Data support for Translation Imbalances in Wikipedia language editions

Notebook dependencies and setup

```
Mix.install([
  {:csv, "~> 3.0"},
  {:mediawiki_client, "~> 0.5.0"},
  {:kino_vega_lite, "~> 0.1.0"},
  {:vega_lite, "~> 0.1.0"}
])
```

Evaluated*

```
:ok
```

Introduction

The data here is support for an extended abstract presented at Wiki Workshop 2025, "Translation Imbalances Between Wikipedia Language Editions" . Although we originally did this work using private data sources, the limited results published in the paper can be recalculated from public sources and we attempt to do so below. Note that there may be slight differences due to the time ranges included.

This file is an Elixir livebook and it should be possible to run locally if you wish.

Fetch site statistics for all Wikipedia language editions

```
{:ok, all_sites} = Wiki.SiteMatrix.get_all()

wikipedias =
  Enum.filter(all_sites, fn site ->
    site.project == "wiki" && !site.closed && !site.private
  end)
```

Evaluated*

```
site_stats =
  wikipedias
  |> Enum.map(fn site ->
    {:ok, %{result: %{"query" => %{"statistics" => result}}}} =
      site
      |> Wiki.Action.new()
      |> Wiki.Action.get(action: :query, meta: :siteinfo, siprop: :statistics)

    {site.dbname, result}
  end)
  |> Enum.into(%{})
```

Evaluated*

Fetch Content Translation statistics

There used to be an API `contenttranslationstats` but this was disabled due to performance concerns (see task T392839). We're forced to use a stored response here, dated 2023-06-12.

In the future we should be able to refresh the data using sources being developed for a new public dashboard for Content Translation.

```
content_translation_pairs =
  File.stream!(Path.join(__DIR__, "content_translation_stats.csv"))
  |> CSV.decode!(headers: true)
  |> Enum.filter(&(&1["status"] == "published"))
```

Evaluated*

Aggregate by language

Here we collapse the raw data into a summary for each language, with the following aggregate calculations:

- Number of article pages in language edition
- Number of active editors for language
- Number of outgoing translations from language
- Number of incoming translations to language
- Translation hegemony (outgoing : incoming)

```elixir
lang_stats =
  wikipedias
  |> Enum.reject(&(&1.dbname == "simplewiki"))
  |> Enum.map(fn %{lang: lang, dbname: dbname} ->
    stats = %{
      article_count: site_stats[dbname]["articles"],
      active_editor_count: site_stats[dbname]["activeusers"],
      outgoing_translations:
        content_translation_pairs
        |> Enum.filter(&(&1["sourceLanguage"] == lang))
        |> Enum.sum_by(&String.to_integer(&1["count"])),
      incoming_translations:
        content_translation_pairs
        |> Enum.filter(&(&1["targetLanguage"] == lang))
        |> Enum.sum_by(&String.to_integer(&1["count"]))
    }

    stats =
      Map.put(
        stats,
        :translation_hegemony,
        case stats.incoming_translations do
          0 -> 0
          _ -> stats.outgoing_translations / stats.incoming_translations
        end
      )

    {lang, stats}
  end)
  |> Enum.into(%{})
```

Evaluated*

```elixir
flat_lang_stats =
  lang_stats
  |> Enum.map(fn {lang, stats} ->
    Map.put(stats, :lang, lang)
  end)
```

Evaluated*

Save the aggregated data back to disk so it can be used by other packages.

```elixir
flat_lang_stats
|> CSV.encode(
  headers: ~w(lang article_count active_editor_count outgoing_translations incoming_translations translation_hegemony
)
|> Enum.into(File.stream!(Path.join(__DIR__, "aggregate_s
```

Evaluated*

Scatterplots of translation hegemony

```elixir
alias VegaLite, as: Vl
```

Evaluated*

```
VegaLite
```

```elixir
zero_safe =
  flat_lang_stats
  |> Enum.reject(& &1.translation_hegemony == 0)
```

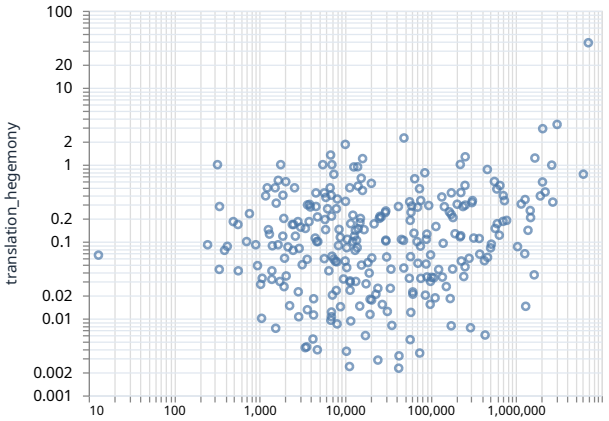
Evaluated*

```elixir
Vl.new(width: 400, height: 300, title: "Wiki article count vs. translation hegemony (log-log)")
|> Vl.data_from_values(zero_safe, only: [:article_count, :translation_hegemony])
|> Vl.mark(:point)
|> Vl.encode_field(:x, "article_count", type: :quantitative, scale: [type: :log])
|> Vl.encode_field(:y, "translation_hegemony", type: :qua
```

Evaluated*

```
cale: [type: :log])
```

Wiki article count vs. translation hegemony (log-log)



article_count

```
Vl.new(width: 400, height: 300, title: "Wiki active editor count vs. translation hegemony (log-log)"
)
|> Vl.data_from_values(zero_safe, only: [:active_editor_count, :translation_hegemony])
|> Vl.mark(:point)
|> Vl.encode_field(:x, "active_editor_count", type: :quantitative, scale: [type: :log])
|> Vl.encode_field(:y, "translation_hegemony", type: :qua Evaluated* , cale: [type: :log])
```

Wiki active editor count vs. translation hege- mony (log-log)