# ggsql:
# A grammar of graphics for SQL

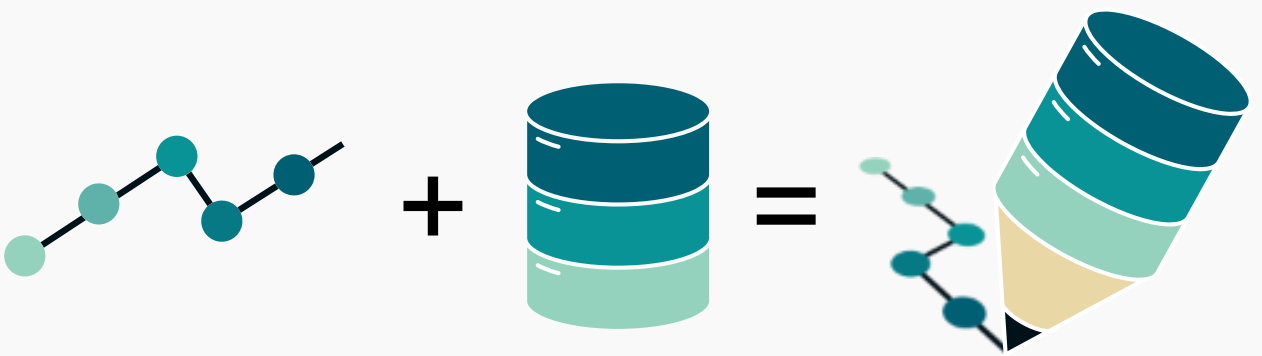
Teun van den Brand
DuckCon #7, Amsterdam
2026-06-24

# What is ggsql?

Grammar of Graphics

+

Structured Query Language



# A quick example

```
SELECT * FROM 'assets/penguins.parquet'
  WHERE island == 'Biscoe'
```

| species | island | bill_len | bill_dep | flipper_len | body_mass | sex | year |
|---|---|---|---|---|---|---|---|
| Adelie | Biscoe | 37.8 | 18.3 | 174 | 3400 | female | 2007 |
| Adelie | Biscoe | 37.7 | 18.7 | 180 | 3600 | male | 2007 |
| Adelie | Biscoe | 35.9 | 19.2 | 189 | 3800 | female | 2007 |
| Adelie | Biscoe | 38.2 | 18.1 | 185 | 3950 | male | 2007 |
| Adelie | Biscoe | 38.8 | 17.2 | 180 | 3800 | male | 2007 |
| Adelie | Biscoe | 35.3 | 18.9 | 187 | 3800 | female | 2007 |
| Adelie | Biscoe | 40.6 | 18.6 | 183 | 3550 | male | 2007 |
| Adelie | Biscoe | 40.5 | 17.9 | 187 | 3200 | female | 2007 |
| Adelie | Biscoe | 37.9 | 18.6 | 172 | 3150 | female | 2007 |
| Adelie | Biscoe | 40.5 | 18.9 | 180 | 3950 | male | 2007 |
| Adelie | Biscoe | 39.6 | 17.7 | 186 | 3500 | female | 2008 |
| Adelie | Biscoe | 40.1 | 18.9 | 188 | 4300 | male | 2008 |
| Adelie | Biscoe | 35 | 17.9 | 190 | 3450 | female | 2008 |
| Adelie | Biscoe | 42 | 19.5 | 200 | 4050 | male | 2008 |

| species | island | bill_len | bill_dep | flipper_len | body_mass | sex | year |
|---|---|---|---|---|---|---|---|
| Adelie | Biscoe | 34.5 | 18.1 | 187 | 2900 | female | 2008 |
| Adelie | Biscoe | 41.4 | 18.6 | 191 | 3700 | male | 2008 |
| Adelie | Biscoe | 39 | 17.5 | 186 | 3550 | female | 2008 |
| Adelie | Biscoe | 40.6 | 18.8 | 193 | 3800 | male | 2008 |
| Adelie | Biscoe | 36.5 | 16.6 | 181 | 2850 | female | 2008 |
| Adelie | Biscoe | 37.6 | 19.1 | 194 | 3750 | male | 2008 |
| Adelie | Biscoe | 35.7 | 16.9 | 185 | 3150 | female | 2008 |
| Adelie | Biscoe | 41.3 | 21.1 | 195 | 4400 | male | 2008 |
| Adelie | Biscoe | 37.6 | 17 | 185 | 3600 | female | 2008 |
| Adelie | Biscoe | 41.1 | 18.2 | 192 | 4050 | male | 2008 |
| Adelie | Biscoe | 36.4 | 17.1 | 184 | 2850 | female | 2008 |
| Adelie | Biscoe | 41.6 | 18 | 192 | 3950 | male | 2008 |
| Adelie | Biscoe | 35.5 | 16.2 | 195 | 3350 | female | 2008 |
| Adelie | Biscoe | 41.1 | 19.1 | 188 | 4100 | male | 2008 |
| Adelie | Biscoe | 35 | 17.9 | 192 | 3725 | female | 2009 |
| Adelie | Biscoe | 41 | 20 | 203 | 4725 | male | 2009 |
| Adelie | Biscoe | 37.7 | 16 | 183 | 3075 | female | 2009 |
| Adelie | Biscoe | 37.8 | 20 | 190 | 4250 | male | 2009 |

| species | island | bill_len | bill_dep | flipper_len | body_mass | sex | year |
|---|---|---|---|---|---|---|---|
| Adelie | Biscoe | 37.9 | 18.6 | 193 | 2925 | female | 2009 |
| Adelie | Biscoe | 39.7 | 18.9 | 184 | 3550 | male | 2009 |
| Adelie | Biscoe | 38.6 | 17.2 | 199 | 3750 | female | 2009 |
| Adelie | Biscoe | 38.2 | 20 | 190 | 3900 | male | 2009 |
| Adelie | Biscoe | 38.1 | 17 | 181 | 3175 | female | 2009 |
| Adelie | Biscoe | 43.2 | 19 | 197 | 4775 | male | 2009 |
| Adelie | Biscoe | 38.1 | 16.5 | 198 | 3825 | female | 2009 |
| Adelie | Biscoe | 45.6 | 20.3 | 191 | 4600 | male | 2009 |
| Adelie | Biscoe | 39.7 | 17.7 | 193 | 3200 | female | 2009 |
| Adelie | Biscoe | 42.2 | 19.5 | 197 | 4275 | male | 2009 |
| Adelie | Biscoe | 39.6 | 20.7 | 191 | 3900 | female | 2009 |
| Adelie | Biscoe | 42.7 | 18.3 | 196 | 4075 | male | 2009 |
| Gentoo | Biscoe | 46.1 | 13.2 | 211 | 4500 | female | 2007 |
| Gentoo | Biscoe | 50 | 16.3 | 230 | 5700 | male | 2007 |
| Gentoo | Biscoe | 48.7 | 14.1 | 210 | 4450 | female | 2007 |
| Gentoo | Biscoe | 50 | 15.2 | 218 | 5700 | male | 2007 |
| Gentoo | Biscoe | 47.6 | 14.5 | 215 | 5400 | male | 2007 |
| Gentoo | Biscoe | 46.5 | 13.5 | 210 | 4550 | female | 2007 |

| species | island | bill_len | bill_dep | flipper_len | body_mass | sex | year |
|---|---|---|---|---|---|---|---|
| Gentoo | Biscoe | 45.4 | 14.6 | 211 | 4800 | female | 2007 |
| Gentoo | Biscoe | 46.7 | 15.3 | 219 | 5200 | male | 2007 |
| Gentoo | Biscoe | 43.3 | 13.4 | 209 | 4400 | female | 2007 |
| Gentoo | Biscoe | 46.8 | 15.4 | 215 | 5150 | male | 2007 |
| Gentoo | Biscoe | 40.9 | 13.7 | 214 | 4650 | female | 2007 |
| Gentoo | Biscoe | 49 | 16.1 | 216 | 5550 | male | 2007 |
| Gentoo | Biscoe | 45.5 | 13.7 | 214 | 4650 | female | 2007 |
| Gentoo | Biscoe | 48.4 | 14.6 | 213 | 5850 | male | 2007 |
| Gentoo | Biscoe | 45.8 | 14.6 | 210 | 4200 | female | 2007 |
| Gentoo | Biscoe | 49.3 | 15.7 | 217 | 5850 | male | 2007 |
| Gentoo | Biscoe | 42 | 13.5 | 210 | 4150 | female | 2007 |
| Gentoo | Biscoe | 49.2 | 15.2 | 221 | 6300 | male | 2007 |
| Gentoo | Biscoe | 46.2 | 14.5 | 209 | 4800 | female | 2007 |
| Gentoo | Biscoe | 48.7 | 15.1 | 222 | 5350 | male | 2007 |
| Gentoo | Biscoe | 50.2 | 14.3 | 218 | 5700 | male | 2007 |
| Gentoo | Biscoe | 45.1 | 14.5 | 215 | 5000 | female | 2007 |
| Gentoo | Biscoe | 46.5 | 14.5 | 213 | 4400 | female | 2007 |
| Gentoo | Biscoe | 46.3 | 15.8 | 215 | 5050 | male | 2007 |

| species | island | bill_len | bill_dep | flipper_len | body_mass | sex | year |
|---|---|---|---|---|---|---|---|
| Gentoo | Biscoe | 42.9 | 13.1 | 215 | 5000 | female | 2007 |
| Gentoo | Biscoe | 46.1 | 15.1 | 215 | 5100 | male | 2007 |
| Gentoo | Biscoe | 44.5 | 14.3 | 216 | 4100 | null | 2007 |
| Gentoo | Biscoe | 47.8 | 15 | 215 | 5650 | male | 2007 |
| Gentoo | Biscoe | 48.2 | 14.3 | 210 | 4600 | female | 2007 |
| Gentoo | Biscoe | 50 | 15.3 | 220 | 5550 | male | 2007 |
| Gentoo | Biscoe | 47.3 | 15.3 | 222 | 5250 | male | 2007 |
| Gentoo | Biscoe | 42.8 | 14.2 | 209 | 4700 | female | 2007 |
| Gentoo | Biscoe | 45.1 | 14.5 | 207 | 5050 | female | 2007 |
| Gentoo | Biscoe | 59.6 | 17 | 230 | 6050 | male | 2007 |
| Gentoo | Biscoe | 49.1 | 14.8 | 220 | 5150 | female | 2008 |
| Gentoo | Biscoe | 48.4 | 16.3 | 220 | 5400 | male | 2008 |
| Gentoo | Biscoe | 42.6 | 13.7 | 213 | 4950 | female | 2008 |
| Gentoo | Biscoe | 44.4 | 17.3 | 219 | 5250 | male | 2008 |
| Gentoo | Biscoe | 44 | 13.6 | 208 | 4350 | female | 2008 |
| Gentoo | Biscoe | 48.7 | 15.7 | 208 | 5350 | male | 2008 |
| Gentoo | Biscoe | 42.7 | 13.7 | 208 | 3950 | female | 2008 |
| Gentoo | Biscoe | 49.6 | 16 | 225 | 5700 | male | 2008 |

| species | island | bill_len | bill_dep | flipper_len | body_mass | sex | year |
|---|---|---|---|---|---|---|---|
| Gentoo | Biscoe | 45.3 | 13.7 | 210 | 4300 | female | 2008 |
| Gentoo | Biscoe | 49.6 | 15 | 216 | 4750 | male | 2008 |
| Gentoo | Biscoe | 50.5 | 15.9 | 222 | 5550 | male | 2008 |
| Gentoo | Biscoe | 43.6 | 13.9 | 217 | 4900 | female | 2008 |
| Gentoo | Biscoe | 45.5 | 13.9 | 210 | 4200 | female | 2008 |
| Gentoo | Biscoe | 50.5 | 15.9 | 225 | 5400 | male | 2008 |
| Gentoo | Biscoe | 44.9 | 13.3 | 213 | 5100 | female | 2008 |
| Gentoo | Biscoe | 45.2 | 15.8 | 215 | 5300 | male | 2008 |
| Gentoo | Biscoe | 46.6 | 14.2 | 210 | 4850 | female | 2008 |
| Gentoo | Biscoe | 48.5 | 14.1 | 220 | 5300 | male | 2008 |
| Gentoo | Biscoe | 45.1 | 14.4 | 210 | 4400 | female | 2008 |
| Gentoo | Biscoe | 50.1 | 15 | 225 | 5000 | male | 2008 |
| Gentoo | Biscoe | 46.5 | 14.4 | 217 | 4900 | female | 2008 |
| Gentoo | Biscoe | 45 | 15.4 | 220 | 5050 | male | 2008 |

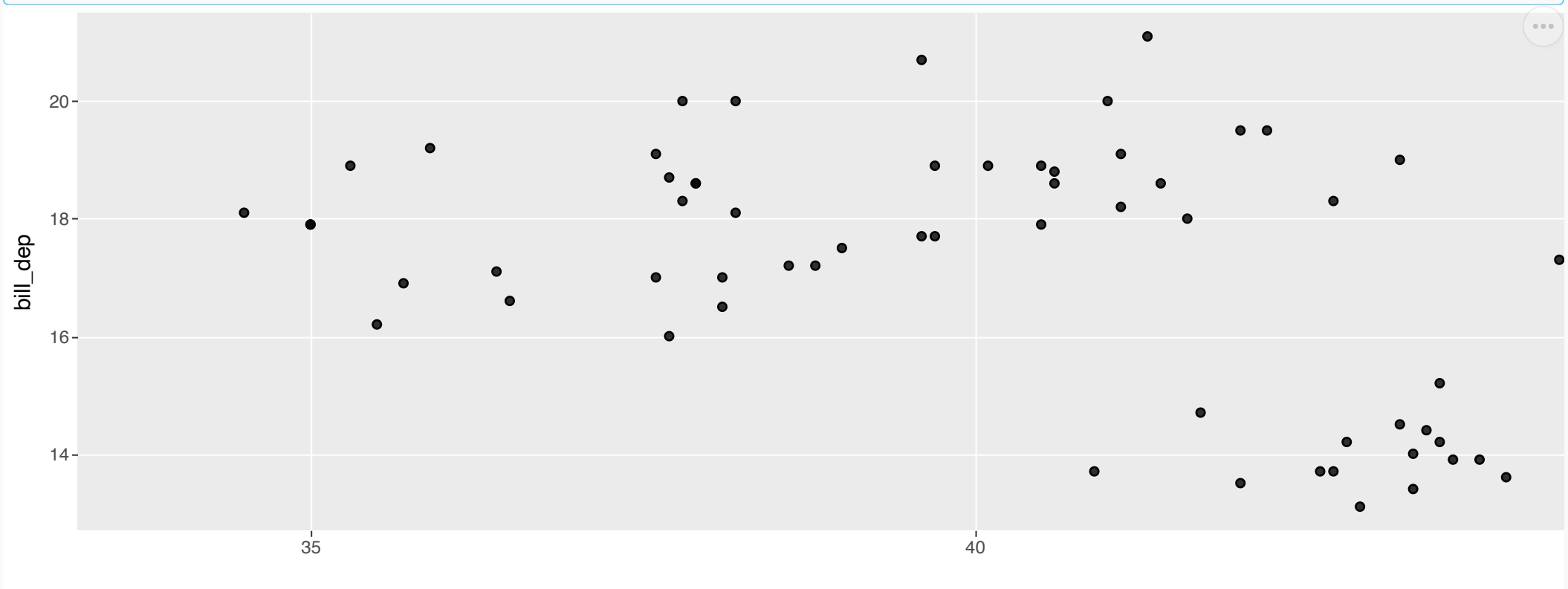
... 68 more rows

# A quick example

```
INSTALL ggsql FROM community;
LOAD ggsql;

SELECT * FROM 'assets/penguins.parquet'
  WHERE island == 'Biscoe'
```

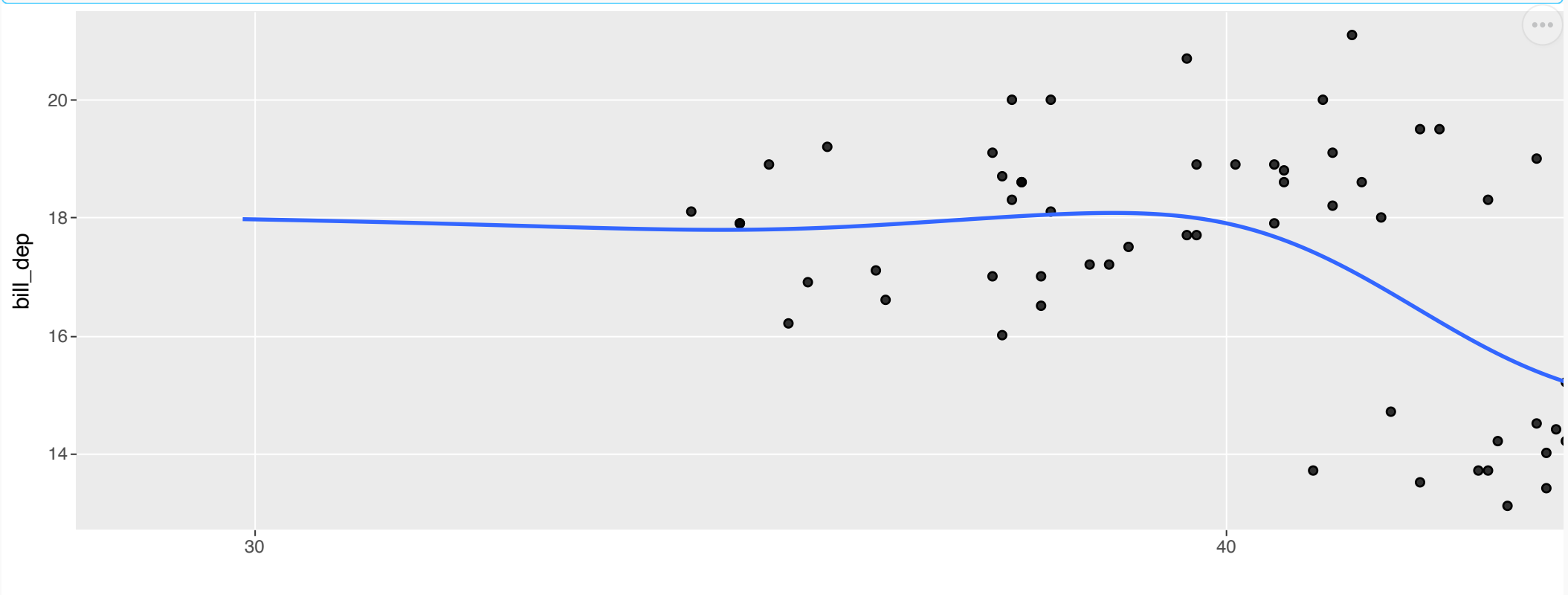
# A quick example

```
INSTALL ggsql FROM community;
LOAD ggsql;

SELECT * FROM 'assets/penguins.parquet'
  WHERE island == 'Biscoe'

VISUALISE bill_len AS x, bill_dep AS y
  DRAW point
```



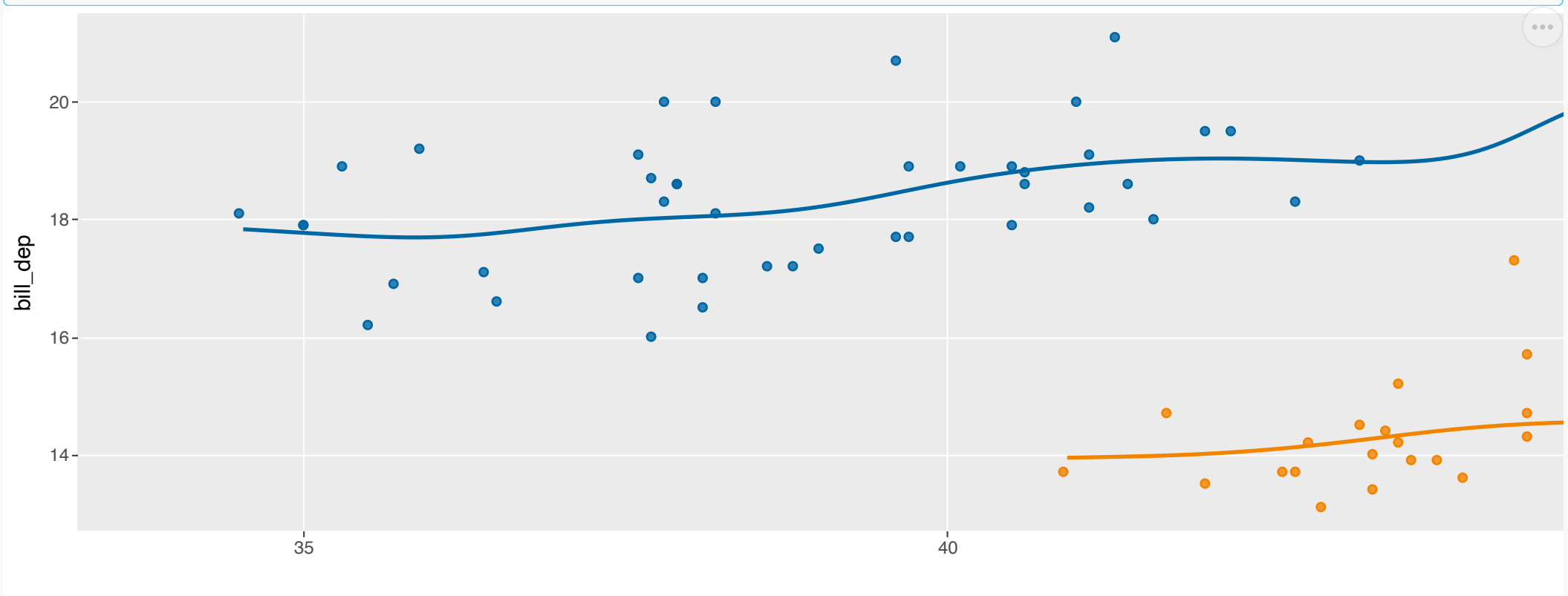
# A quick example

```
INSTALL ggsql FROM community;
LOAD ggsql;

SELECT * FROM 'assets/penguins.parquet'
  WHERE island == 'Biscoe'

VISUALISE bill_len AS x, bill_dep AS y
  DRAW point
  DRAW smooth
```



# A quick example

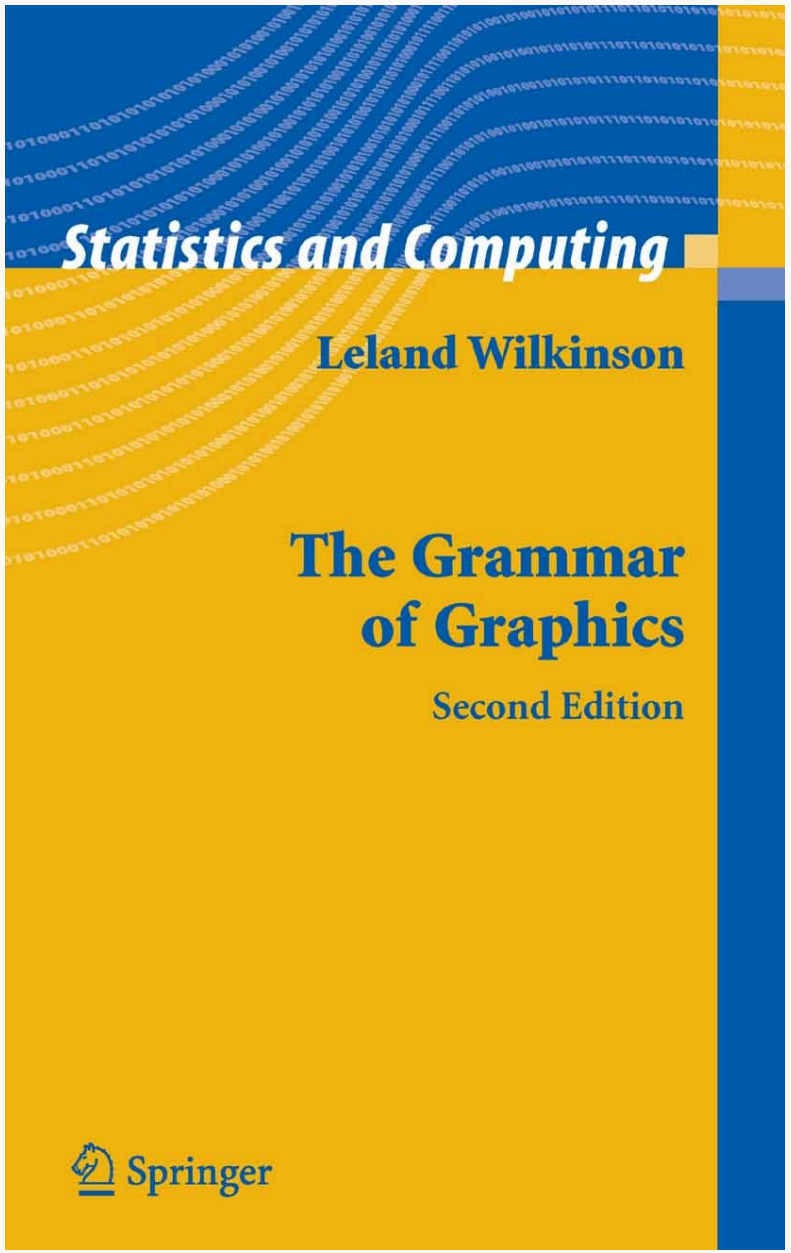
```
INSTALL ggsql FROM community;
LOAD ggsql;

SELECT * FROM 'assets/penguins.parquet'
  WHERE island == 'Biscoe'

VISUALISE bill_len AS x, bill_dep AS y, species AS colour
  DRAW point
  DRAW smooth
```



# ggplot2 is most popular visualisation method in R

- 18 years of ggplot2 development
- Constrained by backward compatibility
- ggsql is carte blanche for new ideas

# Grammar of Graphics



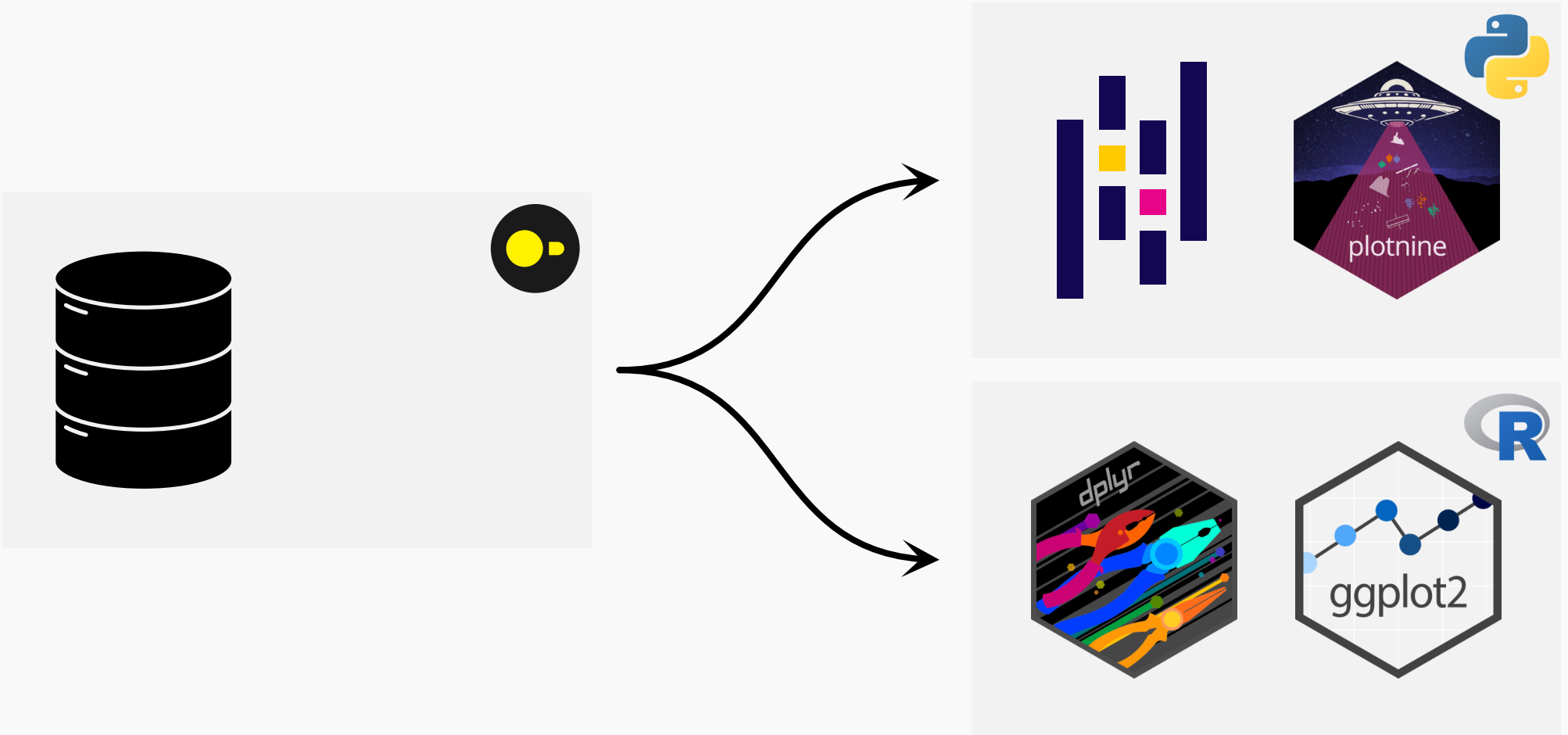

ggplot2

plotnine

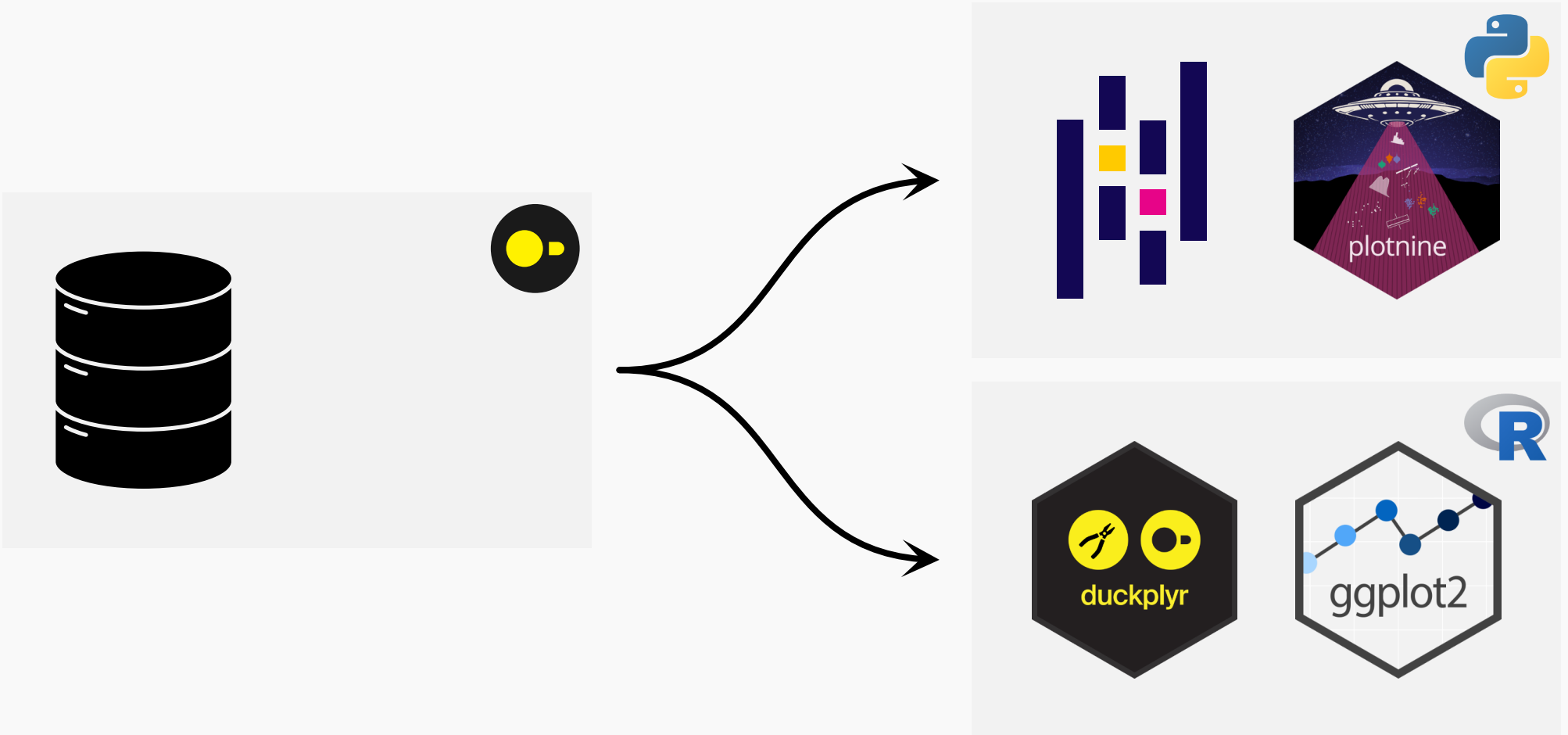
GADFLY

+ableau®

# Why are we building **this**?



# Why are we building **this**?



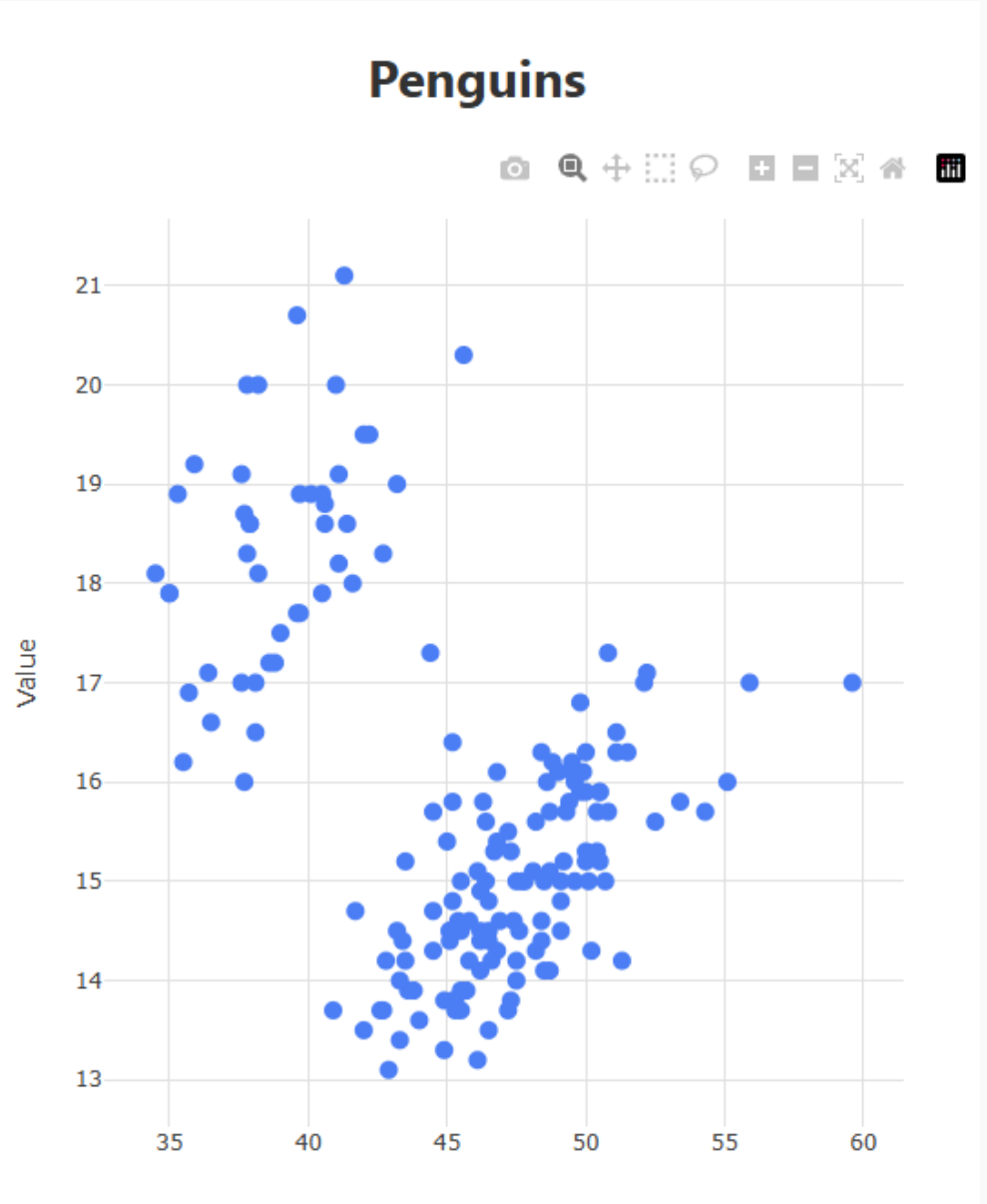
# Why are we building **this**?



# Why are we not building that?

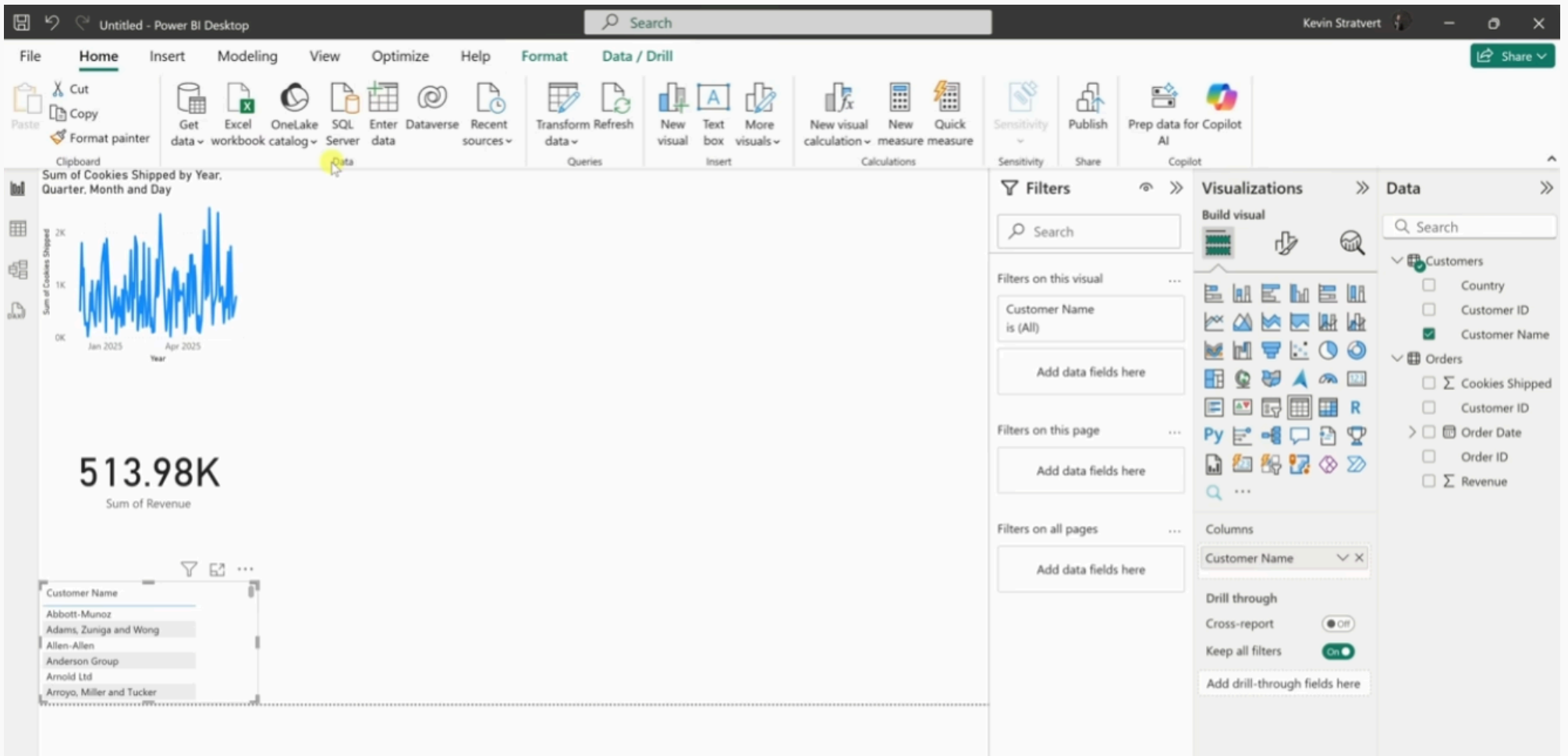
## A functional approach

```
INSTALL miniplot FROM community;
LOAD miniplot;

SELECT
scatter_chart(
  list(bill_len),
  list(bill_dep),
  'Penguins'
)
FROM 'assets/penguins.parquet'
WHERE island == 'Biscoe'
```



# Why are we not building that?

Closed source point and click BI tools



SQL

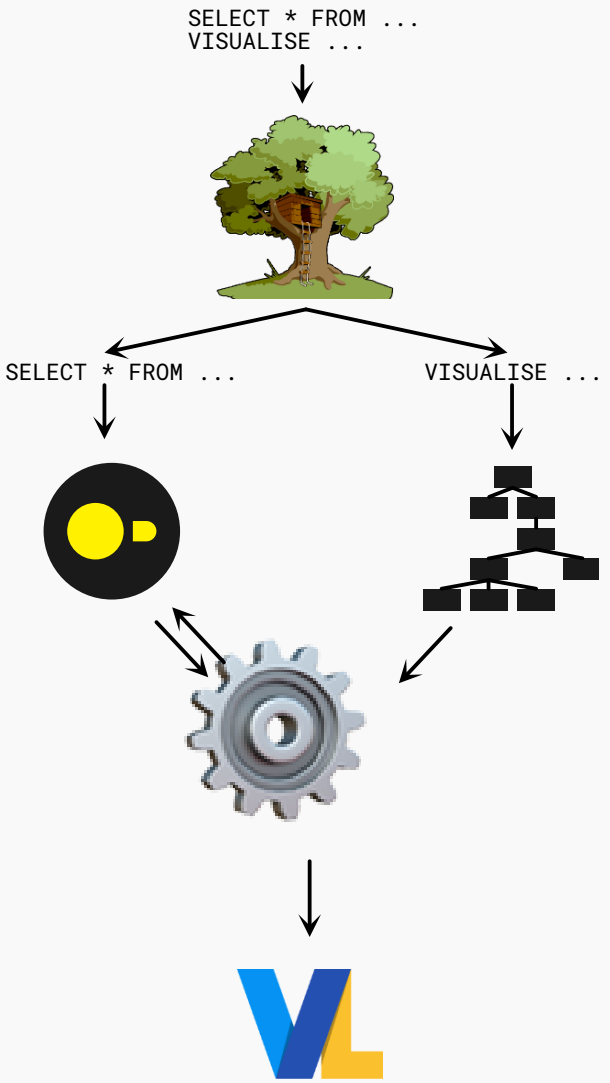
- Declarative
  - Information structures
  - ‘Give revenue grouped by month’
- Compositional
  - select, group by, where
- Portable execution
  - DuckDB, PostgreSQL, MySQL

Grammar of graphics

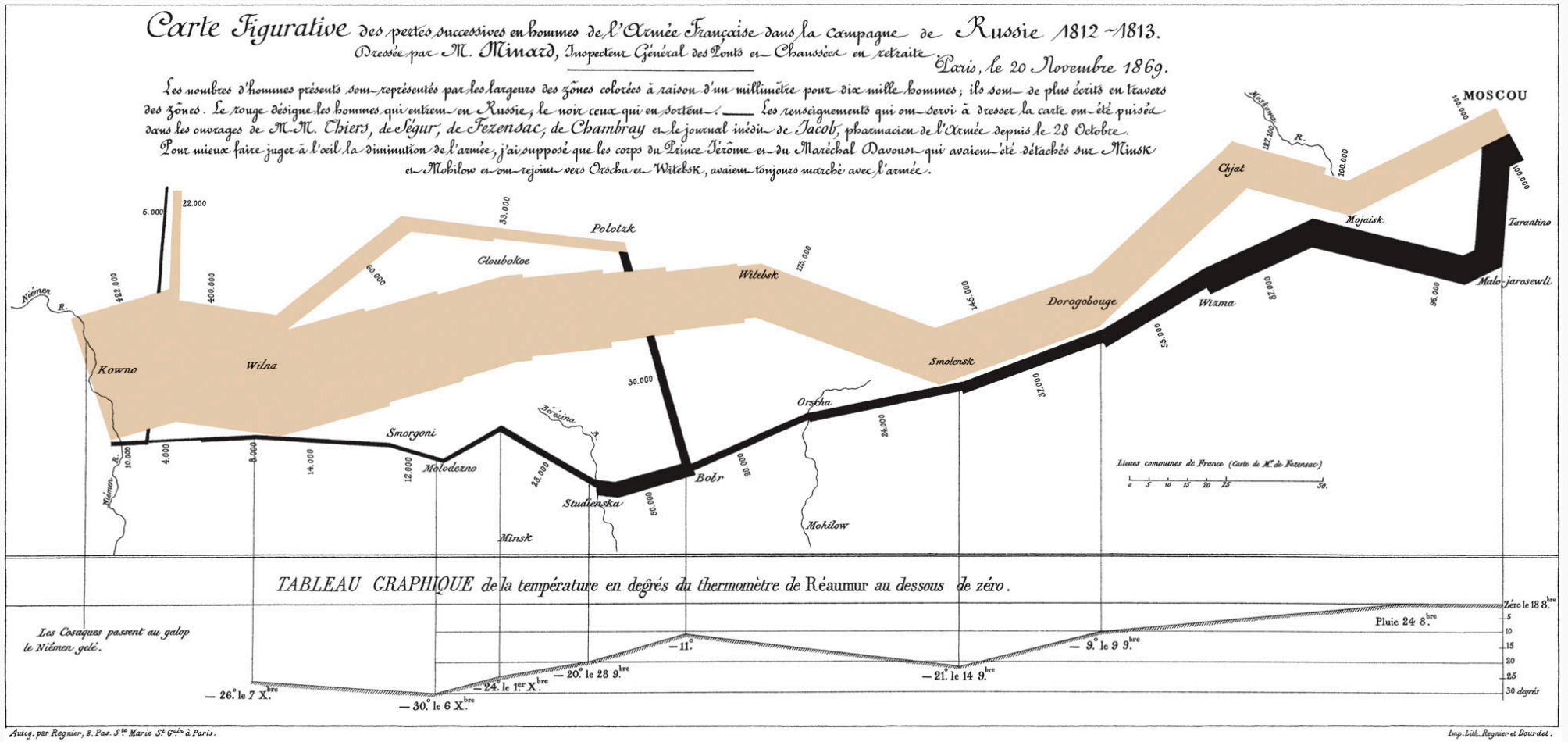
- Declarative
  - Information encodings
  - ‘Show revenue as bars by month’
- Compositional
  - layer, scales, facets
- Portable execution
  - ggplot2, Vega-Lite, plotnine

# How does it work?

- User query
  - Tree sitter
  - AST
- Reader
  - Query DB
  - Computed summaries
- Writer
  - JSON spec (Vega-Lite)
  - Future work



# Minard example

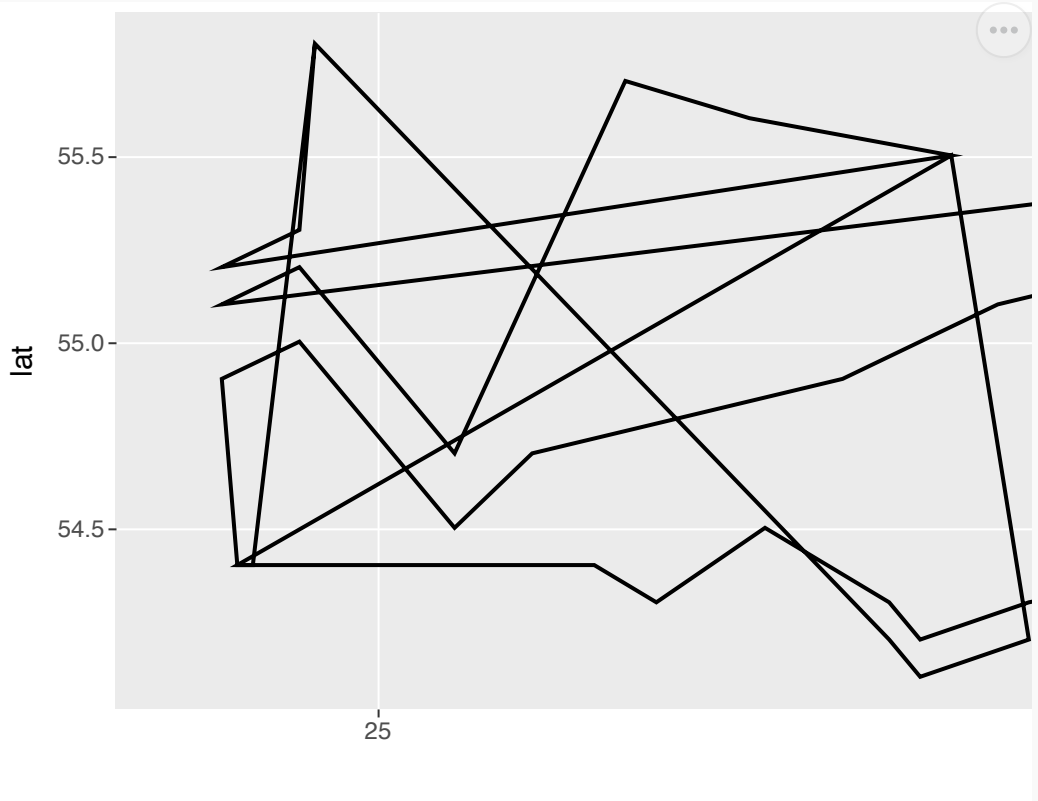


# Minard example

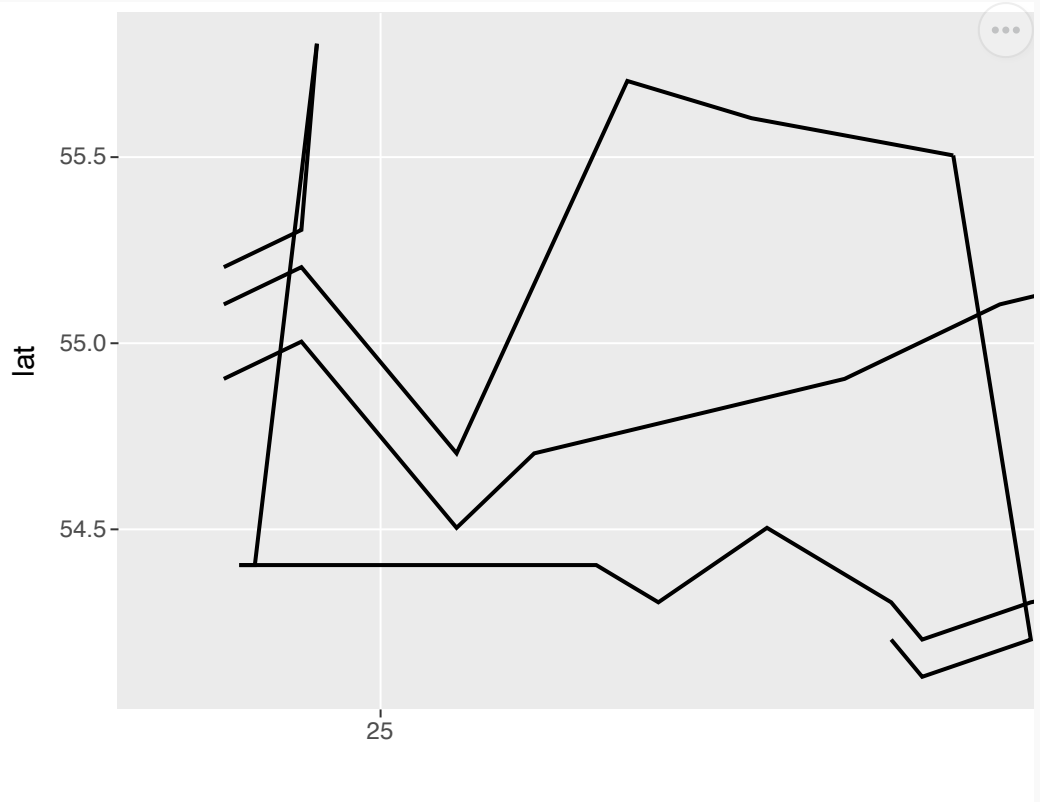
```
SELECT * FROM 'assets/minard_troops.csv'
LIMIT 5
```

| long | lat | survivors | direction | group |
|---|---|---|---|---|
| 37.7 | 55.7 | 100000 | R | 1 |
| 37.5 | 55.7 | 98000 | R | 1 |
| 37 | 55 | 97000 | R | 1 |
| 36.8 | 55 | 96000 | R | 1 |
| 35.4 | 55.3 | 87000 | R | 1 |

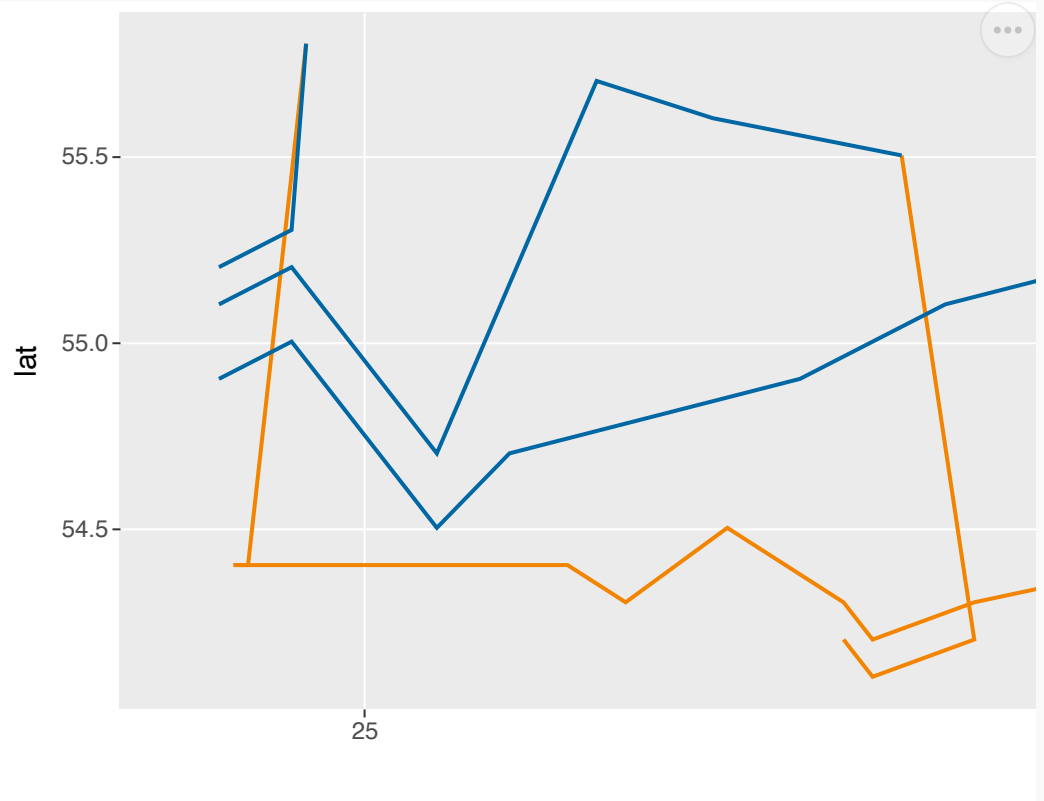
```
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW path
```



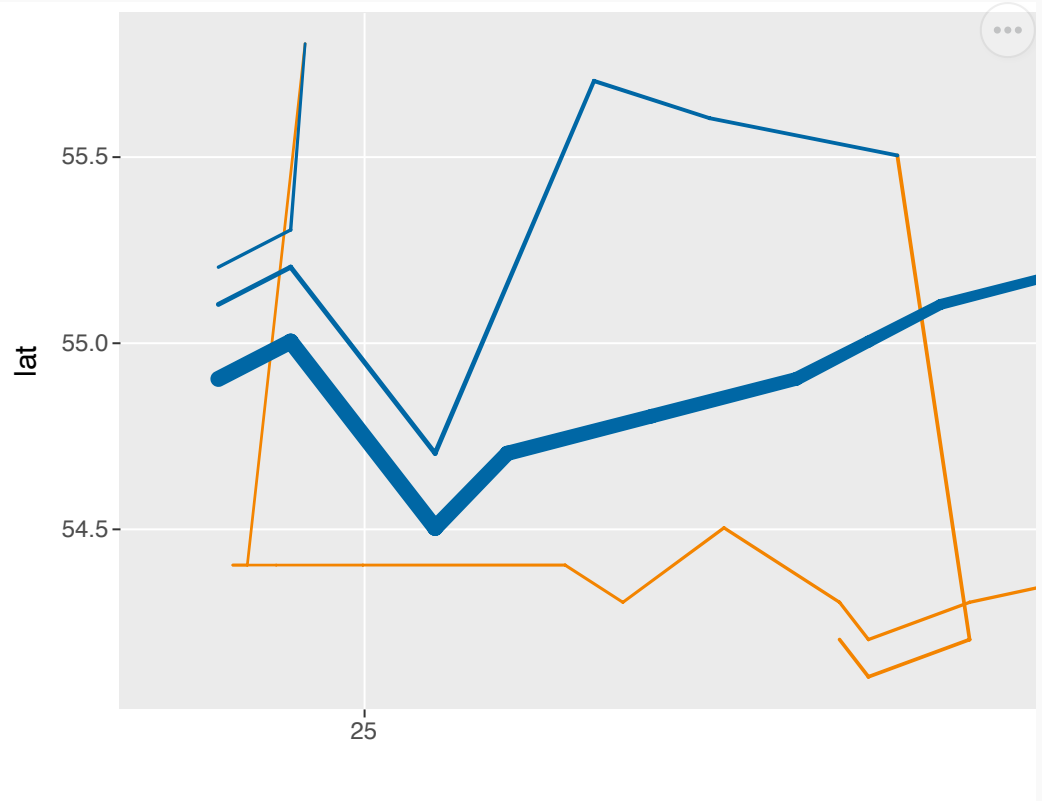
```
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW path
    PARTITION BY direction, group
```



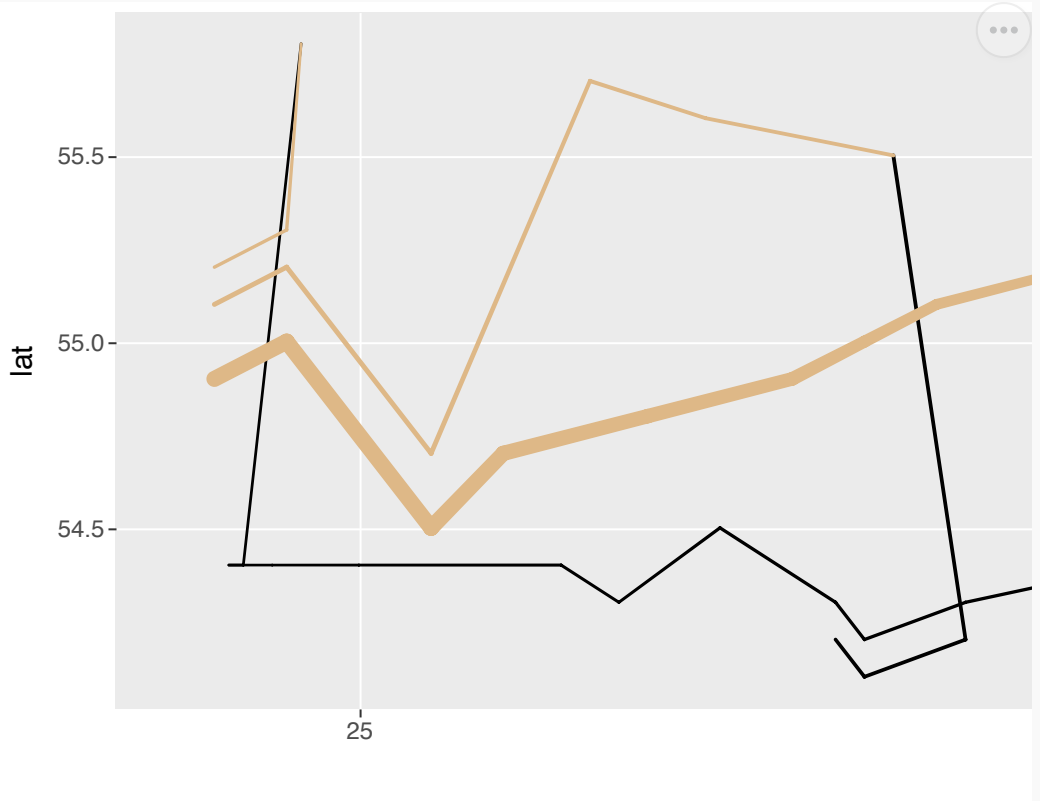
```
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW path
    MAPPING direction AS stroke
    PARTITION BY direction, group
```



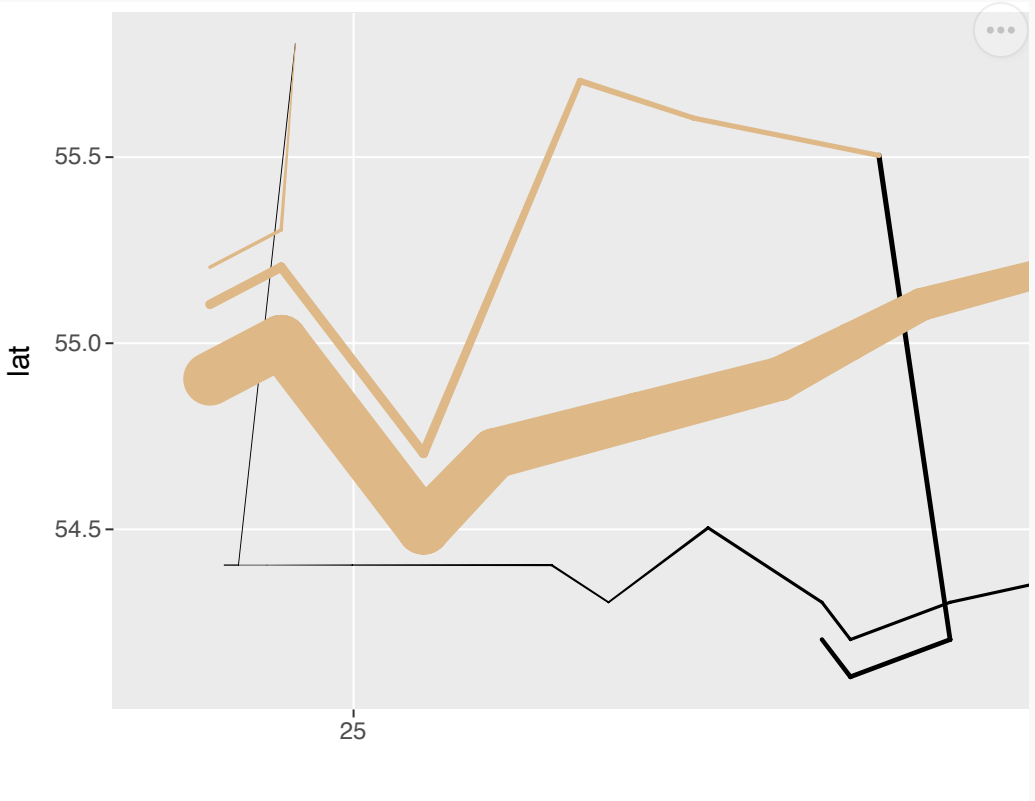
```
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
```



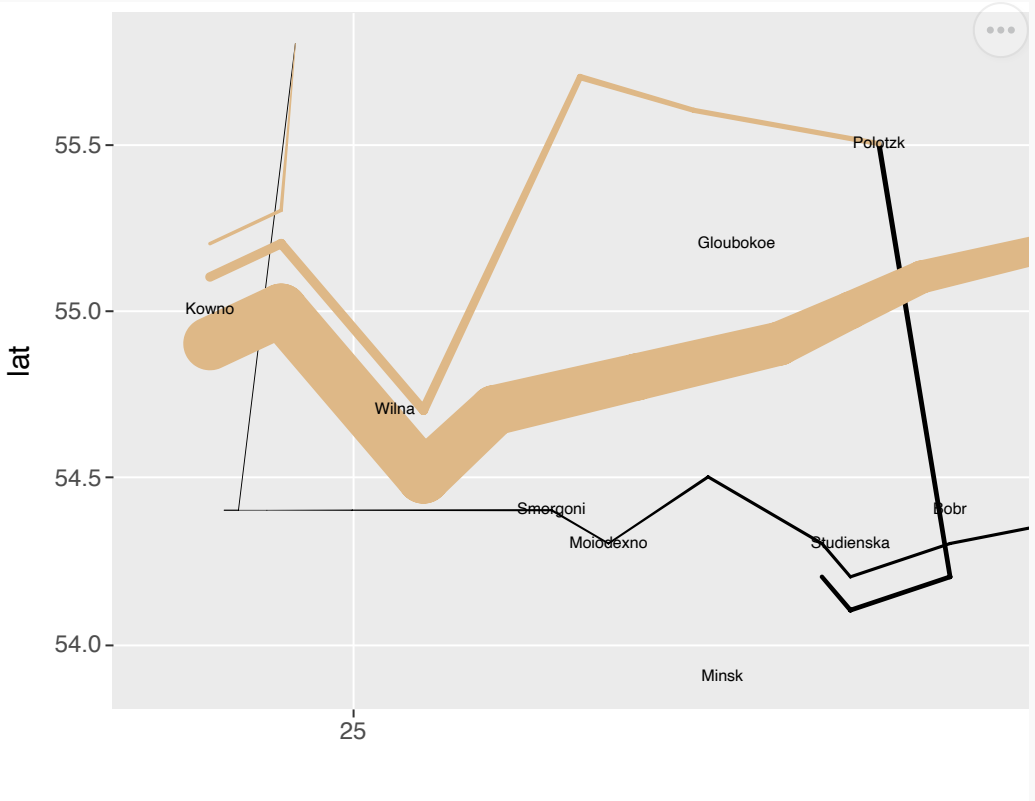
```
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
  SCALE stroke TO ('burlywood', 'black')
    RENAMING 'A' ⇒ 'Advance', 'R' ⇒ 'Retreat'
```



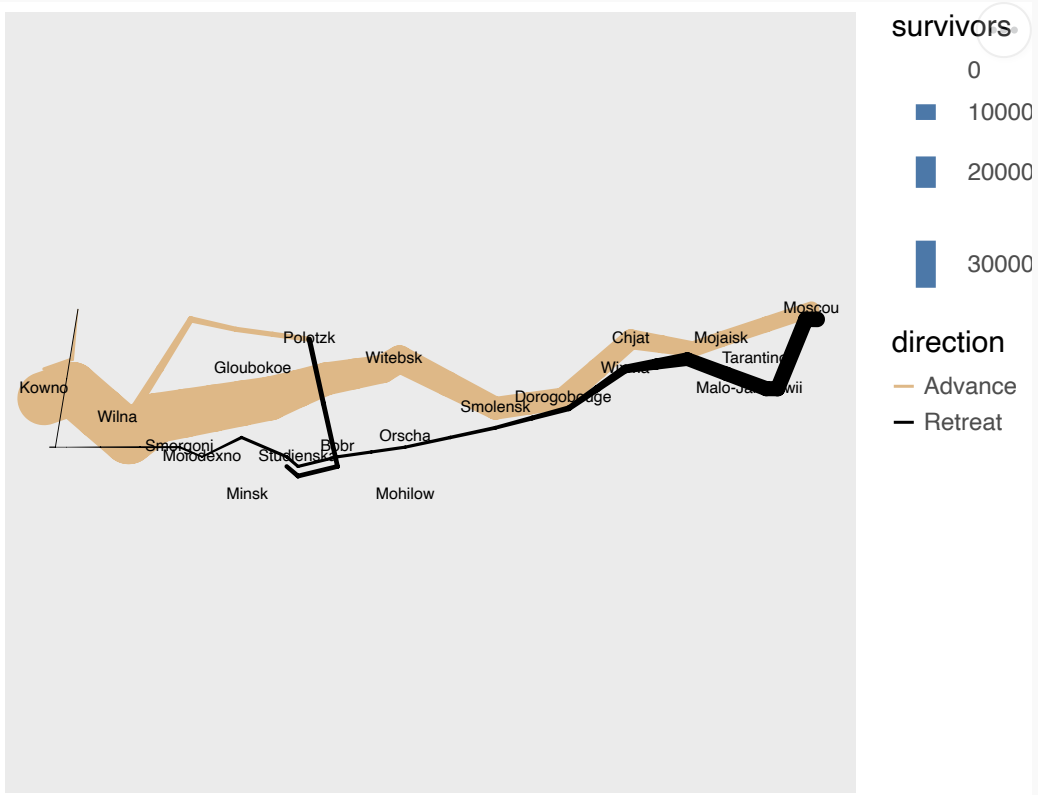
```
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
  SCALE stroke TO ('burlywood', 'black')
    RENAMING 'A' ⇒ 'Advance', 'R' ⇒ 'Retreat'
  SCALE linewidth FROM (0, null) TO (0, 20)
```



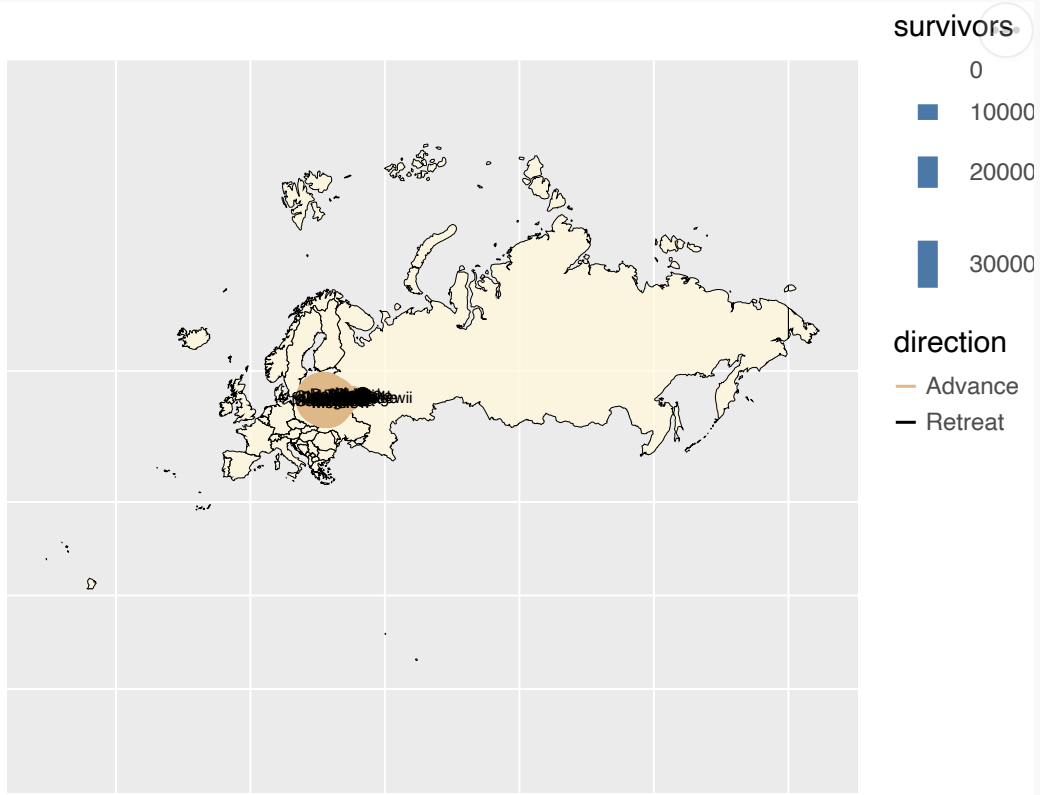
```
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
  DRAW text
    MAPPING city AS label FROM 'assets/minard_ci
    SETTING fontsize ⇒ 6
  SCALE stroke TO ('burlywood', 'black')
    RENAMING 'A' ⇒ 'Advance', 'R' ⇒ 'Retreat'
  SCALE linewidth FROM (0, null) TO (0, 20)
```



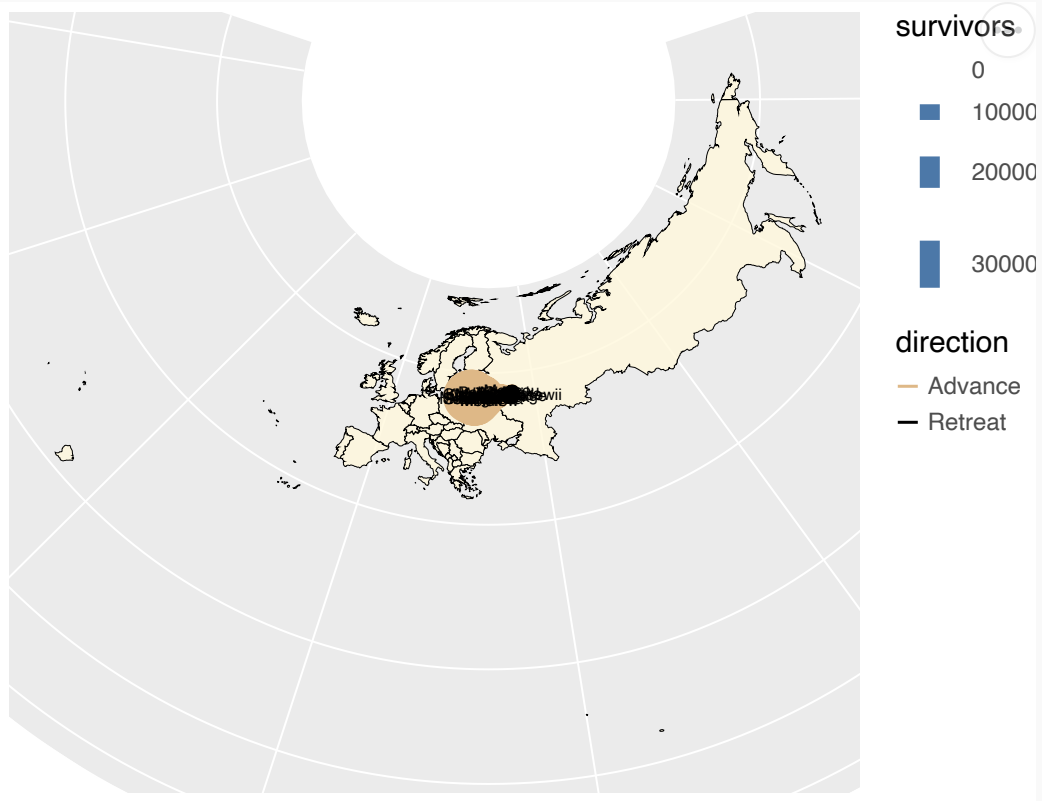
```
LOAD spatial;
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
  DRAW text
    MAPPING city AS label FROM 'assets/minard_ci
    SETTING fontsize ⇒ 6
  PROJECT x, y TO mercator
    SETTING origin ⇒ (30, 55)
  SCALE stroke TO ('burlywood', 'black')
    RENAMING 'A' ⇒ 'Advance', 'R' ⇒ 'Retreat'
  SCALE linewidth FROM (0, null) TO (0, 20)
```



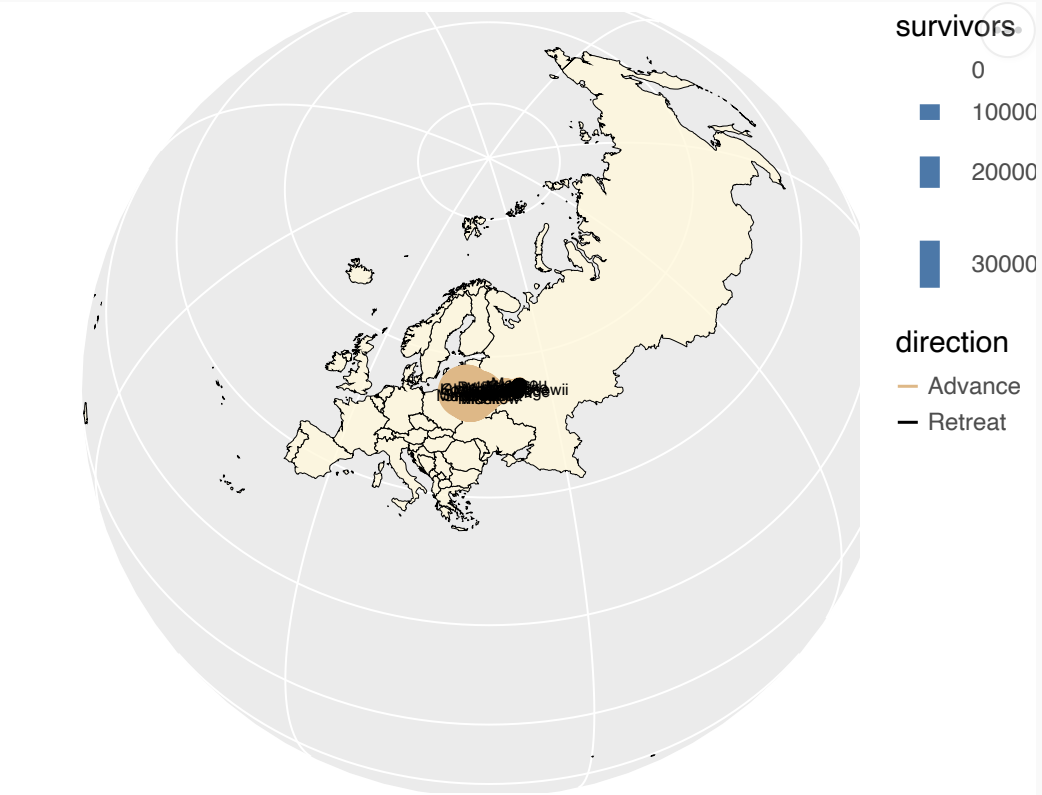
```
LOAD spatial;
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW spatial MAPPING FROM 'assets/countries.pa
    SETTING fill ⇒ 'cornsilk'
    FILTER continent == 'Europe'
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
  DRAW text
    MAPPING city AS label FROM 'assets/minard_ci
    SETTING fontsize ⇒ 6
  PROJECT x, y TO mercator
    SETTING origin ⇒ (30, 55)
  SCALE stroke TO ('burlywood', 'black')
    RENAMING 'A' ⇒ 'Advance', 'R' ⇒ 'Retreat'
  SCALE linewidth FROM (0, null) TO (0, 20)
```



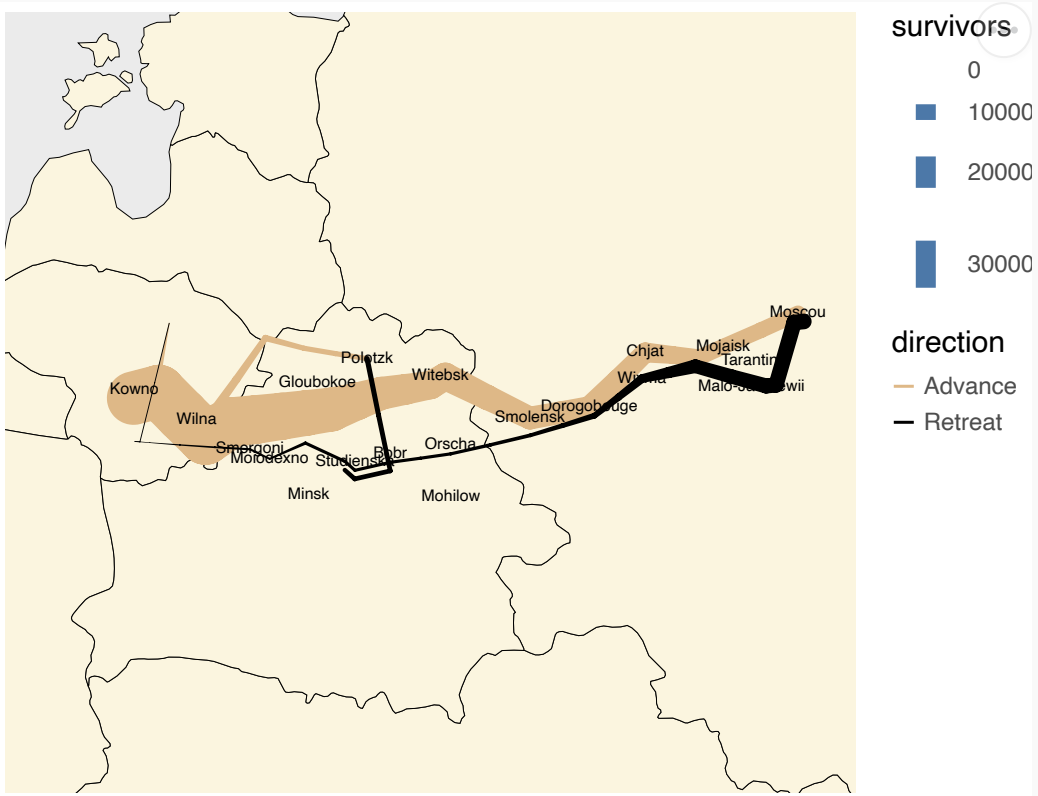
```
LOAD spatial;
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW spatial MAPPING FROM 'assets/countries.pa
    SETTING fill ⇒ 'cornsilk'
    FILTER continent == 'Europe'
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
  DRAW text
    MAPPING city AS label FROM 'assets/minard_ci
    SETTING fontsize ⇒ 6
  PROJECT x, y TO albers
    SETTING origin ⇒ (30, 55)
  SCALE stroke TO ('burlywood', 'black')
    RENAMING 'A' ⇒ 'Advance', 'R' ⇒ 'Retreat'
  SCALE linewidth FROM (0, null) TO (0, 20)
```



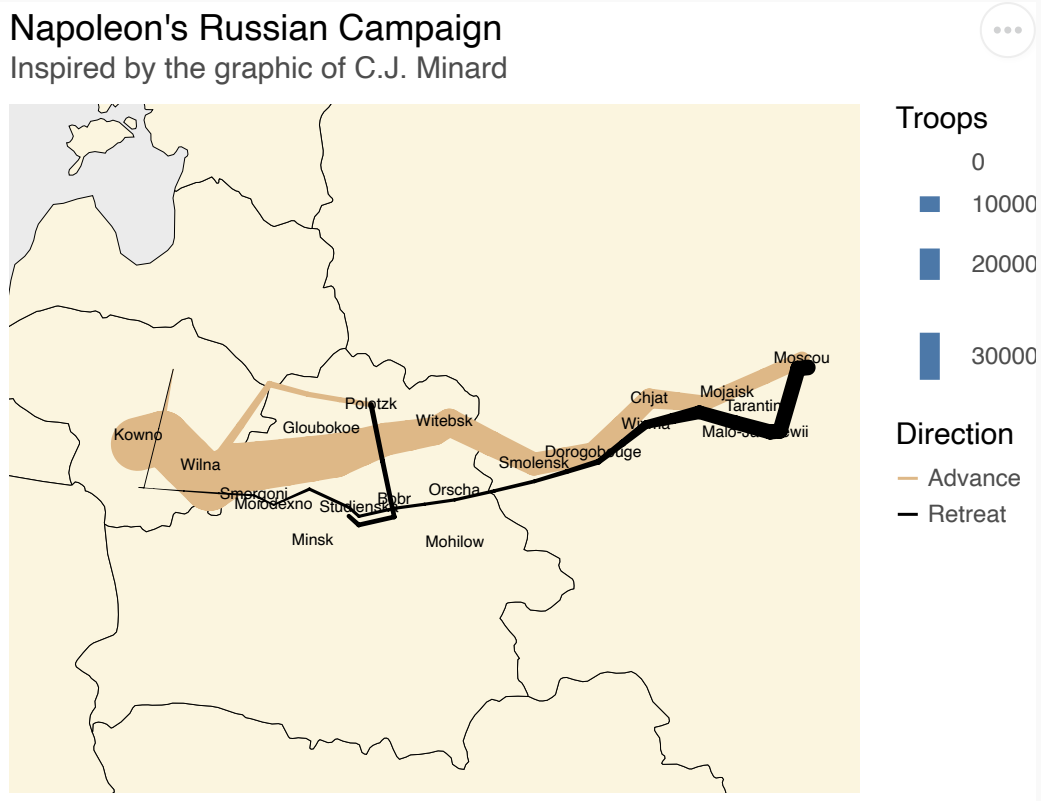
```
LOAD spatial;
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW spatial MAPPING FROM 'assets/countries.pa
    SETTING fill ⇒ 'cornsilk'
    FILTER continent == 'Europe'
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
  DRAW text
    MAPPING city AS label FROM 'assets/minard_ci
    SETTING fontsize ⇒ 6
  PROJECT x, y TO orthographic
    SETTING origin ⇒ (30, 55)
  SCALE stroke TO ('burlywood', 'black')
    RENAMING 'A' ⇒ 'Advance', 'R' ⇒ 'Retreat'
  SCALE linewidth FROM (0, null) TO (0, 20)
```



```
LOAD spatial;
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW spatial MAPPING FROM 'assets/countries.pa
    SETTING fill ⇒ 'cornsilk'
    FILTER continent == 'Europe'
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
  DRAW text
    MAPPING city AS label FROM 'assets/minard_ci
    SETTING fontsize ⇒ 6
  PROJECT x, y TO orthographic SETTING
    origin ⇒ (30, 55),
    bounds ⇒ (-500000, -150000, 500000, 150000)
  SCALE stroke TO ('burlywood', 'black')
    RENAMING 'A' ⇒ 'Advance', 'R' ⇒ 'Retreat'
  SCALE linewidth FROM (0, null) TO (0, 20)
```



```
LOAD spatial;
SELECT * FROM 'assets/minard_troops.csv'
VISUALISE long AS x, lat AS y
  DRAW spatial MAPPING FROM 'assets/countries.pa
    SETTING fill ⇒ 'cornsilk'
    FILTER continent == 'Europe'
  DRAW path
    MAPPING
      direction AS stroke,
      survivors AS linewidth
    PARTITION BY direction, group
  DRAW text
    MAPPING city AS label FROM 'assets/minard_ci
    SETTING fontsize ⇒ 6
  PROJECT x, y TO orthographic SETTING
    origin ⇒ (30, 55),
    bounds ⇒ (-500000, -150000, 500000, 150000)
  SCALE stroke TO ('burlywood', 'black')
    RENAMING 'A' ⇒ 'Advance', 'R' ⇒ 'Retreat'
  SCALE linewidth FROM (0, null) TO (0, 20)
  LABEL
    title ⇒ 'Napoleon\'s Russian Campaign',
```

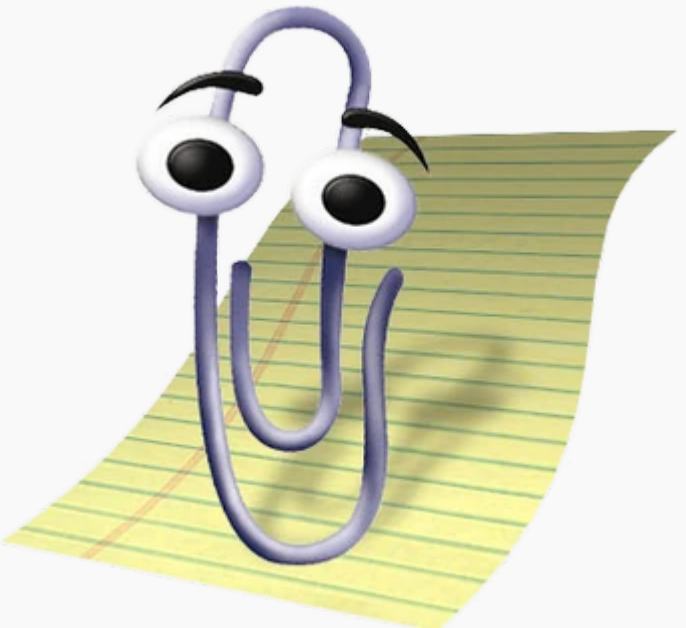


# Works wherever

- Distributed as standalone executable
- No bundling of R/Python
- Constrained execution
- Variety of integrations
  - DuckDB extension, CLI, Jupyter kernel

# LLM-tooling

- Assistants well-versed in SQL
  - Resembles structured natural language
  - Declarative
  - Small core grammar
  - Short context
- LLMs as interface for data visualisation
  - QueryChat
- ggsql skill



# Get yours today

Goal:
*Available wherever SQL is*

- ggsql.org interactive example
  - language support (Positron / VS Code)
  - kernel for Jupyter and Quarto
  - command line interface
  - R/Python wrappers
  - WebAssembly
- DuckDB extension:

```
INSTALL ggsql FROM community;
LOAD ggsql;
```

# In closing

- New SQL-based tool for data visualisation
- Based on declarative grammar of graphics
- DuckDB is a first-class citizen

# Thank you!


Posit PBC:

- Thomas Lin Pedersen
- George Stagg
- Hadley Wickham