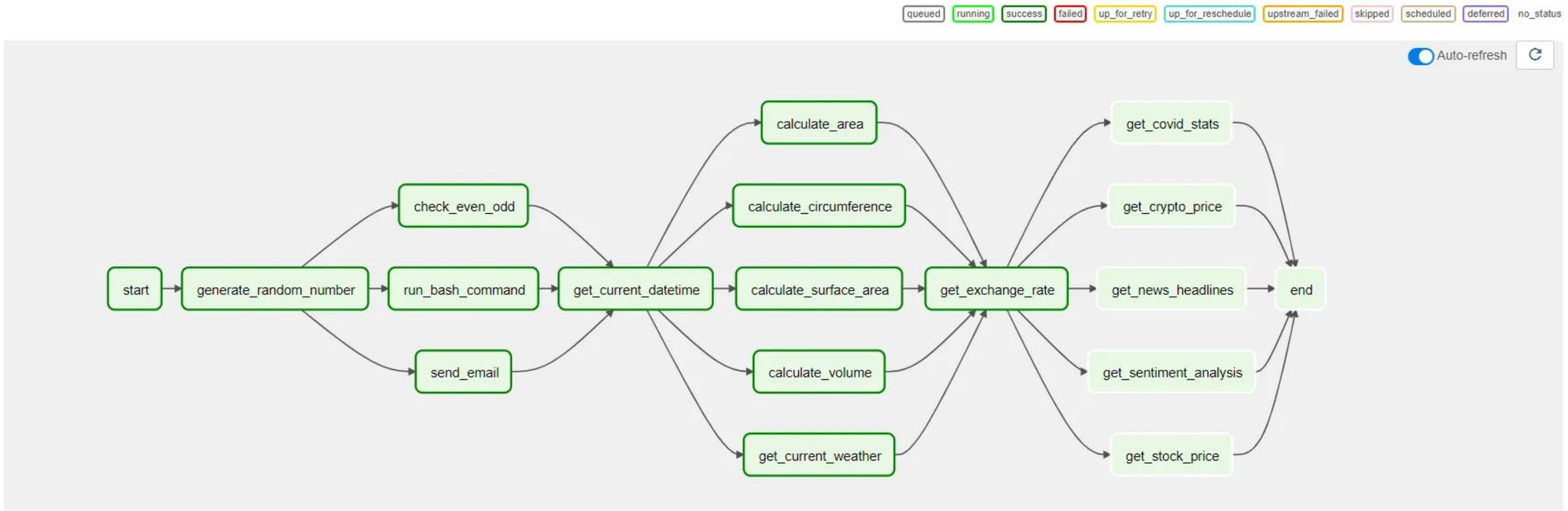
test data to check the accuracy of the model created

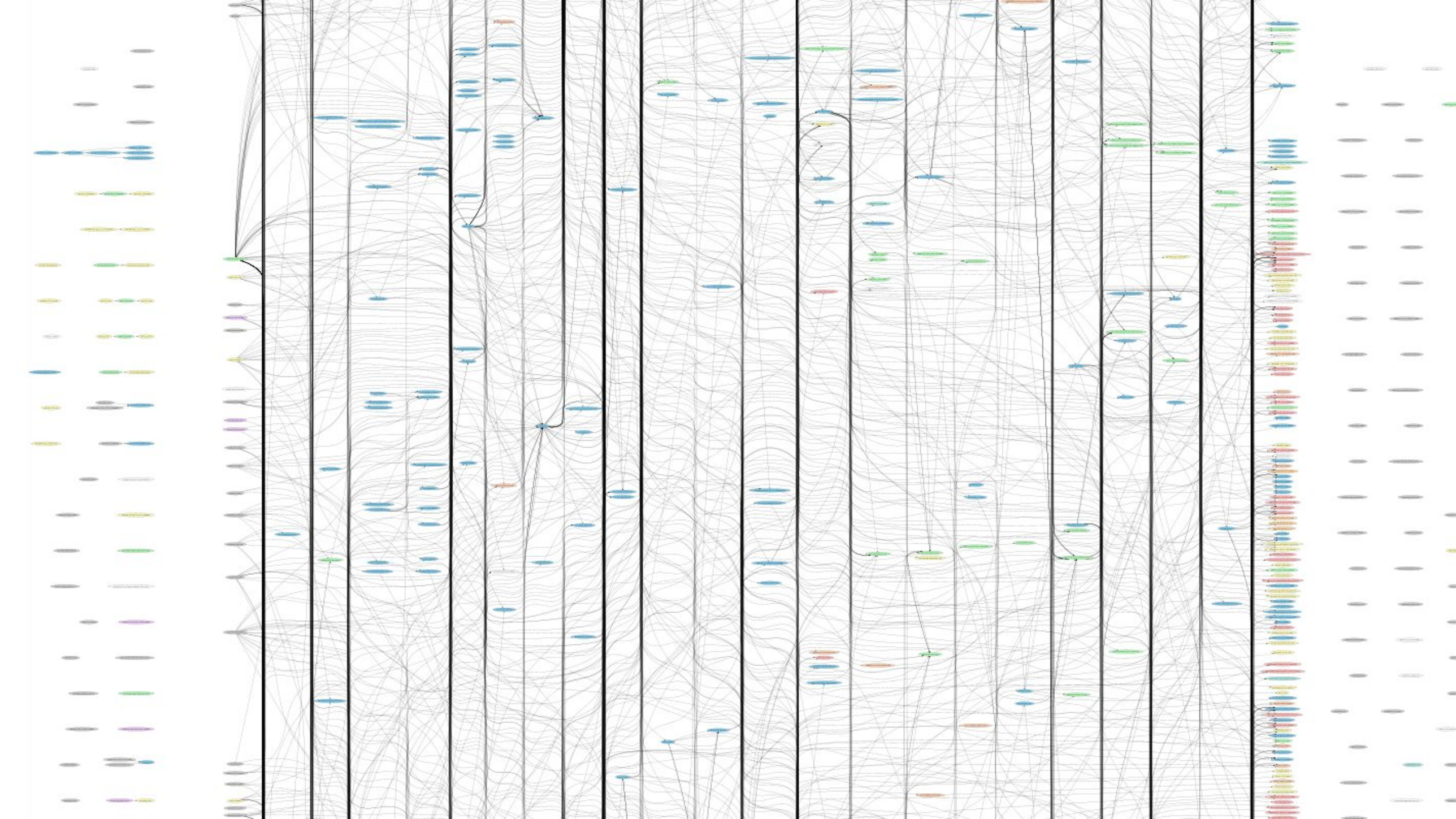
PassengerId	# Pclass	Name	Sex	# Age	# SibSp
892 1309	1 3	418 unique values	male 64% female 36%	0.17 76	0
892	3	Kelly, Mr. James	male	34.5	0
893	3	Wilkes, Mrs. James (Ellen Needs)	female	47	1
894	2	Myles, Mr. Thomas Francis	male	62	0
895	3	Wirz, Mr. Albert	male	27	0
896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22	1
897	3	Svensson, Mr. Johan Cervin	male	14	0
898	3	Connolly, Miss. Kate	female	30	0
899	2	Caldwell, Mr. Albert Francis	male	26	1
900	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	female	18	0
901	3	Davies, Mr. John Samuel	male	21	2
902	3	Ilieff, Mr. Ylio	male		0
903	1	Jones, Mr. Charles Cresson	male	46	0
904	1	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23	1
905	2	Howard, Mr. Benjamin	male	63	1
906	1	Chaffee, Mrs.	female	47	1

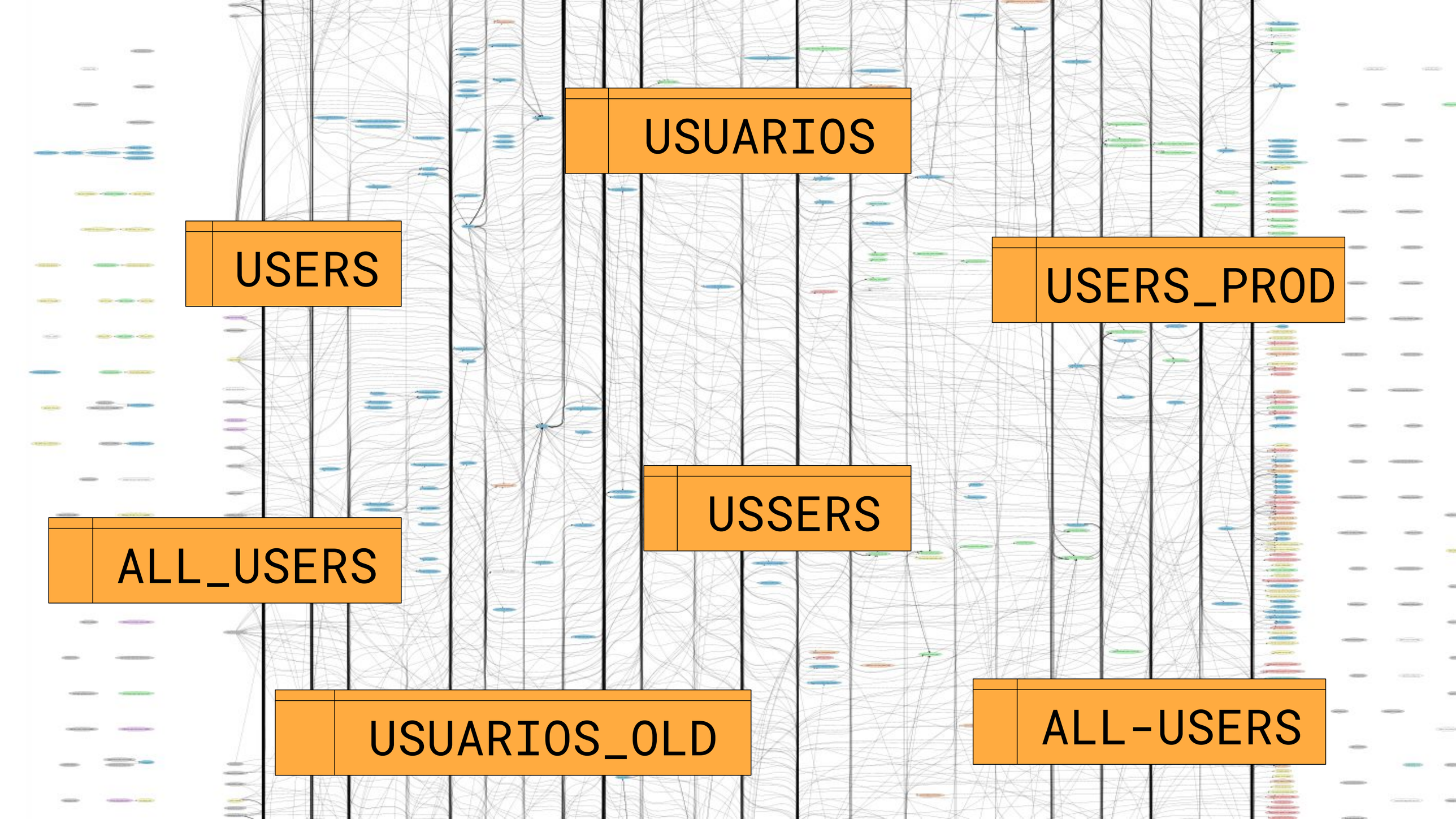




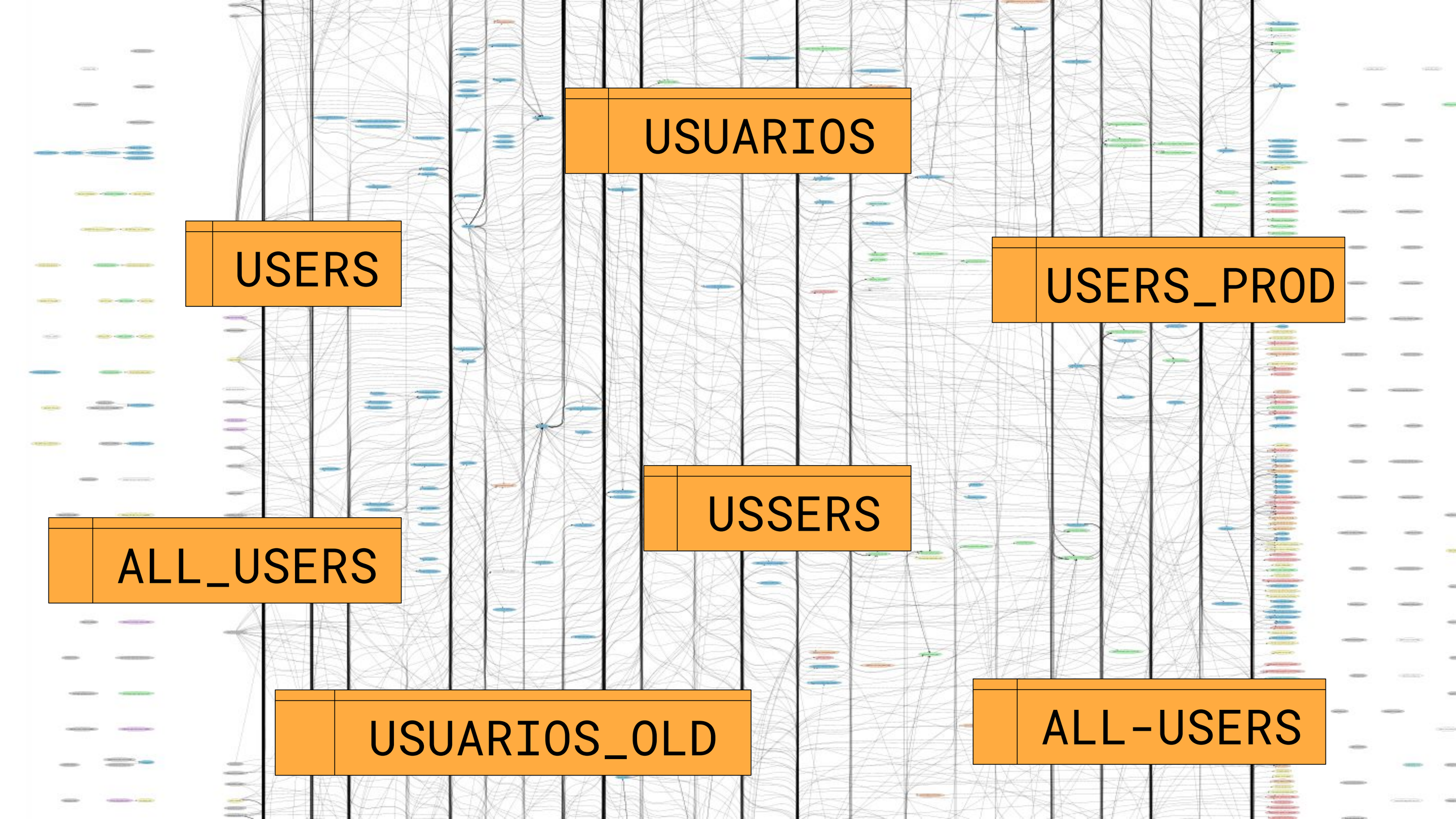
pd.read_csv()
model.fit()
model.predict()


queued
running
success
failed
up_for_retry
up_for_reschedule
upstream_failed
skipped
scheduled
deferred
no_status
Auto-refresh
start
generate_random_number
check_even_odd
run_bash_command
send_email
get_current_datetime
calculate_area
calculate_circumference
calculate_surface_area
calculate_volume
get_current_weather
get_exchange_rate
get_covid_stats
get_crypto_price
get_news_headlines
get_sentiment_analysis
get_stock_price
end




USUARIOS
USERS
USERS_PROD
USSERS
ALL_USERS
USUARIOS_OLD
ALL-USERS


DATA
SCIENCE
ME


USUARIOS
USERS
USERS_PROD
USSERS
ALL_USERS
USUARIOS_OLD
ALL-USERS

API JSON del INE

El servicio API JSON INE *(Java Script Object Notation)* que se describe en esta sección permite acceder mediante peticiones URL a toda la información disponible en INEbase, sistema que utiliza el Instituto Nacional de Estadística (INE) para la publicación de la información estadística.

La estructura de las peticiones a través de URL y la simplicidad del formato JSON, hacen que este tipo de aplicaciones sean ampliamente utilizadas para ofrecer datos y metadatos que permiten la explotación automática de la información estadística.

INEbase es el sistema que utiliza el INE para la publicación de la información estadística. La información que será accesible a través del servicio API JSON del INE que se describe en esta sección, proviene de dos fuentes distintas:

- **Base de datos de difusión (Tempus3)**.

1. Compartir herramientas para trabajar con datos interesantes.

2. Enseñar ejemplos de portales de datos abiertos modernos.

*

1. Open Data es un ecosistema gigante y diverso.
2. Es más “people problem” que “technical problem”.

more Open Data can help make

better decisions

2,549 hours

wasted

finding parking

7,000 lives

saved due to quicker response

5.5% less

road fatalities

629 million

hours saved is equivalent to

€ 27.9 bn

Congestion

costs are 1% of GDP

16% less

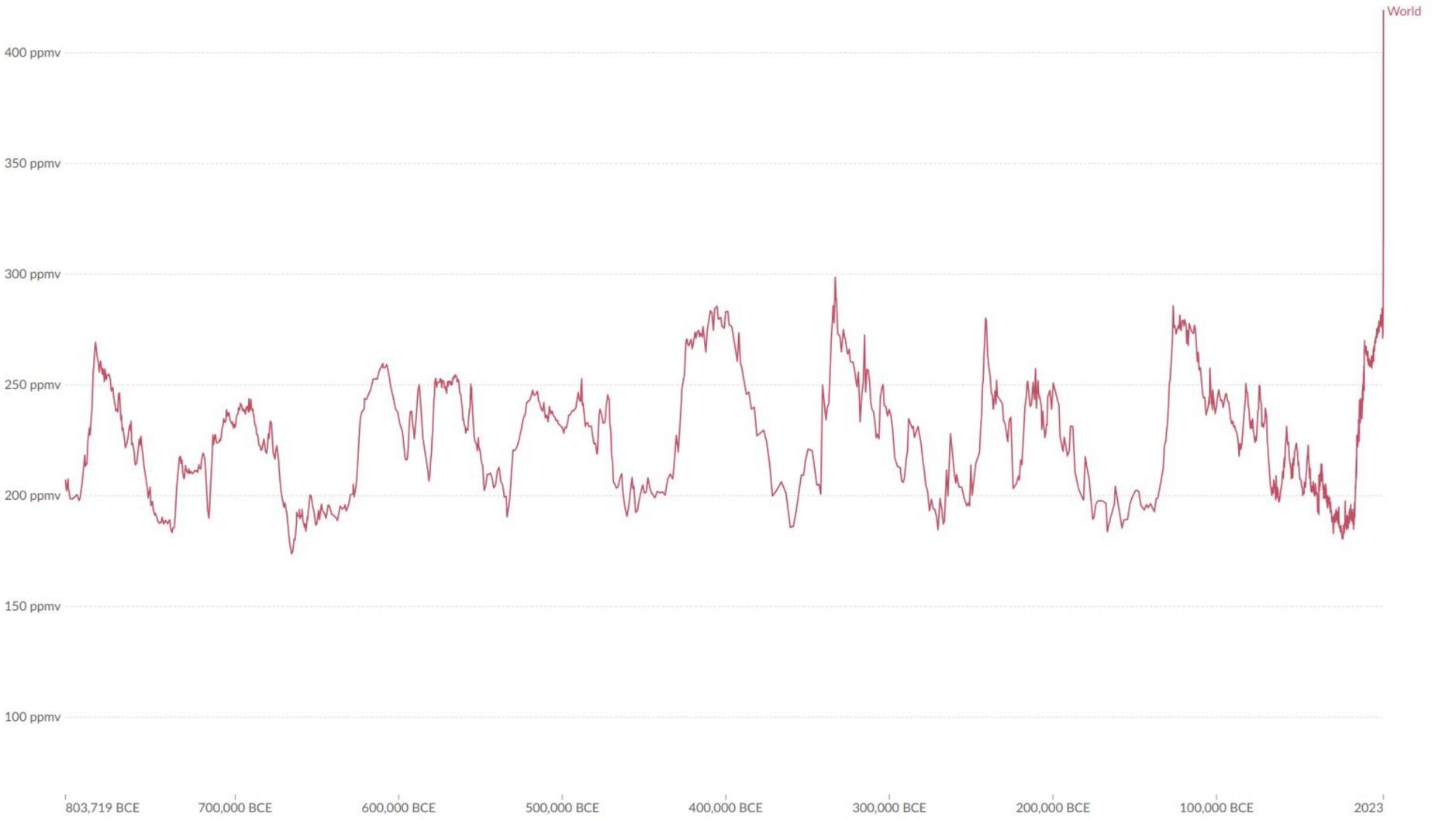
less energy used

Carbon dioxide concentrations in the atmosphere

Atmospheric carbon dioxide (CO_2) concentration is measured in parts per million (ppm). Long-term trends in CO_2 concentrations can be measured at high-resolution using preserved air samples from ice cores.

Our World in Data

Table | Chart



Play time-lapse | 803,719 BCE | 2023

Data source: NOAA Global Monitoring Laboratory - Trends in Atmospheric Carbon Dioxide (2024); EPA based on various sources (2022) – Learn more about this data

OurWorldinData.org/climate-change | CC BY

Download | Share | Exit full-screen

Toyota

KM 0

100000

Price 0

50000

Coches

Fuel Type

☐ Diésel ☑ Gasolina ☑ Híbrido ☑ Híbrido enchufable

☑ Eléctrico ☑ null ☑ Gas natural (CNG)

☑ Gas licuado (GLP)

Offer Type

☑ Ocasión ☑ Nuevo ☑ Km0 ☑ Demostración

☑ Gerencia ☑ Sustitución

Main Province

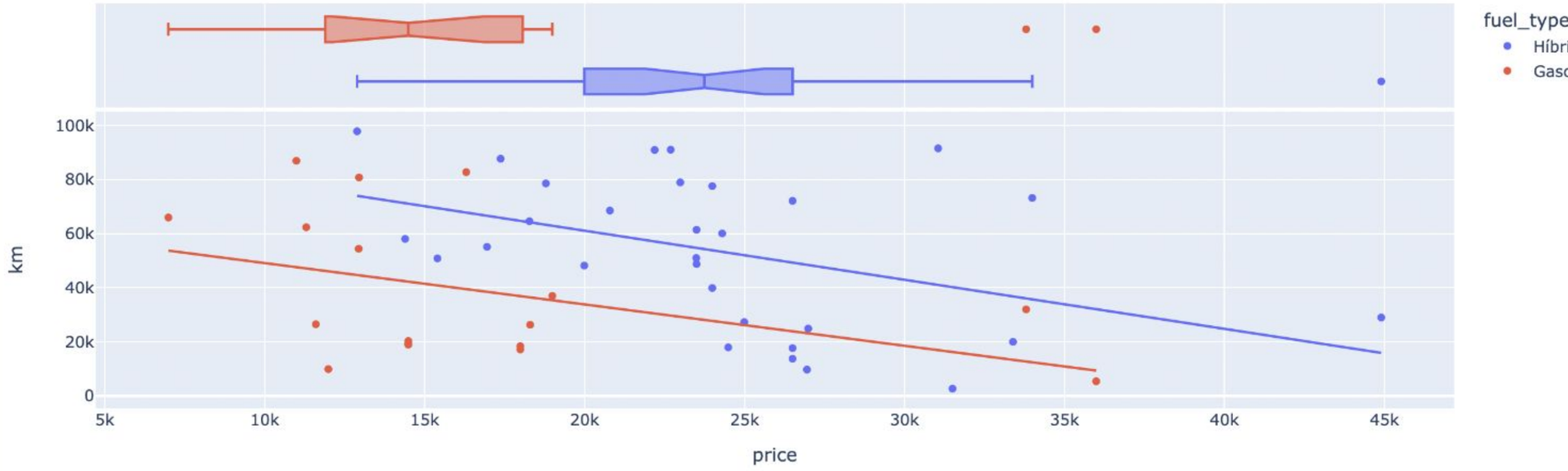
× Madrid × Toledo × Guadalajara

Search

X: Precio

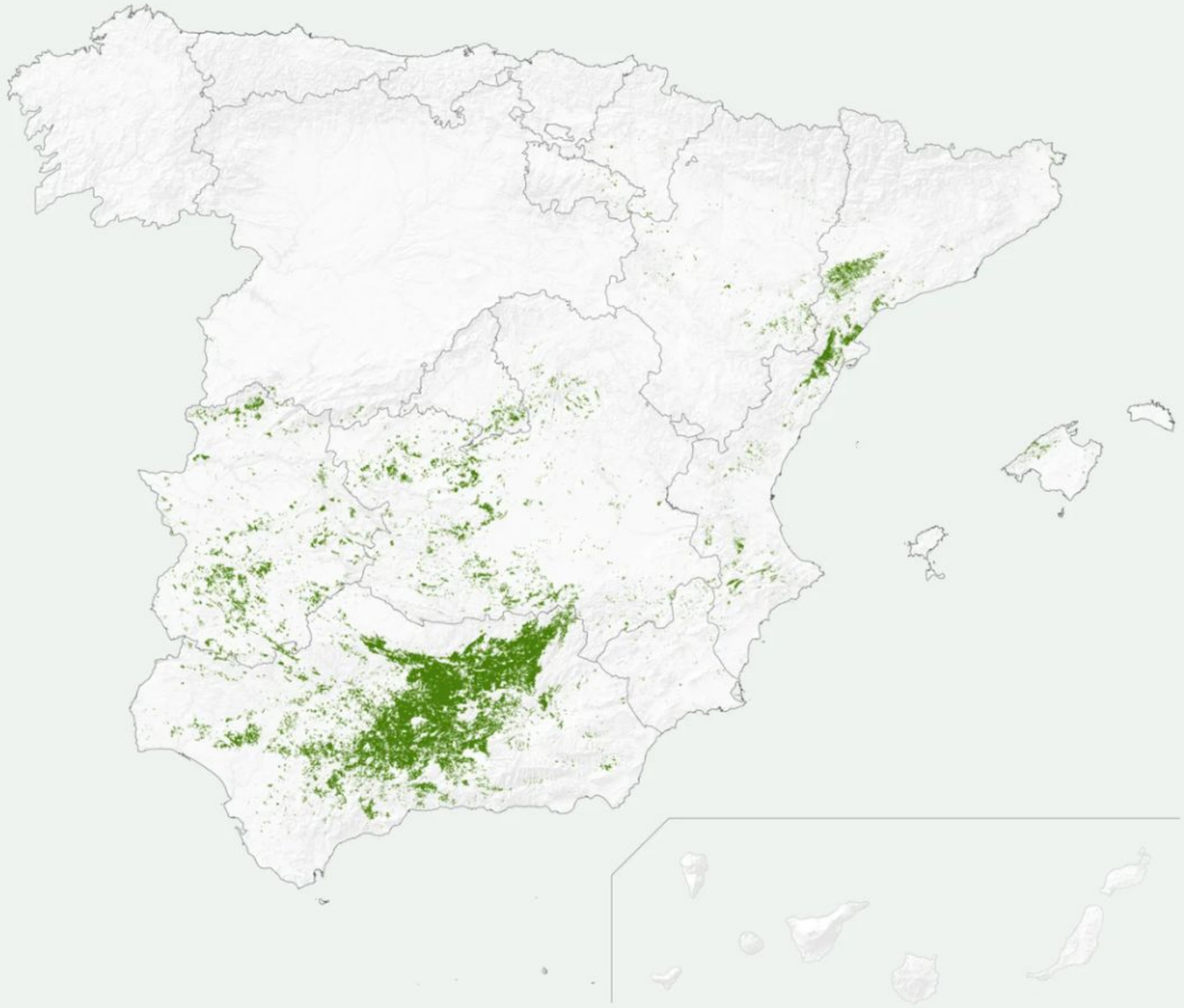
Y: Km

Color: Combustible



	id	title	km	year	main_province	fuel_type	p
	filter data.						
☐	53652920	TOYOTA CHR 1.8 125H Advance	51000	2019	Madrid	Híbrido	2
☐	53510998	TOYOTA Rav4 2.5l hybrid 2WD Advance Pack Drive	72171	2018	Madrid	Híbrido	2
☐	53652392	TOYOTA Aygo 1.0 70 xplay 5p.	19277	2021	Madrid	Gasolina	1
☐	53546077	TOYOTA Corolla 2.0 180H GRSPORT ECVT 5p.	2675	2022	Madrid	Híbrido	3
☐	53338548	TOYOTA Rav4 2.5l 220H Advance 5p.	73237	2019	Madrid	Híbrido	3
☐	53492945	TOYOTA CHR 1.8 125H Advance 5p.	48783	2019	Madrid	Híbrido	2
☐	53652394	TOYOTA Aygo 1.0 70 xplay 5p.	20294	2021	Madrid	Gasolina	1
☐	53652393	TOYOTA Aygo 1.0 70 xplay 5p.	18928	2021	Madrid	Gasolina	1
☐	53651030	TOYOTA Corolla 1.8 125H ACTIVE TECH ECVT 5p.	48216	2019	Madrid	Híbrido	1
☐	53651026	TOYOTA CHR 1.8 125H Advance 5p.	90997	2021	Madrid	Híbrido	2

« ‹ 1 / 5 ›

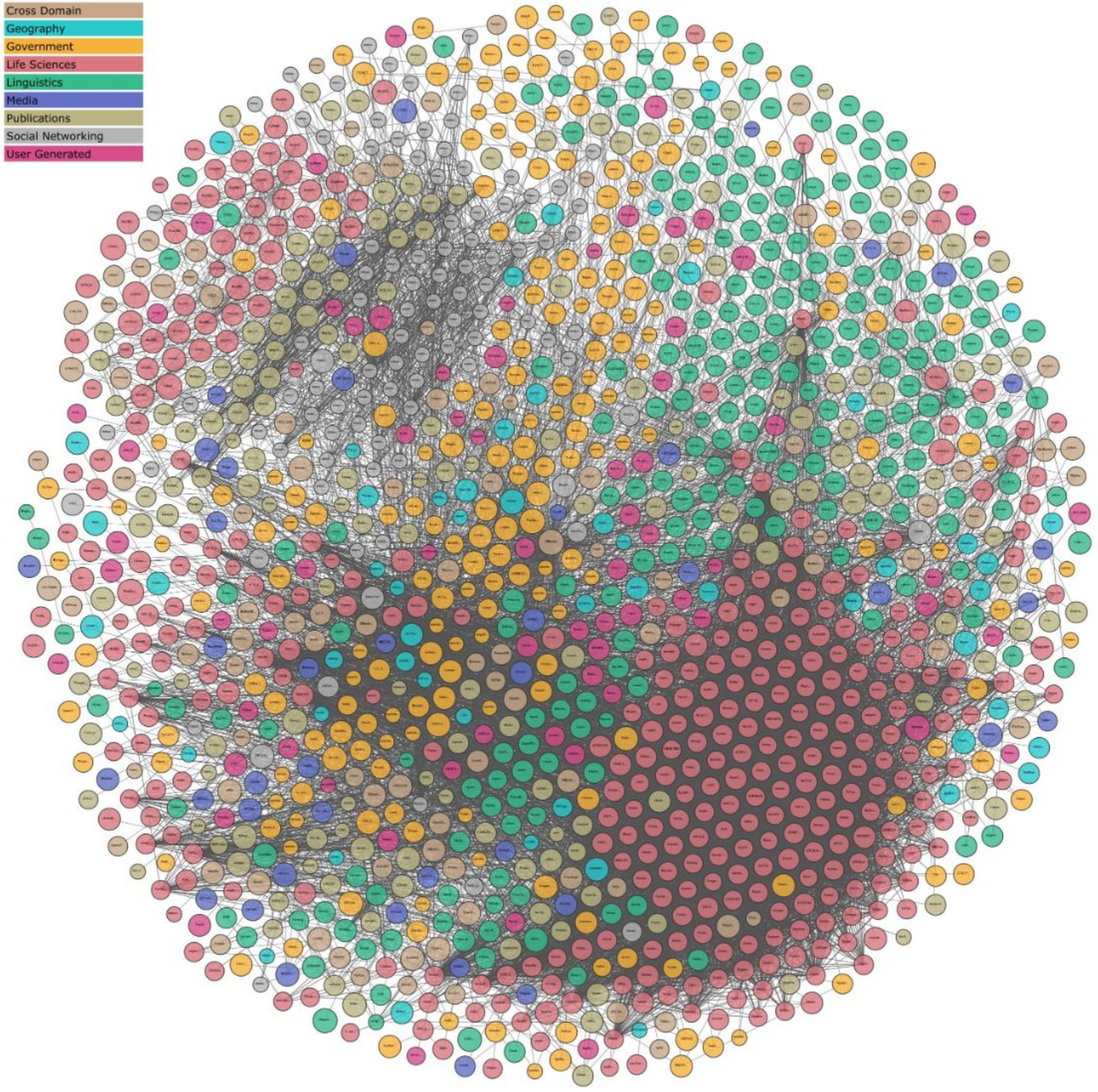


Fuente: El Orden Mundial

¡El mundo está lleno de datos abiertos!

Oh, oh... **Problemas**?

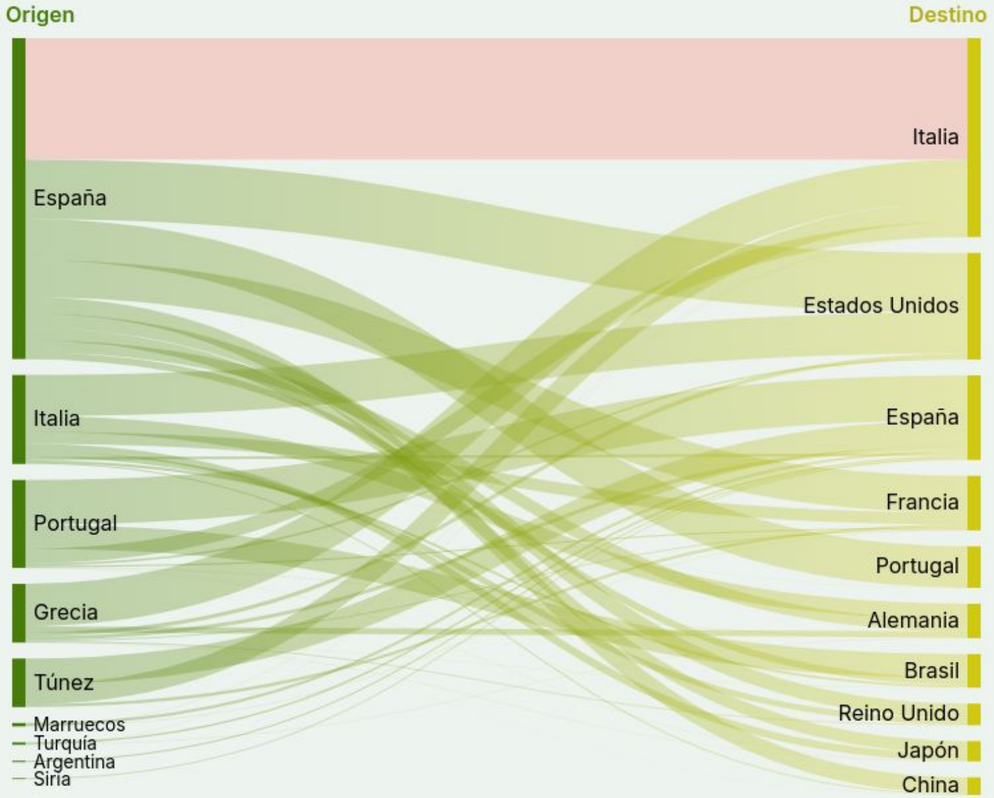
¡El mundo está lleno de datos abiertos!

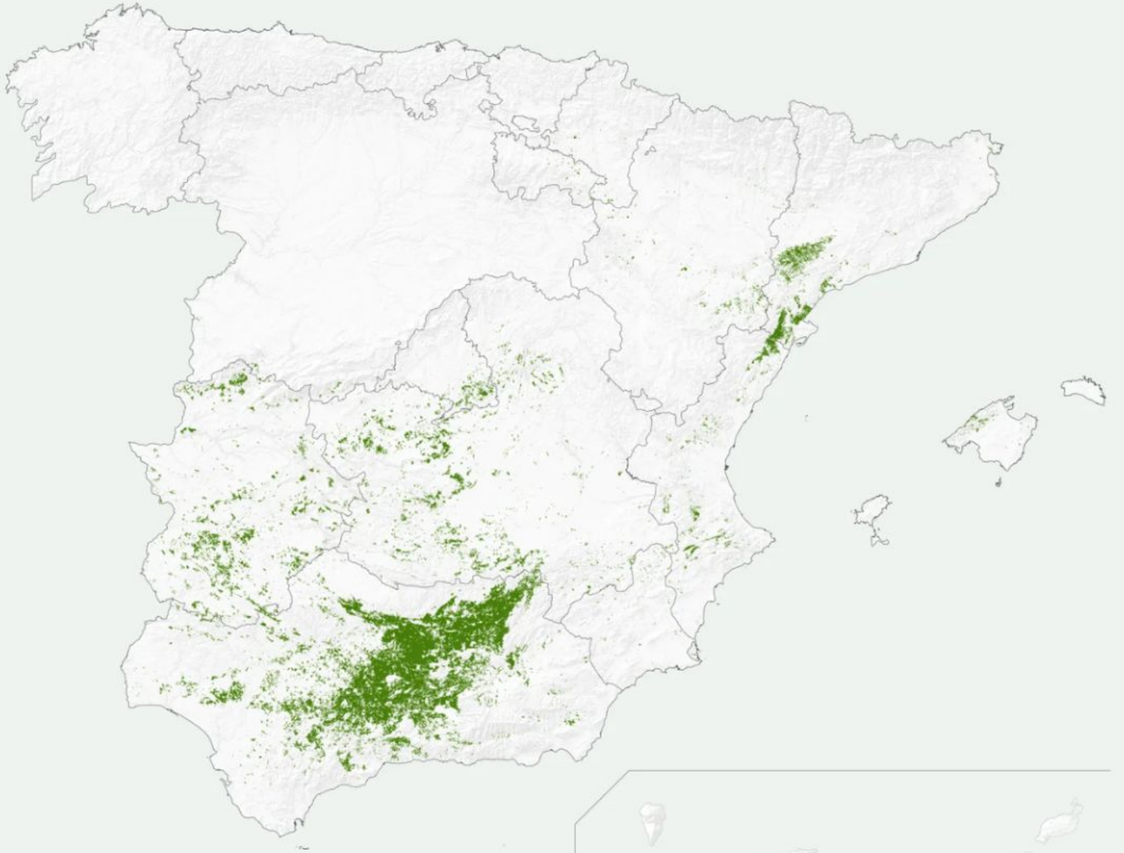

Legend
Cross Domain
Geography
Government
Life Sciences
Linguistics
Media
Publications
Social Networking
User Generated

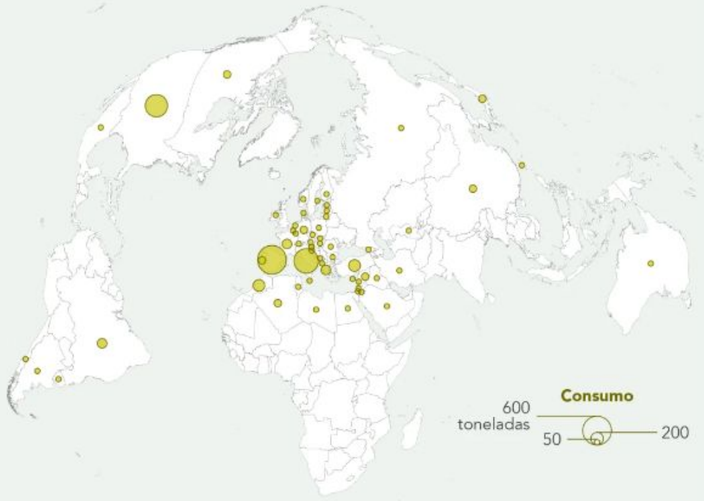
```python
import os
import pandas as pd
from glob import glob

def merge_files(file_pattern):
    all_files = glob(f"../download_epfs_since2016/parquet_raw_files/**/{file_pattern}", recursive=True)
    print(f"Found {len(all_files)} files matching pattern: {file_pattern}")
    df_list = [pd.read_parquet(file) for file in all_files]
    merged_df = pd.concat(df_list, ignore_index=True)
    print(f"Merged dataframe shape: {merged_df.shape}")
    return merged_df

def main():
    print("Starting data merging process...")

    # Merge gastos files
    print("Merging gastos files...")
    gastos_df = merge_files("*_EPFgastos_*.parquet")

    # Merge hogar files
    print("Merging hogar files...")
    hogar_df = merge_files("*_EPFhogar_*.parquet")

    # Merge mhogar files
    print("Merging mhogar files...")
    mhogar_df = merge_files("*_EPFmhogar_*.parquet")

    print("Joining gastos and hogar dataframes...")
    # Join gastos and hogar dataframes
    columns_to_keep = [
        'CCAA', 'NUTS1', 'DENSIDAD', 'NMIEMB', 'TAMANO', 'TIPHOGAR1', 'EDADSP',
        'SEXOSP', 'PAISNACSP', 'NACIONASP', 'ECIVILLEGALSP', 'UNIONSP',
        'CONVIVENCIASP', 'ESTUDIOSSP', 'SITUACTSP', 'IMPEXACPSP', 'INTERINPSP',
        'OCUPA', 'OCUPARED', 'ACTESTB', 'ACTESTBRED', 'SITPROF', 'SECTOR',
        'CONTRATO', 'TIPOCONT', 'SITSOCI', 'SITSOCIRE', 'REGTEN', 'TIPOEDIF',
        'ZONARES', 'TIPOCASA', 'NHABIT', 'ANNOCON', 'SUPERF', 'AGUACALI',
        'FUENAGUA', 'CALEF', 'IMPEXAC', 'INTERIN', 'COMIMH'
    ]

    merged_df = pd.merge(
        gastos_df,
        hogar_df[['ANOENC', 'NUMERO'] + columns_to_keep],
        on=['ANOENC', 'NUMERO'],
        how='left'
    )

    print("Converting 'COMIMH' to float...")
    merged_df['COMIMH'] = pd.to_numeric(merged_df['COMIMH'], errors='coerce')

    print("Checking for other columns that might need conversion...")
    for col in merged_df.columns:
        if merged_df[col].dtype == 'object':
            try:
                merged_df[col] = pd.to_numeric(merged_df[col], errors='raise')
                print(f"Successfully converted column {col} to numeric.")
            except ValueError:
                print(f"Column {col} contains non-numeric data and will be left as is.")

    # Save the merged dataframe
    output_path = "../gastos_master.parquet"
    print(f"Saving merged data to {output_path}...")
    merged_df.to_parquet(output_path, index=False)
    print(f"Merged data saved successfully. Final dataframe shape: {merged_df.shape}")

if __name__ == "__main__":
    main()
```

```python
import os
import requests
import zipfile
import shutil
from tqdm import tqdm

BASE_URL = "https://www.ine.es/ftp/microdatos/epf2006/datos_{}.zip"
SPSS_DIR = "SPSS_raw_files"

def download_file(url, filename):
    response = requests.get(url, stream=True)
    total_size = int(response.headers.get('content-length', 0))

    with open(filename, 'wb') as file, tqdm(
        desc=filename,
        total=total_size,
        unit='iB',
        unit_scale=True,
        unit_divisor=1024,
    ) as progress_bar:
        for data in response.iter_content(chunk_size=1024):
            size = file.write(data)
            progress_bar.update(size)

def process_year(year):
    url = BASE_URL.format(year)
    zip_filename = f"datos_{year}.zip"

    # Download the zip file
    print(f"Downloading data for {year}...")
    download_file(url, zip_filename)

    # Unzip the main zip file
    print(f"Unzipping data for {year}...")
    with zipfile.ZipFile(zip_filename, 'r') as zip_ref:
        zip_ref.extractall(f"temp_{year}")

    # Process the extracted files
    for root, dirs, files in os.walk(f"temp_{year}"):
        for file in files:
            if file.endswith(".zip"):
                zip_path = os.path.join(root, file)
                with zipfile.ZipFile(zip_path, 'r') as zip_ref:
                    zip_ref.extractall(os.path.dirname(zip_path))

    # Move SPSS files to the output directory
    for root, dirs, files in os.walk(f"temp_{year}"):
        if "SPSS" in dirs:
            spss_dir = os.path.join(root, "SPSS")
            for file in os.listdir(spss_dir):
                if file.endswith(".sav"):
                    src = os.path.join(spss_dir, file)
                    dst = os.path.join(SPSS_DIR, f"{year}_{file}")
                    shutil.move(src, dst)

    # Clean up temporary files
    shutil.rmtree(f"temp_{year}")
    os.remove(zip_filename)

def main():
    os.makedirs(SPSS_DIR, exist_ok=True)

    for year in range(2016, 2023):
        process_year(year)

    print("Processing complete.")

if __name__ == "__main__":
    main()
```

```python
import pandas as pd
import numpy as np
import os

# Read the CSV file
df = pd.read_csv('filtered_barometros_files.csv')

# Create a dictionary to map Spanish month names to numbers
month_map = {
    'Enero': '01', 'Febrero': '02', 'Marzo': '03', 'Abril': '04',
    'Mayo': '05', 'Junio': '06', 'Julio': '07', 'Agosto': '08',
    'Septiembre': '09', 'Octubre': '10', 'Noviembre': '11', 'Diciembre': '12'
}

# Create a new column with the numeric month equivalent
df['Mes_numero'] = df['Mes de realización'].map(month_map)

# Create the date_of_study column
df['date_of_study'] = df['Mes_numero'] + '/' + df['Año de realización'].astype(str) + '/01'

# Convert date_of_study to datetime
df['date_of_study'] = pd.to_datetime(df['date_of_study'], format='%m/%Y/%d', errors='coerce')

# Create 'Principales Problemas' as a list of non-null problems
df['Principales Problemas'] = df.apply(lambda row: [prob for prob in [row['Primer problema'], row['Segundo problema'], row['T

# Define the groups and their variables
groups = {
    "Estudio": [
        "Estudio", "Código del estudio", "Año de realización",
        "Mes de realización", "Número de registro"
    ],
    "Demografía": [
        "Sexo de la persona entrevistada", "Edad de la persona entrevistada",
        "Religiosidad de la persona entrevistada", "Frecuencia de asistencia a oficios religiosos",
        "Estado civil de la persona entrevistada", "Estudios de la persona entrevistada", "Nivel de estudios alcanzado por la
        "Escolarización de la persona entrevistada",
        "Población activa e inactiva", "Situación laboral de la persona entrevistada",
        "Ocupación de la persona entrevistada",
        "Clase social subjetiva de la persona entrevistada", "Identificación subjetiva de clase",
        "Nivel de ingresos netos del hogar", "Escala de autoubicación ideológica (1-10)", "Nacionalidad de la persona entrevi
        "Provincia", "Municipio", "Comunidad autónoma", "Tamaño de municipio"
    ],
    "Opiniones": [
        "Principales Problemas",
        "Primer problema", "Segundo problema", "Tercer problema",
        "Valoración de la situación económica personal actual",
        "Valoración de la situación económica general de España"
    ],
    "Voto": [
        "Intención de voto en unas supuestas elecciones generales",
        "Intención de voto en unas supuestas elecciones generales [recodificada]",
        "Recuerdo de voto en las elecciones generales de 2023",
        "Recuerdo de voto en las elecciones generales de 2023 de los votantes [recodificada]",
        "Partido político que considera más cercano a sus ideas",
        "Preferencia personal como presidente del Gobierno central",
        "Intención de voto alternativo en supuestas elecciones generales",
        "Intención de voto alternativo en supuestas elecciones generales [recodificada]",
    ]
}

# Function to check if a column exists and has null values
def check_column(column):
    return column in df.columns and df[column].isnull().any()
```

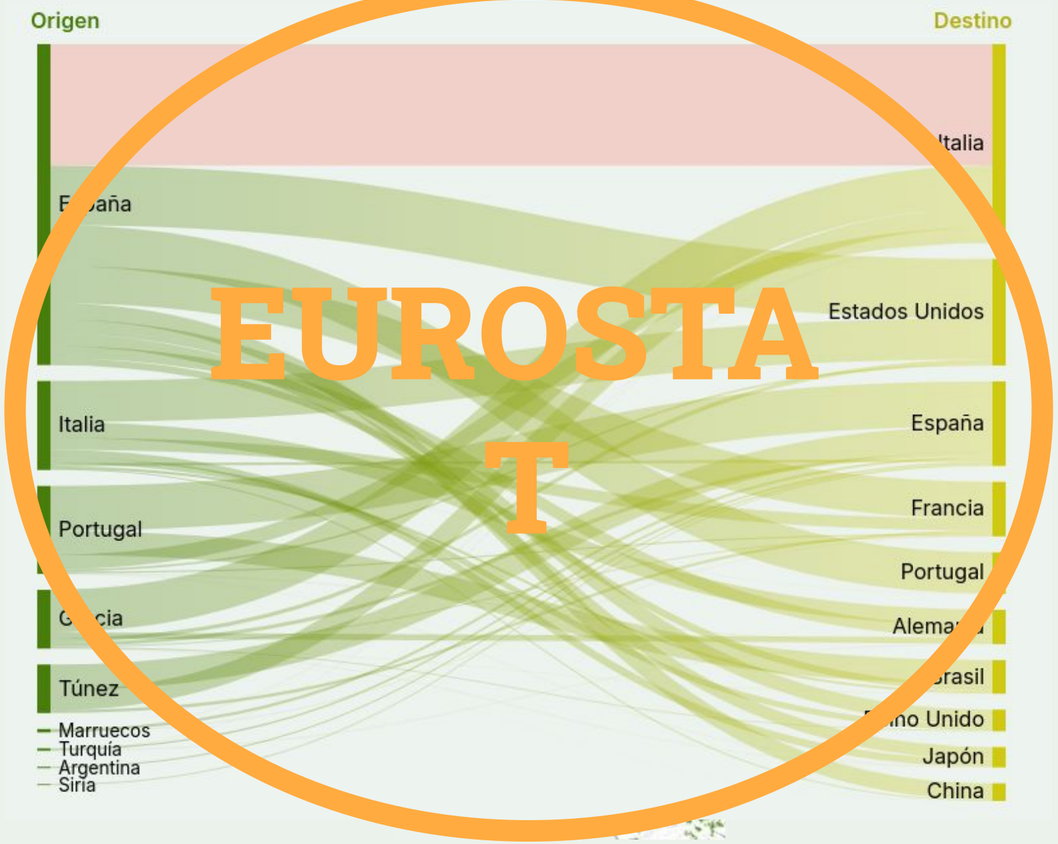






Fuente: El Orden Mundial

EUROSTAT



Corine Land Cover (2018)

Consejo Oleícola Internacional

Fuente: El Orden Mundial

Obtener los nombres de los diez organismos que más conjuntos de datos tienen publicados y visualizar el número de éstos

Para realizar esta consulta vamos a tener que agrupar resultados, ordenarlos y limitar el total a 10.

```
select distinct ?label count(?x) as ?num {
?x a <http://www.w3.org/ns/dcat#Dataset> .
?x <http://purl.org/dc/terms/publisher> ?publicador.
?publicador <http://www.w3.org/2004/02/skos/core#prefLabel> ?label.
}
group by (?label)
order by desc(?num)
limit 10
```

El resultado es este:

label	num
"Gobierno de Aragón"	2659
"Comunidad Autónoma de País Vasco"	2208
"Centro de Investigaciones Sociológicas"	2107
"Ayuntamiento de Málaga"	651
"Ayuntamiento de Gijón"	627
"Xunta de Galicia"	315
"Generalitat Valenciana"	313
"Ayuntamiento de Madrid"	231
"Instituto Nacional de Estadística"	205
"Junta de Castilla y León"	196

Obtener los nombres de los diez organismos que más conjuntos de datos tienen publicados y visualizar el número de éstos

Para realizar esta consulta vamos a tener que agrupar resultados, ordenarlos y limitar el total a 10.

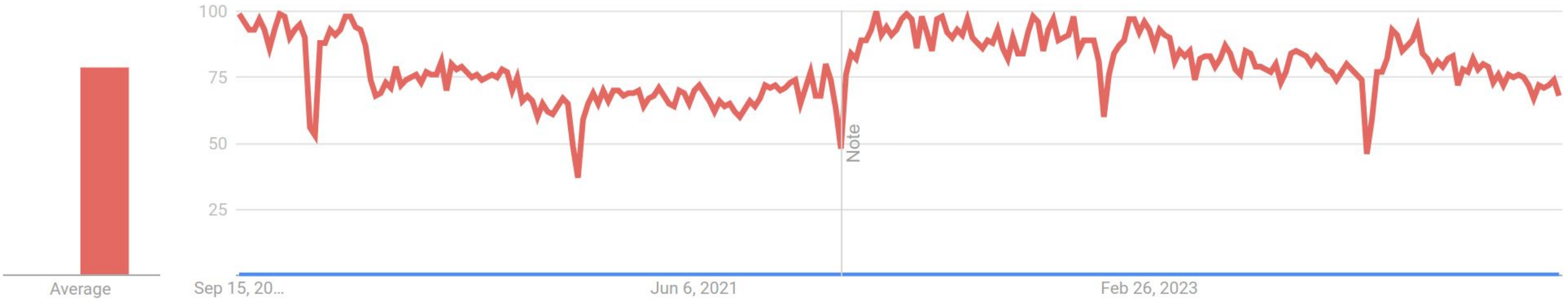
```
select distinct ?label count(?x) as ?num {
?x a <http://www.w3.org/ns/dcat#Dataset> .
?x <http://purl.org/dc/terms/publisher> ?publicador.
?publicador <http://www.w3.org/2004/02/skos/core#prefLabel> ?label.
}
group by (?label)
order by desc(?num)
limit 10
```

SPARQL

El resultado es este:

label	num
"Gobierno de Aragón"	2659
"Comunidad Autónoma de País Vasco"	2208
"Centro de Investigaciones Sociológicas"	2107
"Ayuntamiento de Málaga"	651
"Ayuntamiento de Gijón"	627
"Xunta de Galicia"	315
"Generalitat Valenciana"	313
"Ayuntamiento de Madrid"	231
"Instituto Nacional de Estadística"	205
"Junta de Castilla y León"	196

Interest over time



Wikidata Query Service | Examples | Help | More tools | Query Builder | English

```
SELECT
?asteroidLabel
?discovered
?discovererLabel
WHERE {
  ?asteroid wdt:P31 wd:Q3863;  # Retrieve instances of "asteroid"
            wdt:P61 ?discoverer; # Retrieve discoverer of the asteroid
            wdt:P575 ?discovered; # Retrieve discovered date of the asteroid
            SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }

  BIND (?asteroidLabel AS ?asteroid_label)
  BIND (?discovererLabel AS ?discoverer_label)
}
ORDER BY DESC(?discovered)
```

5:13:46 PM GMT+2, Sep 24, 2024

Data updated **a few seconds ago**

Table | **97876 results in 28536 ms** | Code | Download | Link

Search

asteroidLabel	discovered	discovererLabel
2023 HT4	24 April 2023	Mount Lemmon Survey
2023 FY3	25 March 2023	Catalina Sky Survey
2023 FY3	25 March 2023	Kacper Wierzchoś
2023 DZ2	27 February 2023	http://www.wikidata.org/.well-known/genid/1fc11aca1fac426d0763053510afd316
2023 BU	21 January 2023	Gennady Borisov
2022 YG	15 December 2022	Gennady Borisov
2022 DX	22 February 2022	Erwin Schwab

Wikidata Query Service | Examples | Help | More tools | Query Builder

```
SELECT
?asteroidLabel
?discovered
?discovererLabel
WHERE {
  ?asteroid wdt:P31 wd:Q3863;  # Retrieve instances of "asteroid"
            wdt:P61 ?discoverer; # Retrieve discoverer of the asteroid
            wdt:P575 ?discovered; # Retrieve discovered date of the asteroid
            SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }

  BIND (?asteroidLabel AS ?asteroid_label)
  BIND (?discovererLabel AS ?discoverer_label)
}
ORDER BY DESC(?discovered)
```

Table

asteroidLabel	discovered	discovererLabel
2023 HT4	24 April 2023	Mount Lemmon Survey
2023 FY3	25 March 2023	Catalina Sky Survey
2023 FY3	25 March 2023	Kacper Wierzchoś
2023 DZ2	27 February 2023	http://www.wikidata.org/.well-known/
2023 BU	21 January 2023	Gennady Borisov
2022 YG	15 December 2022	Gennady Borisov
2022 DX	22 February 2022	Erwin Schwab

Write a WIKIDATA SPARQL query that returns asteroids with the following fields:

asteroid name or id, discovered date, discoverer

change the column names accordingly to fit WIKIDATAs ones

Sure! Here's a WIKIDATA SPARQL query that retrieves asteroid data including the asteroid name or ID, discovered date, and discoverer:

ruby Copy code

```
SELECT ?asteroid ?discovered ?discovererLabel
WHERE {
  ?asteroid wdt:P31 wd:Q3863;  # Retrieve instances of "asteroid"
            wdt:P61 ?discoverer; # Retrieve discoverer of the asteroid
            wdt:P575 ?discovered. # Retrieve discovered date of the asteroid
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
```

In the above query, we use the `SELECT` clause to specify the variables we want to retrieve, which are `?asteroid`, `?discovered`, and `?discovererLabel`. The `?asteroid` variable represents the ID or name of the asteroid, the `?discovered` variable represents the discovered date of the asteroid, and the `?discovererLabel` variable represents the discoverer of the asteroid.

In the `WHERE` clause, we use the `wdt:P31` property to filter for instances of "asteroid" (Q3863) in WIKIDATA. We then use the `wdt:P61` property to retrieve the discoverer of the asteroid, and the `wdt:P575` property to retrieve the discovered date of the asteroid. Finally, we use the `SERVICE` clause with `wikibase:label` to retrieve the English label for the discoverer using the `?discovererLabel` variable.

Note: Please note that the results may be limited to the available data in WIKIDATA, and some asteroids may not have complete information.

Datasets: datonic/wikidata_asteroids

like 0

Modalities: Text | Formats: parquet | Size: 10K - 100K | Libraries: Datasets, pandas, Croissant + 1 | License: mit

Dataset card | Viewer | Files and versions | Community 11 | Settings

Dataset Viewer

Auto-converted to Parquet | API | Embed | Full Screen Viewer

Split (1)
train · 97.4k rows

Search this dataset | SQL Console

asteroidLabel large_string · *lengths* 5 – 43	discovered large_string · *lengths* 20 – 20	discovererLabel large_string · *lengths* 4 – 74
2023 HT4	2023-04-24T00:00:00Z	Mount Lemmon Survey
2023 FY3	2023-03-25T00:00:00Z	Catalina Sky Survey
2023 FY3	2023-03-25T00:00:00Z	Kacper Wierzchoś
2023 DZ2	2023-02-27T00:00:00Z	http://www.wikidata.org/.well-known/genid/1fc11aca1fac426d0763053510afd316
2023 BU	2023-01-21T00:00:00Z	Gennady Borisov
2022 YG	2022-12-15T00:00:00Z	Gennady Borisov
2022 DX	2022-02-22T00:00:00Z	Erwin Schwab
2022 AE1	2022-01-06T00:00:00Z	Mount Lemmon Survey

‹ Previous | 1 | 2 | 3 | ... | 975 | Next ›

Downloads last month 2

Use this dataset | Edit dataset card

Size of downloaded dataset files: 848 kB

Size of the auto-converted Parquet files: 848 kB

Number of rows: 97,422

Índice de Precios de Consumo. Base 2021. Medias anuales

Resultados nacionales

Índices nacionales de clases

Unidades: Índice, Tasas

Seleccione valores a consultar

Clases	Tipo de dato	Periodo
Índice general	Media anual	2023
0111 Pan y cereales	Variación de las medias anuales	2022
0112 Carne		2021
0113 Pescado y marisco		2020
0114 Leche, queso y huevos		2019
0115 Aceites y grasas		2018
Seleccionados: 93 Total: 93	Seleccionados: 1 Total: 2	Seleccionados: 1 Total: 22

Elija forma de presentación de la tabla

Decimales a mostrar: Por defecto

Notas

Total: 93 series y 93 datos

Consultar selección Consultar todo

English

Instituto Nacional de Estadística

Escriba el texto para buscar

Censo Electoral | Sede electrónica | Compartir

INEbase / Índice… / Nivel y… / Índice … / Índice de Precios de Consumo. Base 2019. M…

INEbase

Índice de Precios de Consumo. Base 2021. Medias anuales

Resultados nacionales

Índices nacionales de clases

Unidades: Índice, Tasas

Seleccione valores a consultar

Clases	Tipo de dato	Periodo
Índice general	Media anual	2023
0111 Pan y cereales	Variación de las medias anuales	2022
0112 Carne		2021
0113 Pescado y marisco		2020
0114 Leche, queso y huevos		2019
0115 Aceites y grasas		2018
Seleccionados: 93 Total: 93	Seleccionados: 1 Total: 2	Seleccionados: 1 Total: 22

Elija forma de presentación de la tabla

Tipo de dato
Periodo
Clases

Decimales a mostrar: Por defecto

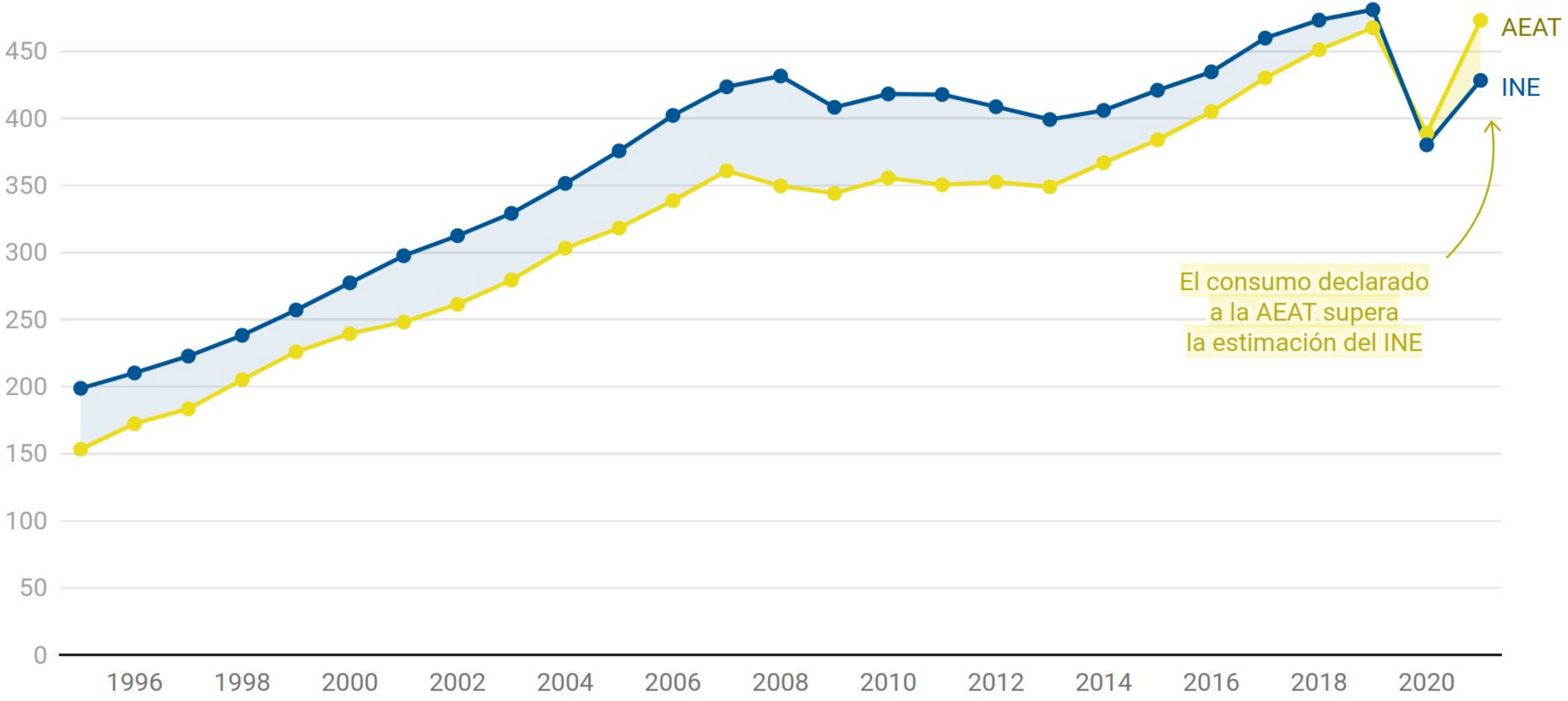
Notas

Total: 93 series y 93 datos

Consultar selección | Consultar todo

El consumo declarado en 2021 supera al estimado por el INE

Evolución del consumo sujeto a IVA según la estimación del PIB (INE) y los datos de consumo declarado a la Agencia Tributaria.
Datos en miles de millones de euros



Muchos datos, fragmentados y desactualizados.

Para usarlos, tenemos que repetir procesamiento.

Juntar datos no es divertido ni fácil.

Pocas herramientas para colaborar y mejorar el ecosistema.

Con datos se pueden tomar mejores decisiones.

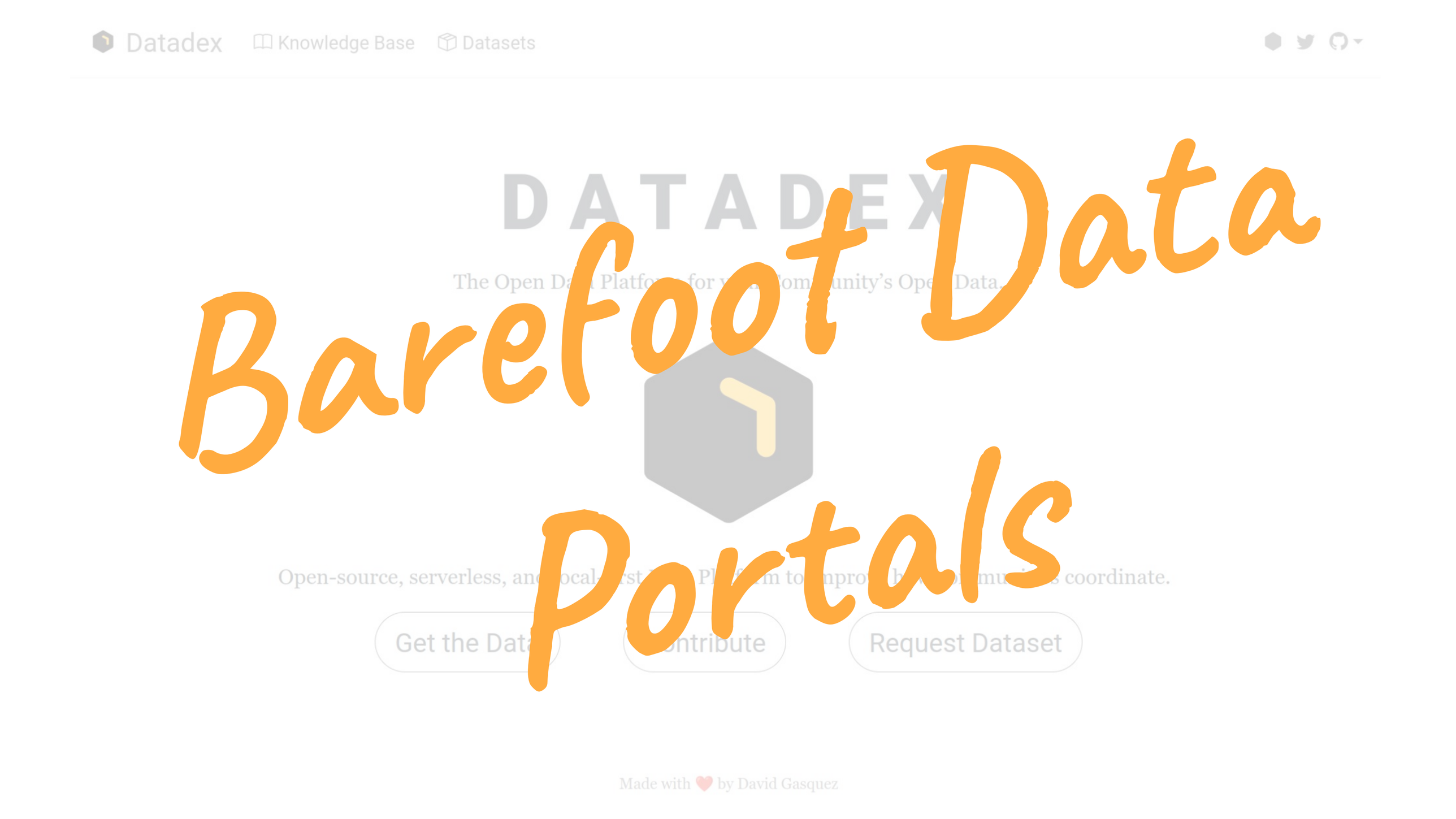
Conectar datos entre sí los hace más útiles.

DATADEX

The Open Data Platform for your Community's Open Data.

Open-source, serverless, and local-first Data Platform to improve how communities coordinate.

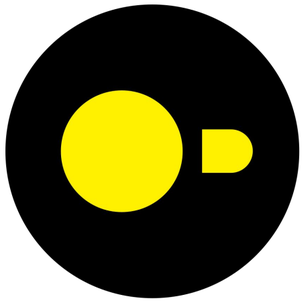
Get the Data | Contribute | Request Dataset


Datadex
Knowledge Base
Datasets
DATADEX
Get the Data
Request Dataset
Made with ❤ by David Gasquez
Barefoot Data Portals

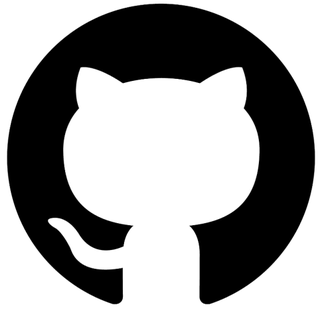

raw_ipc
Datos de la serie histórica del índice de Precios d...
Never materialized
ipc
Índice de Precios de Consumo (IPC) en España. E...
Never materialized
Checks
3
DuckDB
dbt
huggingface_ipc
Materialización del dataset en HuggingFace Data...
Never materialized

dagster

dbt

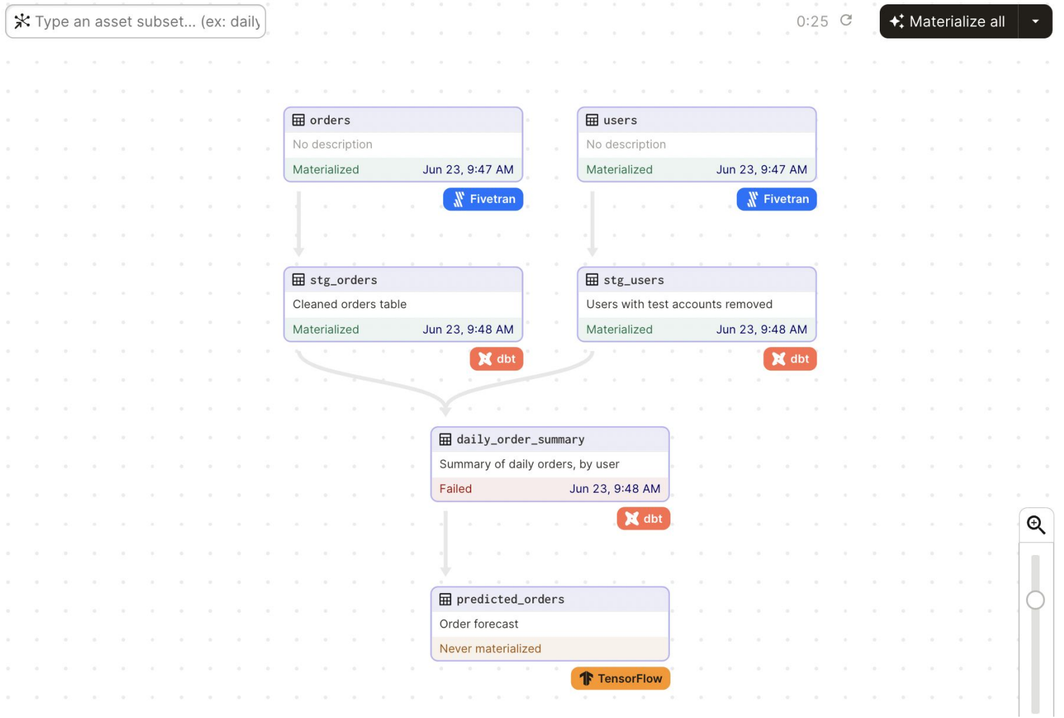


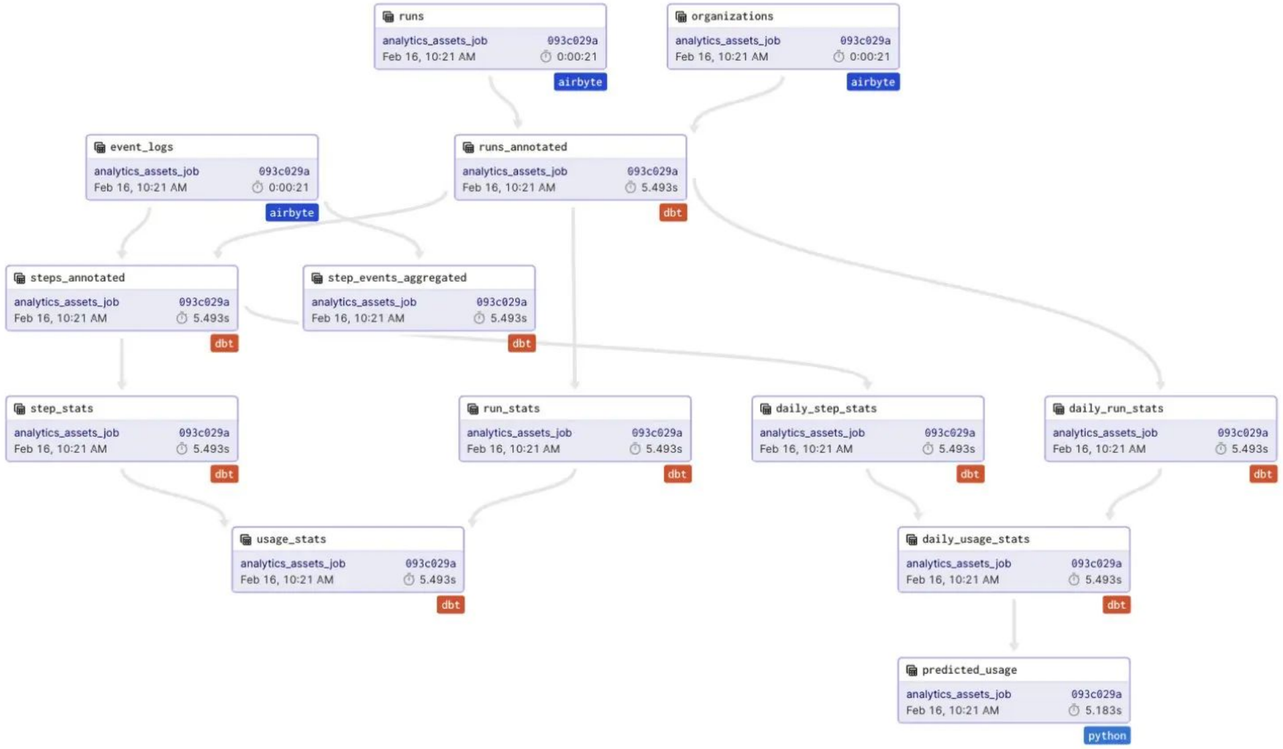
DuckDB

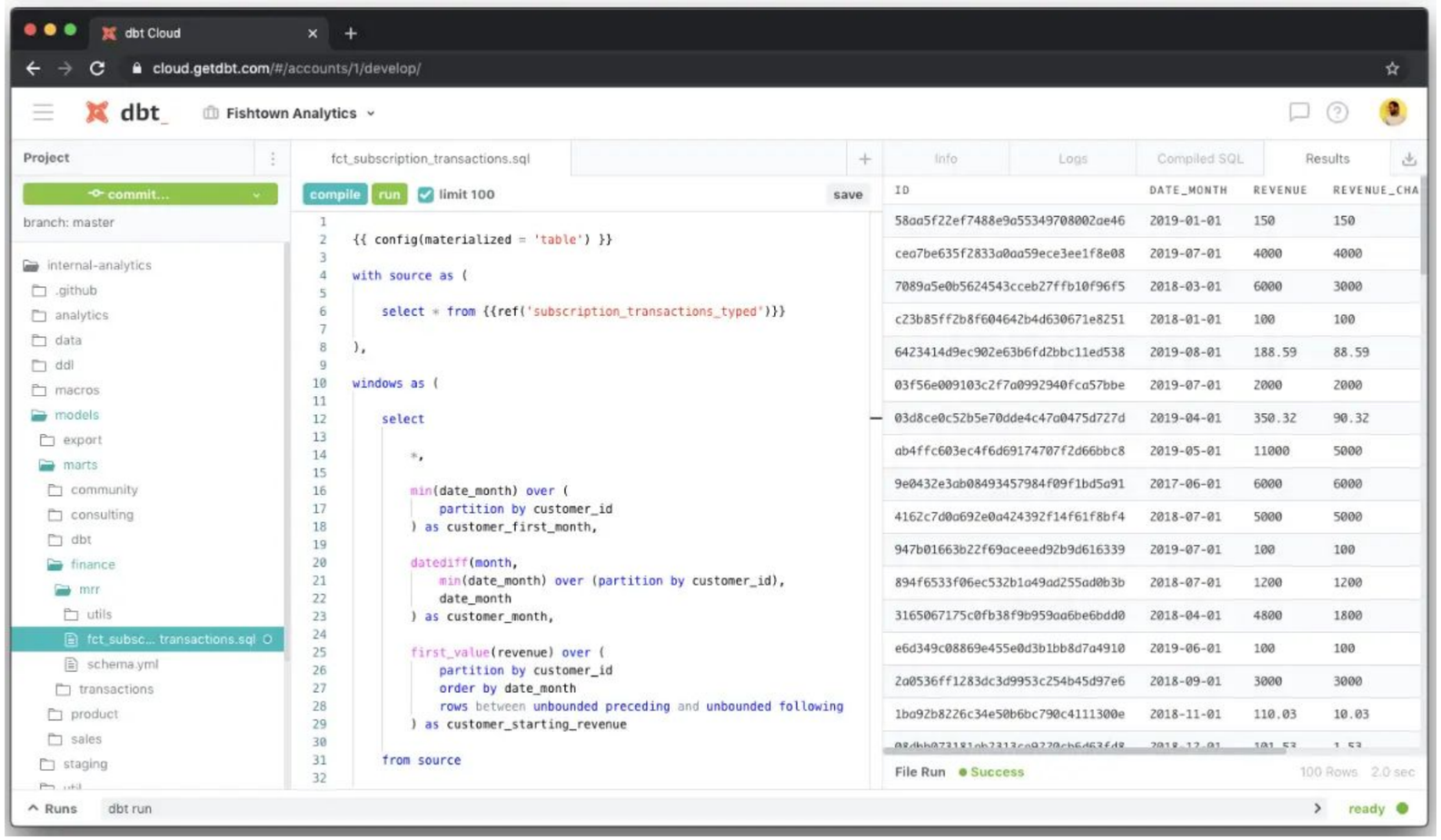


GitHub

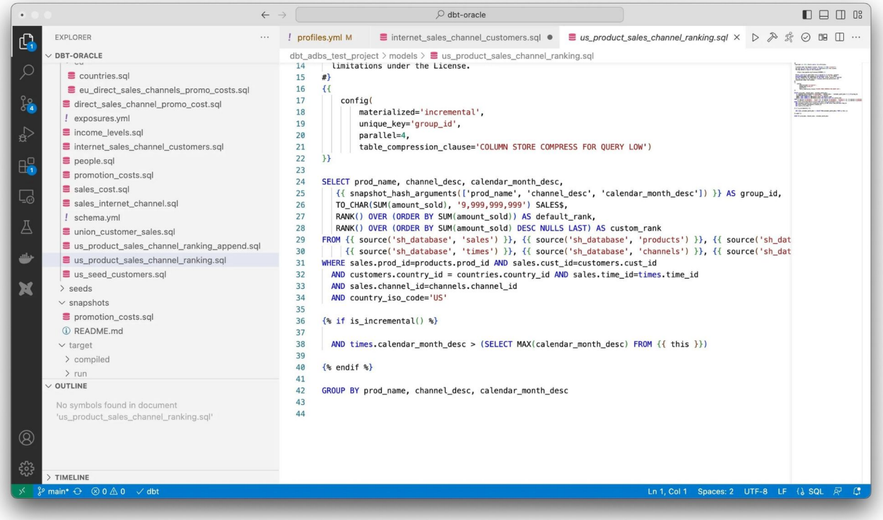
dagster

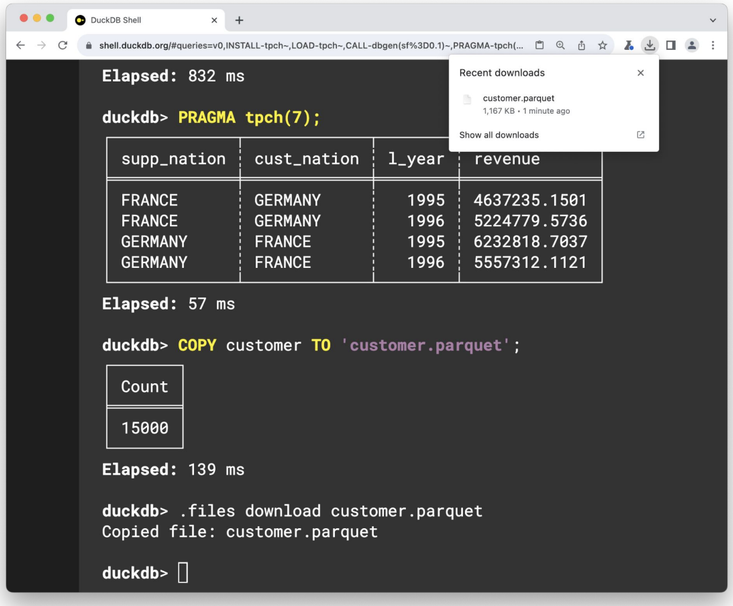

Type an asset subset... (ex: daily
0:25
Materialize all
orders
No description
Materialized Jun 23, 9:47 AM
Fivetran
users
No description
Materialized Jun 23, 9:47 AM
Fivetran
stg_orders
Cleaned orders table
Materialized Jun 23, 9:48 AM
dbt
stg_users
Users with test accounts removed
Materialized Jun 23, 9:48 AM
dbt
daily_order_summary
Summary of daily orders, by user
Failed Jun 23, 9:48 AM
dbt
predicted_orders
Order forecast
Never materialized
TensorFlow


runs
analytics_assets_job 093c029a
Feb 16, 10:21 AM 0:00:21
airbyte
organizations
analytics_assets_job 093c029a
Feb 16, 10:21 AM 0:00:21
airbyte
event_logs
analytics_assets_job 093c029a
Feb 16, 10:21 AM 0:00:21
airbyte
runs_annotated
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.493s
dbt
steps_annotated
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.493s
dbt
step_events_aggregated
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.493s
dbt
step_stats
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.493s
dbt
run_stats
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.493s
dbt
daily_step_stats
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.493s
dbt
daily_run_stats
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.493s
dbt
usage_stats
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.493s
dbt
daily_usage_stats
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.493s
dbt
predicted_usage
analytics_assets_job 093c029a
Feb 16, 10:21 AM 5.183s
python


dbt Cloud
cloud.getdbt.com/#/accounts/1/develop/
dbt_
Fishtown Analytics
Project
commit...
branch: master
internal-analytics
.github
analytics
data
ddl
macros
models
export
marts
community
consulting
dbt
finance
mrr
utils
fct_subsc... transactions.sql
schema.yml
transactions
product
sales
staging
fct_subscription_transactions.sql
compile
run
limit 100
save
{{ config(materialized = 'table') }}
with source as (
select * from {{ref('subscription_transactions_typed')}}
),
windows as (
select
*,
min(date_month) over (
partition by customer_id
) as customer_first_month,
datediff(month,
min(date_month) over (partition by customer_id),
date_month
) as customer_month,
first_value(revenue) over (
partition by customer_id
order by date_month
rows between unbounded preceding and unbounded following
) as customer_starting_revenue
from source
Info
Logs
Compiled SQL
Results
ID
DATE_MONTH
REVENUE
REVENUE_CHA
58aa5f22ef7488e9a55349708002ae46 2019-01-01 150 150
cea7be635f2833a0aa59ece3ee1f8e08 2019-07-01 4000 4000
7089a5e0b5624543cceb27ffb10f96f5 2018-03-01 6000 3000
c23b85ff2b8f604642b4d630671e8251 2018-01-01 100 100
6423414d9ec902e63b6fd2bbc11ed538 2019-08-01 188.59 88.59
03f56e009103c2f7a0992940fca57bbe 2019-07-01 2000 2000
03d8ce0c52b5e70dde4c47a0475d727d 2019-04-01 350.32 90.32
ab4ffc603ec4f6d69174707f2d66bbc8 2019-05-01 11000 5000
9e0432e3ab08493457984f09f1bd5a91 2017-06-01 6000 6000
4162c7d0a692e0a424392f14f61f8bf4 2018-07-01 5000 5000
947b01663b22f69aceeed92b9d616339 2019-07-01 100 100
894f6533f06ec532b1a49ad255ad0b3b 2018-07-01 1200 1200
3165067175c0fb38f9b959aa6be6bdd0 2018-04-01 4800 1800
e6d349c08869e455e0d3b1bb8d7a4910 2019-06-01 100 100
2a0536ff1283dc3d9953c254b45d97e6 2018-09-01 3000 3000
1ba92b8226c34e50b6bc790c4111300e 2018-11-01 110.03 10.03
File Run
Success
100 Rows
2.0 sec
Runs
dbt run
ready

dbt™


dbt-oracle
EXPLORER
DBT-ORACLE
countries.sql
eu_direct_sales_channels_promo_costs.sql
direct_sales_channel_promo_cost.sql
exposures.yml
income_levels.sql
internet_sales_channel_customers.sql
people.sql
promotion_costs.sql
sales_cost.sql
sales_internet_channel.sql
schema.yml
union_customer_sales.sql
us_product_sales_channel_ranking_append.sql
us_product_sales_channel_ranking.sql
us_seed_customers.sql
seeds
snapshots
promotion_costs.sql
README.md
target
compiled
run
OUTLINE
No symbols found in document 'us_product_sales_channel_ranking.sql'
TIMELINE
profiles.yml M
internet_sales_channel_customers.sql
us_product_sales_channel_ranking.sql
dbt_adbs_test_project > models > us_product_sales_channel_ranking.sql
limitations under the License.
#}
{{
config(
materialized='incremental',
unique_key='group_id',
parallel=4,
table_compression_clause='COLUMN STORE COMPRESS FOR QUERY LOW')
}}
SELECT prod_name, channel_desc, calendar_month_desc,
{{ snapshot_hash_arguments(['prod_name', 'channel_desc', 'calendar_month_desc']) }} AS group_id,
TO_CHAR(SUM(amount_sold), '9,999,999,999') SALES$,
RANK() OVER (ORDER BY SUM(amount_sold)) AS default_rank,
RANK() OVER (ORDER BY SUM(amount_sold) DESC NULLS LAST) AS custom_rank
FROM {{ source('sh_database', 'sales') }}, {{ source('sh_database', 'products') }}, {{ source('sh_dat
{{ source('sh_database', 'times') }}, {{ source('sh_database', 'channels') }}, {{ source('sh_dat
WHERE sales.prod_id=products.prod_id AND sales.cust_id=customers.cust_id
AND customers.country_id = countries.country_id AND sales.time_id=times.time_id
AND sales.channel_id=channels.channel_id
AND country_iso_code='US'
{% if is_incremental() %}
AND times.calendar_month_desc > (SELECT MAX(calendar_month_desc) FROM {{ this }})
{% endif %}
GROUP BY prod_name, channel_desc, calendar_month_desc
main*
dbt
Ln 1, Col 1
Spaces: 2
UTF-8
LF
SQL

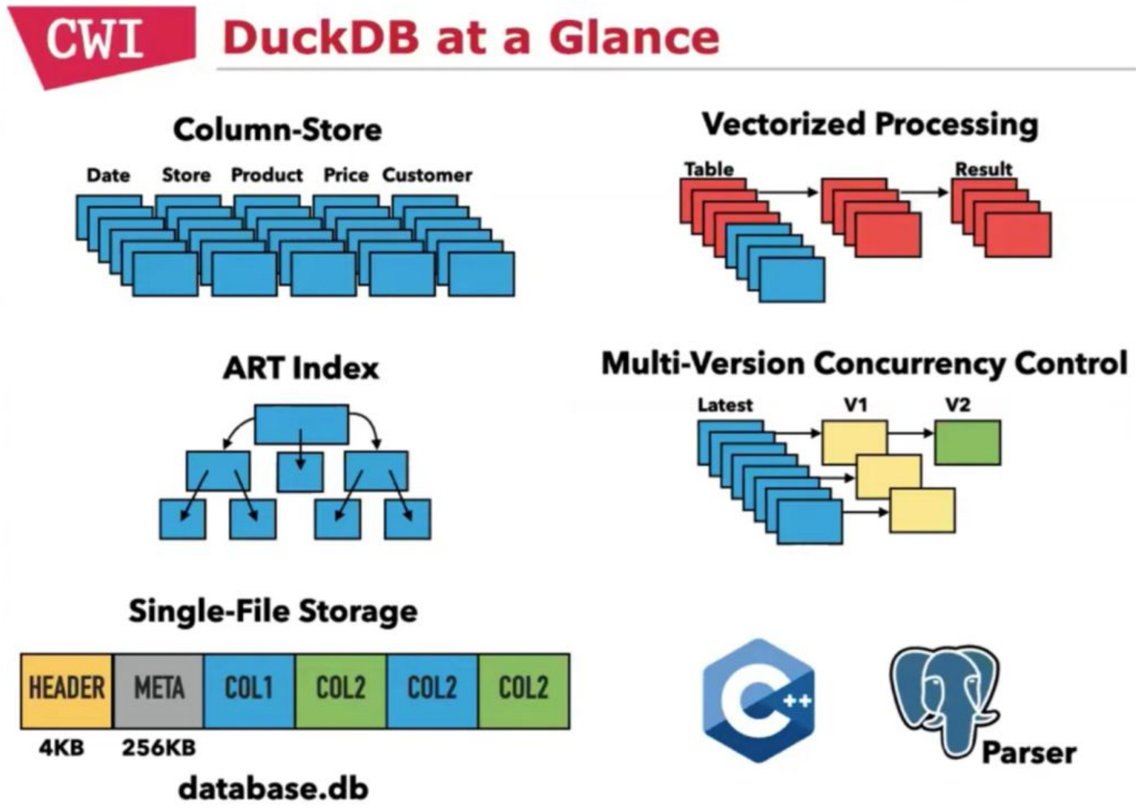



```
input_df = pd.DataFrame.from_dict({"i":[1, 2, 3],
                                   "j":["Apple", "Mango", "Oranges"]})
```

```
%sql output_df << SELECT sum(i) as total_i FROM input_df
output_df
```

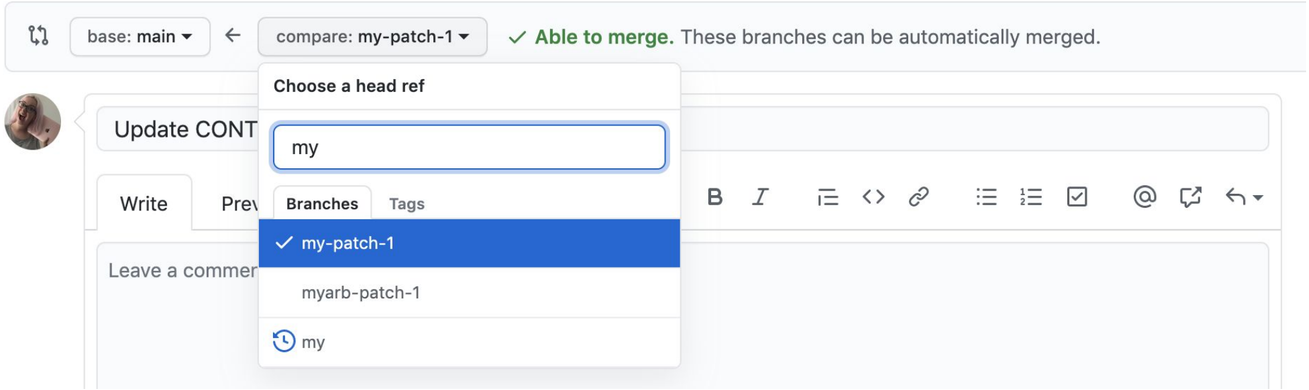
	total_i
0	6.0

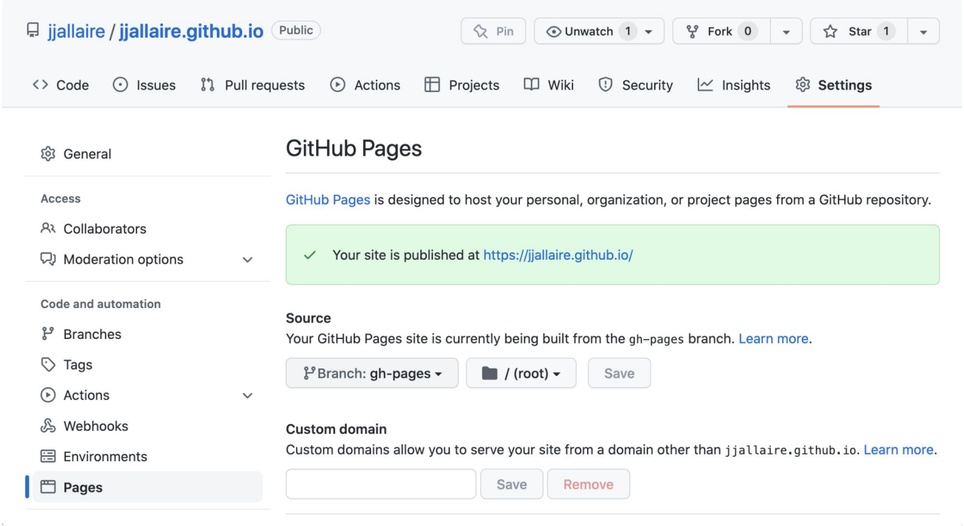
DuckDB

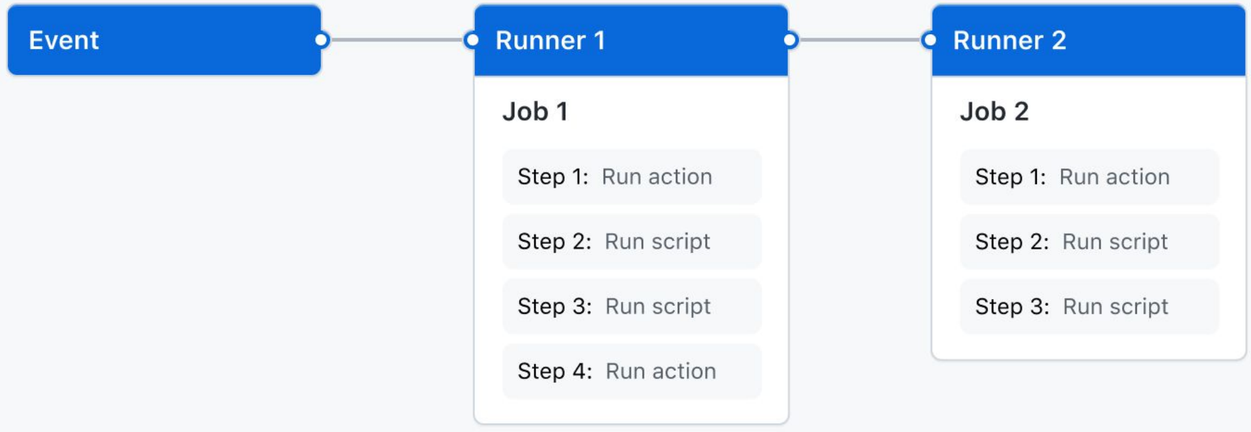


Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.



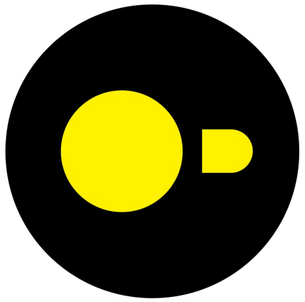




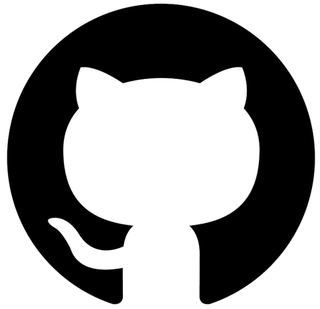
GitHub

dagster

dbt



DuckDB



GitHub

Apache
Airflow

datadex

Public

Edit Pins | Unwatch 4 | Fork 15 | Star 246

main | 1 Branch | 9 Tags | Go to file t | Add file | Code

davidgasquez feat: ✨ Add multi-commit support in DatasetPublisher 93da061 · 2 months ago 329 Commits

.devcontainer	chore: 📌 remove sqfmt [skip ci]	6 months ago
.github/workflows	chore: 🔧 adjust concurrency and python version in ci	5 months ago
data	fix: 🙈 don't ignore the data folder	9 months ago
datadex	feat: ✨ Add multi-commit support in DatasetPublisher	2 months ago
dbt	feat: ⚡ enable spain weather assets	5 months ago
portal	feat: 🚀 Improve performance and UX with multiple enh...	5 months ago
.editorconfig	chore: 🔧	last year
.gitignore	refactor: 💡 consolidate indicators assets under polars a...	5 months ago
Dockerfile	feat: 🐍 Upgrade Python and refactor for polars integrati...	5 months ago
LICENSE	Initial commit	2 years ago
Makefile	feat: 🎨 update code to use new dbt classes	2 months ago
README.md	docs: 📝 add lung-sarg [skip ci]	2 months ago
pyproject.toml	feat: ✨ Add multi-commit support in DatasetPublisher	2 months ago

README | MIT license

DATADEX

The Open Data Platform for your community Open Data

About

📦 Serverless and local-first Open Data Platform

datadex.datonic.io

sql open-data dbt quarto duckdb

Readme
MIT license
Activity
Custom properties
246 stars
4 watching
15 forks

Contributors 5

Languages

Jupyter Notebook 75.2%
Python 23.2%
Other 1.6%

```python
@asset()
def spain_aemet_stations_data(aemet_api: AEMETAPI) -> pl.DataFrame:
    """
    Spain AEMET stations data.
    """

    df = pl.DataFrame(aemet_api.get_all_stations())
    df.with_columns(pl.col("indsinop").cast(pl.Int32, strict=False).alias("indsinop"))

    # Clean latitud and longitud
    def convert_to_decimal(coord):
        degrees = int(coord[:-1][:2])
        minutes = int(coord[:-1][2:4])
        seconds = int(coord[:-1][4:])
        decimal = degrees + minutes / 60 + seconds / 3600
        if coord[-1] in ["S", "W"]:
            decimal = -decimal
        return decimal

    df = df.with_columns(
        [
            pl.col("latitud").map_elements(convert_to_decimal).alias("latitud"),
            pl.col("longitud").map_elements(convert_to_decimal).alias("longitud"),
        ]
    )
```

ESTACIONES

```python
@asset()
def spain_aemet_weather_data(
    context: AssetExecutionContext, aemet_api: AEMETAPI
) -> pl.DataFrame:
    """
    Spain weather data since 1940.
    """

    start_date = datetime(1940, 1, 1)
    end_date = datetime.now()

    r = aemet_api.get_weather_data(start_date, end_date)

    df = pl.DataFrame()
    for d in r:
        ndf = pl.DataFrame(d)
        df = pl.concat([df, ndf], how="diagonal_relaxed")

    df = df.with_columns(pl.col("fecha").str.strptime(pl.Date, format="%Y-%m-%d"))

    float_columns = [
        "prec",
        "presMax",
        "presMin",
        "racha",
        "sol",
        "tmax",
        "tmed",
        "tmin",
        "velmedia",
    ]

    df = df.with_columns(
        [
            pl.col(col).str.replace(",", ".").cast(pl.Float64, strict=False)
            for col in float_columns
        ]
    )
```

TIEMPO

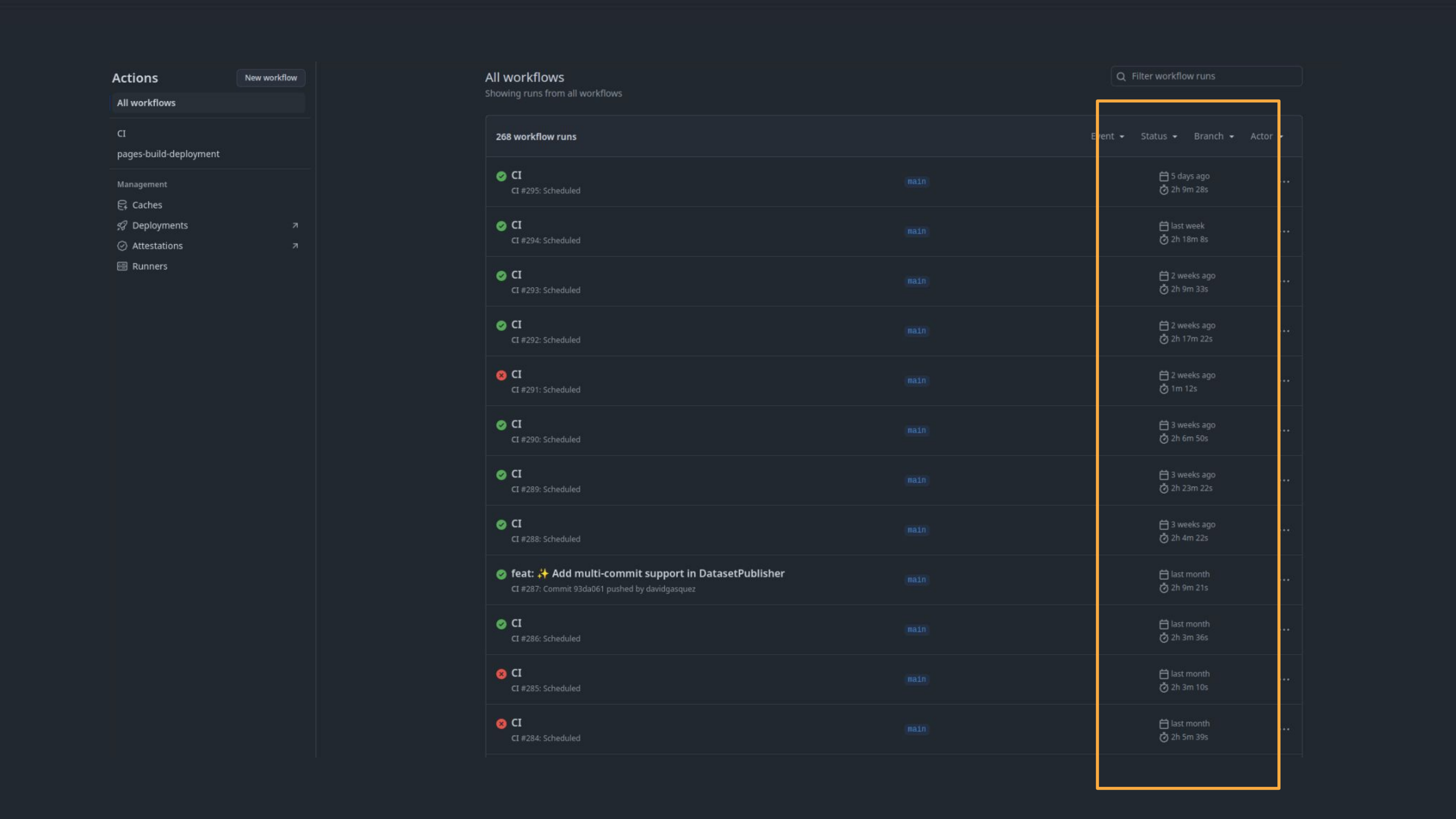
```python
@asset()
def spain_aemet_stations_data(aemet_api: AEMET
    """
    Spain AEMET stations data.
    """

    df = pl.DataFrame(aemet_api.get_all_stati
    df.with_columns(pl.col("indsinop").cast(p

    # Clean latitud and longitud
    def convert_to_decimal(coord):
        degrees = int(coord[:-1][:2])
        minutes = int(coord[:-1][2:4])
        seconds = int(coord[:-1][4:])
        decimal = degrees + minutes / 60 + se
        if coord[-1] in ["S", "W"]:
            decimal = -decimal
        return decimal

    df = df.with_columns(
        [
            pl.col("latitud").map_elements(co
            pl.col("longitud").map_elements(c
        ]
    )

```

```python
@asset()
def spain_aemet_weather_data(
, aemet_api: AEMETAPI

 1)

(start_date, end_date)

how="diagonal_relaxed")

echa").str.strptime(pl.Date, format="%Y-%m-%d"))

e(",", ".").cast(pl.Float64, strict=False)
ns
    ]
)
```

```sql
select
    cast(w.fecha as date) as fecha,
    w.indicativo,
    w.nombre,
    w.provincia,
    s.latitud,
    s.longitud,
    cast(w.altitud as int) as altitud,
    cast(w.tmed as float) as tmed,
    cast(w.prec as float) as prec,
    cast(w.tmin as float) as tmin,
    w.horatmin,
    cast(w.tmax as float) as tmax,
    w.horatmax,
    cast(w.dir as int) as dir,
    cast(w.velmedia as float) as velmedia,
    cast(w.racha as float) as racha,
    w.horaracha,
    cast(w.presMax as float) as presMax,
    w.horaPresMax,
    cast(w.presMin as float) as presMin,
    w.horaPresMin,
    cast(w.hrMedia as int) as hrMedia,
    cast(w.hrMax as int) as hrMax,
    w.horaHrMax,
    cast(w.hrMin as int) as hrMin,
    w.horaHrMin,
    cast(w.sol as float) as sol
from {{ source('main', 'spain_aemet_weather_data') }} as w
left join {{ source('main', 'spain_aemet_stations_data') }} as s
    on w.indicativo = s.indicativo
order by w.fecha, w.indicativo
```

TIEMPO

ESTACIONES


Actions
New workflow
All workflows
CI
pages-build-deployment
Management
Caches
Deployments
Attestations
Runners
All workflows
Showing runs from all workflows
Filter workflow runs
268 workflow runs
Event
Status
Branch
Actor
CI
CI #295: Scheduled
main
5 days ago
2h 9m 28s
CI
CI #294: Scheduled
main
last week
2h 18m 8s
CI
CI #293: Scheduled
main
2 weeks ago
2h 9m 33s
CI
CI #292: Scheduled
main
2 weeks ago
2h 17m 22s
CI
CI #291: Scheduled
main
2 weeks ago
1m 12s
CI
CI #290: Scheduled
main
3 weeks ago
2h 6m 50s
CI
CI #289: Scheduled
main
3 weeks ago
2h 23m 22s
CI
CI #288: Scheduled
main
3 weeks ago
2h 4m 22s
feat: ✨ Add multi-commit support in DatasetPublisher
CI #287: Commit 93da061 pushed by davidgasquez
main
last month
2h 9m 21s
CI
CI #286: Scheduled
main
last month
2h 3m 36s
CI
CI #285: Scheduled
main
last month
2h 3m 10s
CI
CI #284: Scheduled
main
last month
2h 5m 39s

Datasets: datonic / **spain_aemet_historical_weather** like 1

Dataset card | Viewer | Files and versions | Community 9 | Settings

Split (1)
train · 8.11M rows

Ordered by fecha (desc)

Share results 8,114,317 rows

fecha unknown	indicativo large_string · classes 947 values	nombre large_string · classes 934 values	provincia large_string · classes 53 values	latitud float64 27.7 43.8	longitud float64 -18.12 4.32	altitud int32 1 3.09k	tmed float32 -21.5 40.4 ⌀	prec float32 0 360 ⌀	tmin float32 -30 50 ⌀	horatmin large_string · lengths 5 6 ⌀	tmax float32 -50 47.6 ⌀	horatmax large_string · le 5 6 ⌀
"2024-09-15"	0016A	REUS AEROPUERTO	TARRAGONA	41.145	1.163611	71	19	0	10	05:35	28	12:38
"2024-09-15"	0034X	VALLS	TARRAGONA	41.293056	1.260833	233	18.4	0	8.8	05:14	28	13:58
"2024-09-15"	0042Y	TARRAGONA	TARRAGONA	41.123889	1.249167	55	19.799999	0	11.7	06:10	28	12:07
"2024-09-15"	0061X	PONTONS	BARCELONA	41.416944	1.519167	632	16.5	0	7.4	02:24	25.6	14:43
"2024-09-15"	0066X	VILAFRANCA DEL PENEDÈS	BARCELONA	41.330278	1.676944	177	19.4	0	10.3	05:31	28.5	14:37
"2024-09-15"	0073X	SITGES	BARCELONA	41.243889	1.8525	58	19.6	0	12.6	04:38	26.6	13:28
"2024-09-15"	0076	BARCELONA AEROPUERTO	BARCELONA	41.292778	2.07	4	20.200001	0	14	05:30	26.299999	14:13
"2024-09-15"	0092X	BERGA	BARCELONA	42.101389	1.8575	682	16.299999	0	7.6	05:06	25	14:46
"2024-09-15"	0106X	BALSARENY	BARCELONA	41.866389	1.8725	361	16	0	4.7	05:14	27.299999	14:14
"2024-09-15"	0114X	PRATS DE LLUÇANÈS	BARCELONA	42.006944	2.026667	700	15.8	0	6	05:54	25.5	15:31
"2024-09-15"	0120X	MOIÀ	BARCELONA	41.813333	2.095278	742	17	0	10.3	05:23	23.700001	13:55
"2024-09-15"	0149X	MANRESA	BARCELONA	41.72	1.840278	291	17.4	0	7.4	06:10	27.5	14:40
"2024-09-15"	0158X	MONISTROL DE MONTSERRAT	BARCELONA	41.594444	1.839167	738	17	0	11.7	05:23	22.299999	13:24
"2024-												

main | 1 Branch | 0 Tags | Go to file t | Add file | Code

Automated Sat Oct 12 16:04:47 UTC 2024 1187e25 · 24 minutes ago 177 Commits

.github/workflows	Scrape outage.tecoenergy.com	2 days ago
duke-energy.app	Sat Oct 12 16:04:47 UTC 2024	24 minutes ago
fplmaps.com	Sat Oct 12 16:04:47 UTC 2024	24 minutes ago
outage.tecoenergy.com	Sat Oct 12 16:04:47 UTC 2024	24 minutes ago

```yaml
jobs:
  fetch:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v4
    - name: Scrape fplmaps.com
      run: |
        mkdir -p fplmaps.com
        if ! curl 'https://www.fplmaps.com/customer/outage/StormFeedRestoration.json' \
          -H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:131.0) Gecko/20100101 Firefox/131.0' \
          -H 'Accept: application/json, text/javascript, */*; q=0.01' \
          | jq > fplmaps.com/StormFeedRestoration.json; then
          echo "Failed to scrape fplmaps.com, but continuing..."
        fi
```

About

No description, website, or topics provided.

- Activity
- 5 stars
- 1 watching
- 1 fork

Report repository

Releases

No releases published

Sponsor this project

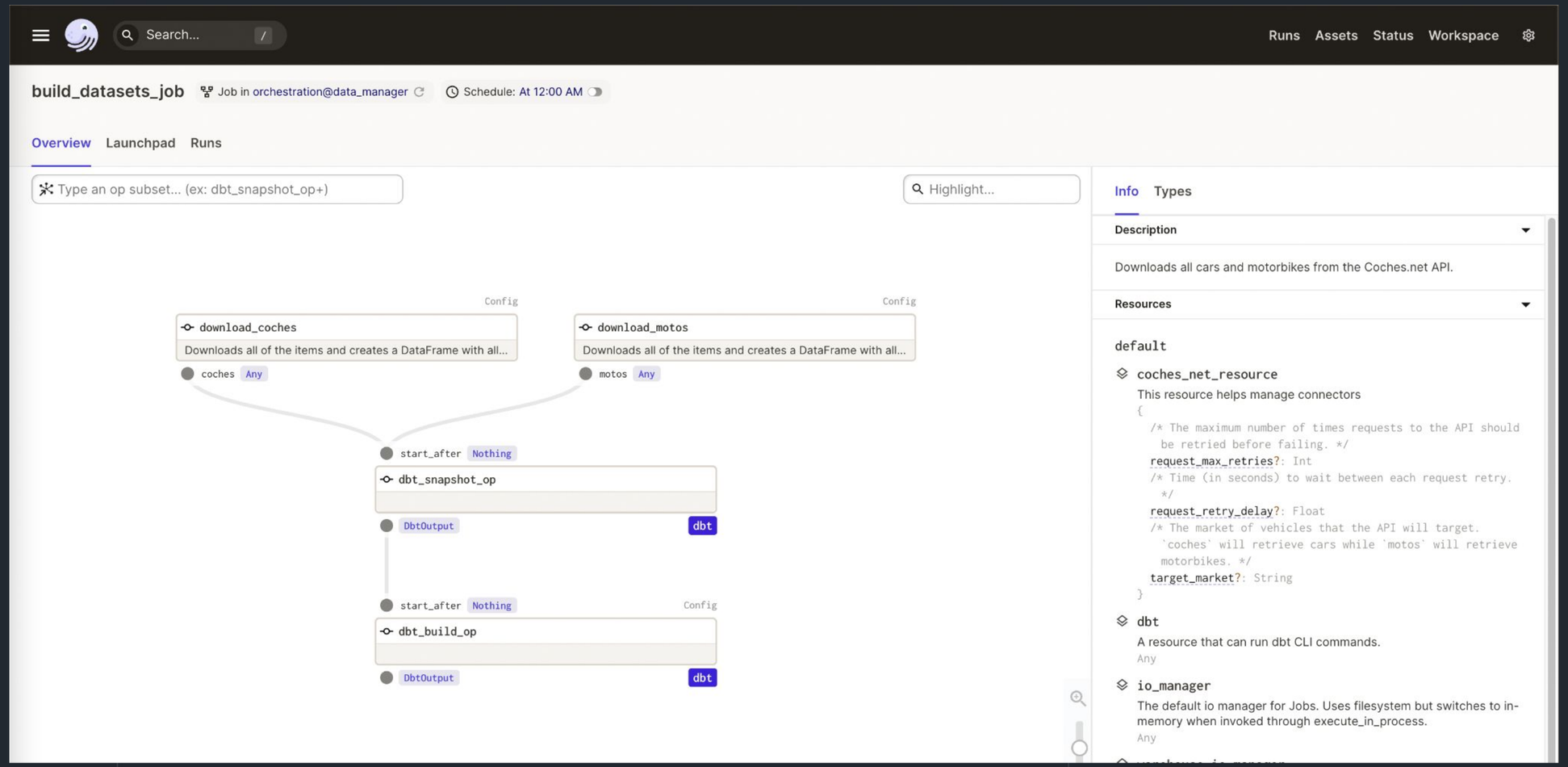
simonw Simon Willison

Sponsor

Learn more about GitHub Sponsors

Packages

No packages published



coches-net-dashboard

Coches.net dashboard is an application made by @franloza and @Jorjatorz to visualize Spanish car and motorcycle market through the data obtained from the websites coches.net and motos.coches.net.

This project is composed of a data pipeline run with Dagster, which extract data from coches.net, apply some transformations using dbt, and stores the data in a DuckDB database. The data can be analyzed with a dashboard built with Dash.

This application should not be considered suitable for production, and it's intended to be used only locally for analytical purposes.

Contributors 2

- Jorjatorz Jorge Sanchez
- franloza Fran Lozano

Languages

- Python 97.6%
- Dockerfile 2.4%

The Public Utility Data Liberation (PUDL) Project

ACCESS PUDL DATA PUDL ON GITHUB

Electric utilities report a huge amount of information to the US government and other public agencies. This includes yearly, monthly, and even hourly data about fuel burned, electricity generated, operating expenses, power plant usage patterns and emissions. Unfortunately, much of this data is not released in well documented, ready-to-use, machine readable formats. Data from different agencies tends not to be standardized or easily used in tandem. Several commercial data services clean, package, and re-sell this this data, but at prices which are too high to be accessible to many smaller stakeholders.

The **Public Utility Data Liberation (PUDL) project** takes information that's already publicly *available*, and makes it publicly *usable*, by cleaning, standardizing, and cross-linking utility data from different sources in a single database. Thus far our primary focus has been on fuel use, generation, operating costs, and operation history. As of November, 2022 PUDL integrates data from:

- EIA Form 860: 2001-2023
- EIA Form 860m: 2023-06
- EIA Form 861: 2001-2022
- EIA Form 923: 2001-2022
- EPA Continuous Emissions Monitoring System (CEMS): 1995-2022
- FERC Form 1: 1994-2022
- FERC Form 714: 2006-2020

Research and data to make progress against the world's largest problems.

Try "Life expectancy", "Poverty Nigeria Vietnam", "CO2 France"...

13,189 charts across 119 topics — All free: open access and open source

OUR MISSION

What do we need to know to make the world a better place?

To make progress against the pressing problems the world faces, we need to be informed by the best research and data.

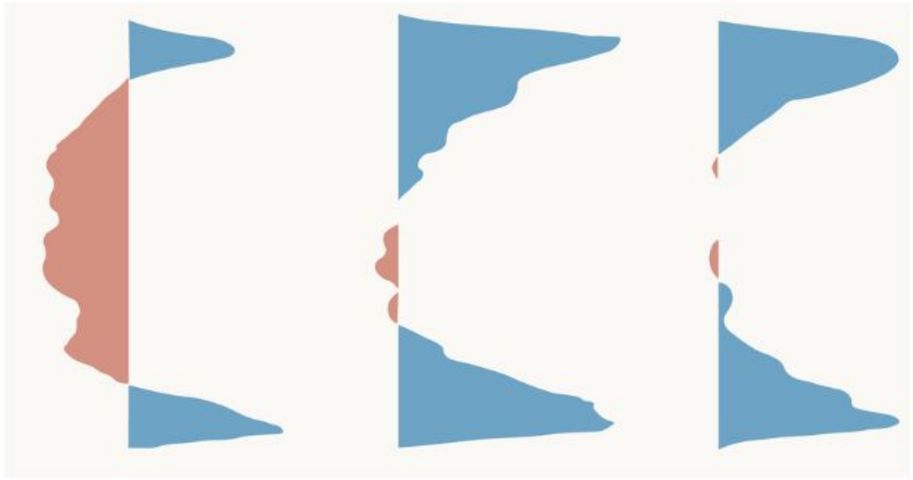
Our World in Data makes this knowledge accessible and understandable, to empower those working to build a better world.

Read about our mission →

Subscribe to our newsletter

We are a non-profit — all our work is free to use and open source. Consider supporting us if you find our work valuable.

FEATURED WORK



ARTICLE · 15 MIN READ

How will climate change affect crop yields in the future?

Maize yields could see significant declines, but wheat could increase. Impacts across the world will be very different.

Hannah Ritchie

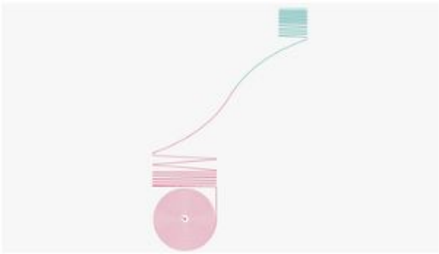
REVISED & UPDATED

Cancer

Saloni Dattani, Veronika Samborska, Hannah

ARTICLE · 12 MIN READ

Crop yields have increased dramatically in recent decades, but

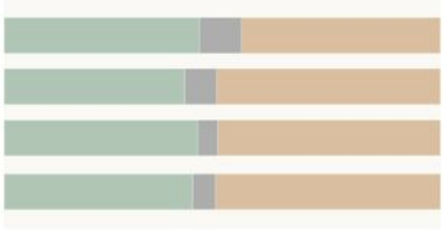


FROM OUR CLASSICS

Technology over the long run: zoom out to see how dramatically the world can change within a lifetime

It is easy to underestimate the magnitude of this change. Understanding this can help us see how different the world could be in the future.

Max Roser



ETL model

This section presents a high-level view of our ETL processes.

In general, an ETL consists of three main processes:

- **Extract**: Downloads data from a source or multiple sources.
- **Transform**: Processes and combines the downloaded data so it is usable.
- **Load**: Moves the transformed data to a data repository (e.g. database).

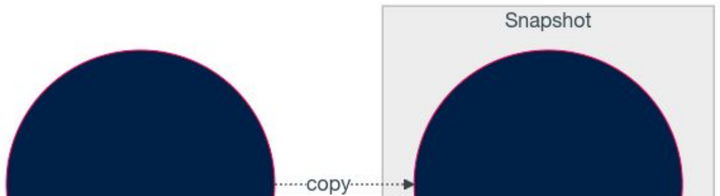
Our ETL follows these same principles, with different names, and has the following three main building blocks:

- **Snapshots**: We download files from external sources and store them as snapshots.
- **Datasets**: Next, we curate the downloaded data to have clean datasets. These datasets can depend on snapshots, but also on other curated datasets.
- **Grapher views**: We adapt the curated datasets for Grapher, which are then loaded to Grapher.

As a consequence, nodes in our DAG can come in different flavours.

Snapshots

Snapshots are edge nodes in the computational graph. They represent a copy of a upstream data file — preserving its original format — at a particular point in time. That is, they are entry points to the ETL, and therefore don't have dependencies within the computational graph.



Plataformas de Datos que no añaden datos nuevos, sino que juntan, limpian datos existentes, haciéndolos útiles y fáciles de acceder para una comunidad.

1. Un repositorio donde poner el código para colaborar y un sitio para ejecutarlo.
2. Trabajar con estándares y tecnologías modernas (ahorra tiempo y dinero).
3. Publicar ficheros estáticos (no APIs) que se actualizan automáticamente.
4. Trabajar cerca de la comunidad para saber cuáles son sus problemas y donde pueden ayudar los datos.

¡Gracias!

¿Preguntas? ¿Feedback?

@davidgasquez

DATALIA

Datos, sin complicaciones. 💃

Datalia es una plataforma de datos abiertos a nivel de España con el objetivo de unificar y armonizar información proveniente de diferentes fuentes.

Datasets

Gracias a la colaboración de la comunidad, Datalia ofrece una variedad de conjuntos de datos. A continuación, se muestran algunos de los conjuntos de datos disponibles.

- IPC

Recursos

Algunos recursos que pueden ser de utilidad relacionados con datos abiertos y transparencia a nivel de España.

Portales

- INE.
- Datos Abiertos de España.
- DGT. Tambien en mapa.
- AEMET.
- WikiData.
- ESRI.
- Datadista.
- Newtral.
- Spanish Origin Destination Data
- DataMarket.