

Publishing Wikipedia usage data with strong privacy guarantees

Temilola Adeleye¹, Skye Berghel², Damien Desfontaines², Michael Hay²,
Isaac Johnson¹, Cléo Lemoisson¹, Ashwin Machanavajjhala², Tom Magerlein²,
Gabriele Modena¹, David Pujol², Daniel Simmons-Marengo², and Hal Tiedman¹

¹Wikimedia Foundation – htriedman@wikimedia.org

²Tumult Labs – science@tmlt.io

Abstract

For almost 20 years, the Wikimedia Foundation has been publishing statistics about how many people visited each Wikipedia page on each day. This data helps Wikipedia editors determine where to focus their efforts to improve the online encyclopedia, and enables academic research. In June 2023, the Wikimedia Foundation, helped by Tumult Labs, addressed a long-standing request from Wikipedia editors and academic researchers: it started publishing these statistics with finer granularity, including the country of origin in the daily counts of page views. This new data publication uses differential privacy to provide robust guarantees to people browsing or editing Wikipedia. This paper describes this data publication: its goals, the process followed from its inception to its deployment, the algorithms used to produce the data, and the outcomes of the data release.

1 Introduction

Wikipedia and other projects supported by the Wikimedia Foundation are among the most used online resources in the world, garnering hundreds of billions of visits each year from around the world. As such, the Foundation has access to terabytes of data about visits to a page on a Wikimedia project. This is called *pageview* data in this document.

The Foundation has been publishing statistics about this data for almost 20 years, through the *Pageview API* [17]. This data helps Wikipedia editors measure the impact of their work, and focus their efforts where they are most needed. Pageview data is also a rich resource for academic research: it has been used to better understand many topics, ranging from user behavior [14] and browsing patterns [15] to information dissemination [1], epidemiology [19], online harassment [33], and others. Over time, the Wikimedia Foundation received a number of requests to make these statistics more granular, and publish pageview counts *by country*, to make it even more useful to Wikipedia editors, and enable further academic research.

Addressing such requests for more granular data is aligned with the Foundation’s open access policy [12], which seeks to provide as much transparency as possible about how Wikimedia projects operate. However, the Foundation also considers privacy to be a key component of the free knowledge movement: there cannot be creation or consumption of free knowledge without a strong guarantee of privacy. These guarantees are expressed by the Foundation’s strict privacy policy [13] and data retention guidelines [6], which govern how the infrastructure underlying Wikipedia works. Concretely, people browsing Wikipedia may expect their behavior on the website to stay private: it is crucial to prevent motivated actors to combine this data with outside other data sources in order to spy on or persecute Wikipedia users for their view history, edit history, or other behavior. It is well-known that simply aggregating data is not, on its own, enough to prevent re-identification risk [34, 23, 22, 25], so publishing data with a finer geographic granularity warrants an approach with rock-solid privacy guarantees for Wikipedia users and editors.

Differential privacy [27] (DP) provides a way of easing this tension: it allows organizations to both lower and more fully understand the risks of releasing data. Therefore, the Wikimedia Foundation decided

to investigate the use of differential privacy to release daily pageview data, sliced by country. After an in-depth comparison of available open-source tools [5], the Wikimedia Foundation decided to use Tumult Analytics [30, 18] and started a collaboration with Tumult Labs to design and deploy a DP pipeline for this data release. The pipeline is now deployed, and the published data provides useful insights to anyone interested in better understanding Wikipedia usage.

This document describes this data release in more detail.

- In Section 2, we present the high-level workflow that we followed towards the deployment of a differentially private data release.
- In Section 3, we outline the problem statement and the success metrics for this data release.
- In Section 4, we describe the technical algorithms used for this data release.
- In Section 5, we summarize the results of this deployment.

2 High-level workflow for differential privacy deployments

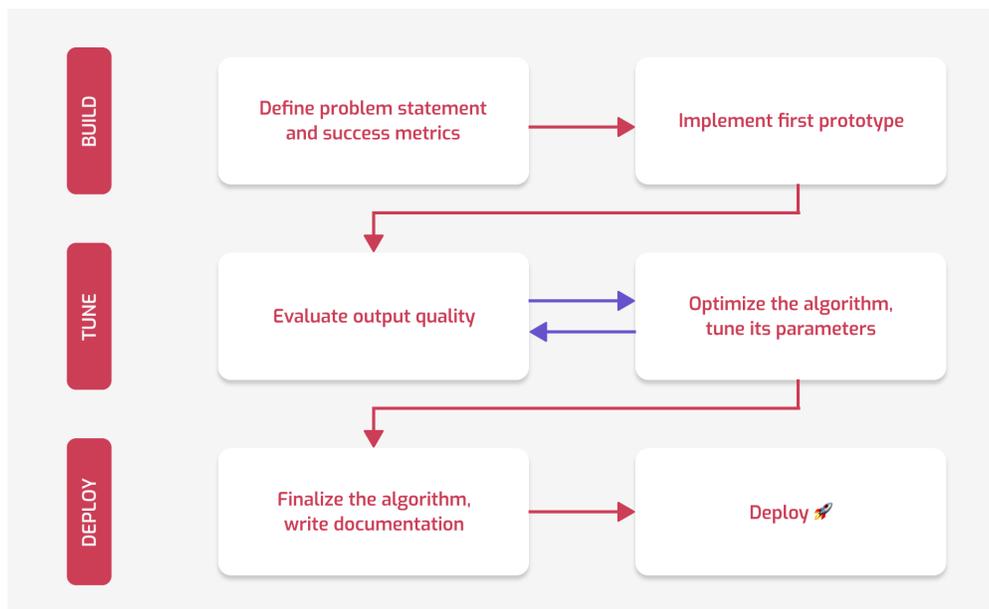


Figure 1: A standardized workflow for differentially private data releases.

The process to launch a DP data product follows a standard workflow, with three main stages: *Build*, *Tune*, and *Deploy*. The entire process is outlined in Figure 1; its three main stages are as follows.

1. In the initial Build stage, the goal is to gain a good understanding of the problem and its requirements, and implement a first-cut algorithm. There are two steps in this initial stage. First, we properly define the problem, and determine what success looks like for this project. This involves talking to stakeholders to understand what the data will be used for, and what accuracy metrics capture downstream use cases well. Second, we build a prototype mechanism. This is a first rough attempt at solving the data release problem, and it exposes the “levers” inherent to the project. Which choices did we have to make while building the prototype? Which of these choices can later be modified to pick different trade-offs between utility or privacy metrics?

2. Then, in the Tune step, we use these levers to experiment with different settings and optimize the algorithm. Using the success metrics defined in the previous step, we iteratively evaluate and adjust the algorithm, making changes until it produces data that is fit for use and satisfies the privacy requirements.
3. Finally, in the Deploy stage, we finalize the algorithm, obtain the necessary approvals to publish the data, write documentation about the data publication mechanism for future data users and pipeline maintainers, and deploy it in production.

In Section 3, we outline the output of the very first step: the definition of problem statement and its success metrics. Then, in Section 4, we will describe the output of the Tune stage: what does the final algorithm look like, after the multiple rounds of iteration on the initial prototype.

3 Problem statement and success metrics

In this Section, we describe the desired output data (Section 3.1), the schema and characteristics of the input data (Section 3.2), the privacy goals of this data release (Section 3.3), and the accuracy metrics used to quantify success (Section 3.4).

3.1 Desired output data

The pre-existing Pageview API publishes data about the number of times each Wikimedia page was visited during a given day. Each page is identified by two fields:

- its *project*, e.g. `fr.wikipedia` (the French-language version of Wikipedia), `zh.wikibooks` (the Chinese version of Wikibooks, an open-content textbook collection), `wikidata` (a central storage for structured data), etc.;
- its *page ID*, a numeric identifier uniquely identifying each page within a project.

Table 1 is a fictitious sample of the kind of data available via the Pageview API. For example, the first line indicates that there were 4217 visits to the page with ID 23110294 on the English version of Wikipedia on April 2nd, 2023.

Project	Page ID	Date	Count
en.wikipedia	23110294	2023-04-02	4217
fr.wikipedia	28278	2023-04-02	710
...

Table 1: A fictitious sample from the data made publicly available via the Pageview API.

The goal of this project is to publish more granular data, and also release daily pageview counts *per country*. A fictitious sample of the desired output data appears in Table 2. For example, the first line indicates that 92 of the previously-mentioned visits originated from Switzerland.

Project	Page ID	Date	Country	Count
en.wikipedia	23110294	2023-04-02	CH	92
fr.wikipedia	28278	2023-04-02	FR	101
...

Table 2: A fictitious sample from the data that we would like to publish as part of this project.

3.2 Input data

This project uses two input datasets: the *current pageviews dataset*, and the *historical pageviews dataset*.

Current pageviews dataset As users visit the site, their individual pageviews are recorded and stored in the current pageviews dataset. This dataset contains all pageviews across all Wikimedia projects for the last 90 days. Because of the Wikimedia Foundation’s commitment to minimal data retention, this data is only kept in this form for 90 days. Table 3 is a fictitious sample of the current pageviews dataset, showing only the columns of interest for this project: project, page ID, date and time, and country.

Project	Page ID	Date and Time	Country
en.wikipedia	23110294	2023-04-02 10:32:45	CH
fr.wikipedia	28278	2023-04-02 18:53:11	FR
...

Table 3: A fictitious sample of the columns of interest from the current pageviews dataset.

Note that in contrast to similar logging infrastructure for most websites, this data does not contain a persistent user identifier. Most visits to Wikimedia projects come from logged-out users, and the Wikimedia Foundation intentionally did not implement a user tracking mechanism which would provide a cookie ID and allow the Foundation’s systems to recognize whether two records came from the same user. This practice is good for data minimization, but it makes it more difficult to obtain user-level differential privacy guarantees, which requires bounding the number of contributions coming from the same user. We come back to this challenge in Section 4.1.1.

Historical pageviews dataset Past the initial 90-day retention period, pageviews are aggregated as hourly totals, broken down by project, page id, country, and a number of user characteristics. These aggregates are then stored in the historical pageviews dataset. Table 4 is a fictitious sample of the historical pageviews dataset, again showing only the columns of interest.

Project	Page ID	Date and Time	Country	Count
en.wikipedia	23110294	2023-04-02 10:00	CH	11
fr.wikipedia	28278	2023-04-02 18:00	FR	15
...

Table 4: A fictitious sample of the columns of interest from the pre-aggregated historical pageviews dataset.

This pre-aggregated data also poses a challenge for performing DP calculations: it is not possible to determine which contributions came from which users, and therefore to bound the number contributions coming from each user.

3.3 Privacy goal

When using differential privacy, one has to decide what to protect in the data; or, equivalently, what the definition of the neighboring databases should be. For long-running pipelines that publish data regularly over an unbounded time period, there are two aspects to this choice: what are the intervals of time considered as part of the unit of privacy, and what are we protecting in each of these intervals. Then, a follow-up question is the choice of privacy parameters: the numeric value of ϵ and δ .

Our goal is to publish data daily: it is natural to use a daily time period in the unit of privacy. This interval is consistent with almost all other long-running DP deployments, like Apple’s telemetry collection, or Google’s and Meta’s data releases related to the COVID-19 crisis. Other releases use a shorter period, like Microsoft’s telemetry in Windows. There is no overlap between days: the privacy parameters for each user-day are fixed and do not increase over time.

This choice of unit of privacy means that if a user were to regularly visit the same page from the same country across multiple devices (or clearing their cookies between each page visit) over a long period of time, this behavior could potentially be observed in the output data. Another caveat is that this data release surfaces group-level trends, like minority language group activity on Wikimedia projects within a country. These insights can be helpful (e.g. allow for dedicated support to that minority language group) but could also carry risks (e.g. by causing government persecution of this minority group). We mitigate these risks by choosing conservative privacy parameters, which translate to a reasonable level of protection over longer time periods, by holding off on releasing data for certain countries, and by only releasing aggregates that are above a certain threshold.

Protecting each individual Wikipedia user during each day is impossible to achieve entirely without a way to link people’s identities across records and devices. Because the Wikimedia Foundation does not have nor want the capability to link records in such a way, we instead attempt to protect the contribution of each *device* during each day. For the data based on the current pageviews dataset, we achieve this goal using *client-side contribution bounding*, as described in Section 4.1.1. For the data based on the historical pageviews dataset, we cannot bound user contributions. Instead, we choose to protect a fixed number of daily pageviews, denoted by m . This provides an equivalent level of protection to users who contribute fewer than m pageviews per day. Users who contribute more than m pageviews will incur a larger privacy loss, proportional to the amount by which their contributions exceed m . This number is set to 300 for data prior to February 8th, 2017, and to 30 for data between February 9th, 2017 to February 5th, 2023. Table 5 summarizes the privacy units chosen for this project.

Time period of the input data	Unit of privacy
July 1st, 2015 – February 8th, 2017	300 daily pageviews
February 9th, 2017 – February 5th, 2023	30 daily pageviews
February 6th, 2017 – present	one user-day

Table 5: A summary of the privacy units used in this project.

This difference in how many contributions we protect is due the fact that in February 2017, a change occurred to the way the input data was generated. Prior to February 8th, 2017, users who were editing a Wikimedia page and used the Web UI to preview their changes were recorded as one pageview each time the preview refreshed. This meant that during a lengthy editing session, an editor could plausibly rack up many pageviews on the same page. When combined with our inability to limit user contributions, this created a markedly different risk level before/after this date, that our historical pageviews algorithm had to address. Starting on February 9th, 2017, previews were no longer recorded as pageviews.

For privacy parameters, we use zero-concentrated DP [20] with $\rho = 0.015^1$ for the more recent data, and pure DP with $\epsilon = 1$ for the historical data. These values are generally considered to be conservative among differential privacy researchers and practitioners [31], and are lower than most practical DP deployments [24].

3.4 Accuracy metrics

We measure utility along three dimensions: the *relative error distribution*, the *drop rate*, and the *spurious rate*. Each of these metrics is computed using the *true data* as a baseline: the data that corresponds to simply running a group-by query (either counting the number of rows, for the current pageviews dataset, or summing the counts, for the historical pageviews dataset), without any contribution bounding, noise addition, nor suppression.

Relative error distribution We are releasing pageview counts, and the DP process will inject statistical noise into these counts. Thus, it is natural to want to measure how much noise is added to these counts. We measure accuracy according to *relative error*: the relative error of each noisy count \hat{c} is $|\hat{c}/c|$, where c is

¹Which is a strictly stronger guarantee than (ϵ, δ) -DP [26] with $\epsilon = 1$ and $\delta = 10^{-7}$.

the true count. Of course, we are releasing many counts, so we need to look at the *distribution* of relative error. More specifically, we look at the percentage of released counts having a relative error smaller than 10%, 25%, and 50%.

Drop rate The DP algorithm uses *suppression*: if a noisy count is lower than a given threshold, we remove it from the output data. To quantify the data loss due to this suppression step, it is natural to compute the *drop rate*: the percentage of counts that do not appear in the output, even though they were non-zero in the true data. In the true data, however, many of the counts are very low; suppressing such counts is not as bad as suppressing a popular page. Therefore, we compute the percentage of pages that were suppressed among pages whose true counts is larger than a fixed threshold t (the *drop rate above t*), as well as the percentage of pages that were suppressed among the top 1000 rows in the true data (the *top-1000 drop rate*).

Spurious rate Many page, project, and country combinations receive zero pageviews on any particular day. When noise is added to these zero counts, it is likely that they will end up with positive (though comparatively small) counts. We refer to these as *spurious* counts. Spurious counts can mislead data users by wrongly indicating that some combinations had activity. They also increase the size of the output dataset, which can pose a usability challenge. Therefore, we compute an additional metric: the *spurious rate*, which captures the ratio of spurious counts among all counts that appear in the output.

4 Technical description of the algorithms

In this Section, we describe the algorithms used to generate the differentially private data. For simplicity, we refer to a <page ID, project> pair as a *page*.

4.1 Current pageviews

For the data using the current pageviews dataset, we want to provide privacy guarantees that protect each user during each day. This requires bounding the maximum number of pageviews that each user can contribute during a single day. The typical way to perform such contribution bounding is to use a user identifier to sub-sample the number of contributions from each user, taking the first k records [29], or using reservoir sampling [32]. However, without a user identifier, we had to use a novel and alternative approach to this problem: *client-side filtering*.

4.1.1 Client-side filtering

Without user IDs, the server cannot know whether multiple contributions come from the same user, and perform contribution bounding to get user-level privacy guarantees. Instead, we add some logic to the client side. Each end-user device counts their number of contributions logged in each day, and sends each contribution along with a boolean flag, indicating whether this contribution should be used in the server-side DP computation. The criteria used for inclusion in the input to the DP algorithm is as follows: each day, we include the first 10 *unique* pageviews. This means that if a user visits the same page multiple times in a day, only the first visit will be counted. This also means that if a user visits more than 10 distinct pages in a day, all pageviews after the 10th visits will not be included.

Pseudocode for this client-side filtering step can be found in Algorithm 1. Note that this algorithm does not keep track of the raw page IDs in the client-side cookie. Instead, it uses a salted hash function [16] to remember which page IDs were already visited. This provides an additional level of protection against an attacker that would obtain access to this cookie.

Client-side filtering upholds the Wikimedia Foundation’s data minimization principle: only the absolute minimal information needed to perform the contribution bounding — a boolean value associated with each pageview to indicate whether it should be counted — is added to the logging infrastructure. Alternatives

Algorithm 1 Client-side filtering algorithm

Require: $P = p_1, p_2, \dots$: a stream of pageviews.

Require: H : a salted hash function

Require: k : the number of unique pageviews to include.

Ensure: The output is a stream of the same pageviews, each one annotated with a boolean indicating whether it should be included in the DP computation. This boolean is `true` iff the pageview comes from a page not output before, and

```
1:  $S \leftarrow \{\}$ 
2: for  $p$  in  $P$  do
3:   if  $|S| \geq k$  or  $H(p) \in S$  then
4:     Output  $\langle p, \text{false} \rangle$ 
5:   else
6:      $S \leftarrow S \cup H(p)$ 
7:     Output  $\langle p, \text{true} \rangle$ 
8:   end if
9: end for
```

such as using identifiers or a counter that increments for each contribution would have required sending more data to the server, and increase fingerprinting risk.

4.1.2 Server-side algorithm

Once each pageview was annotated by the client-side filtering algorithm, it is used as input in a server-side differentially private algorithm. This algorithm, run daily on the data from the previous day, has three stages.

1. First, we collect the list of $\langle \text{page}, \text{country} \rangle$ tuples to aggregate over.
2. Second, we count the number of pageviews in each group, and we add noise to each count.
3. Finally, we suppress low counts, and publish the data.

The list of all possible tuples is, in theory, known in advance: the list of Wikimedia pages and countries are both public information. However, the majority of $\langle \text{page}, \text{country} \rangle$ combinations do not appear in the input data: including all of them would be inefficient and lead to increased spurious data. Instead, we use existing public data to only include a small fraction of these possible counts. On each day, we list all Wikimedia pages with more than t global pageviews, according to the existing Pageview API, where t is an arbitrary ingestion threshold. Then, we take the cross-product between these pages and the list of countries² to create the groups.

The second step uses the Gaussian mechanism [28] to add noise to counts. This provides two advantages. First, because each user can contribute to at most 10 *different* $\langle \text{page}, \text{country} \rangle$ tuples, but only once to each, we get a tighter L_2 sensitivity bound (\sqrt{k}) than if we had used L_1 sensitivity (k): this allows us to add less noise. Second, because the tails of the Gaussian noise distribution decay very fast, this makes the thresholding step more efficient in preventing zero counts from appearing in the output, keeping the spurious rate to acceptably low levels. We quantify the privacy guarantees of the Gaussian mechanism using zero-concentrated DP [20] (zCDP).

The third step is straightforward: all counts below a threshold τ are removed from the output. This step is necessary because the first step produces many $\langle \text{page}, \text{country} \rangle$ tuples for which the non-noisy user count is very low or even 0. Such counts lead to unacceptable high relative error and spurious rate. Conversations with data users showed that these made the output dataset hard to use, and that users were most interested

²This list is based on [7]; excluding countries identified by the Wikimedia Foundation as potentially dangerous for journalists or internet freedom [3].

in the most-viewed pages, rather than the long tail of pages with few views. Suppressing counts below a fixed and configurable threshold τ addresses this problem, at the cost of a non-zero drop rate.

The mechanism is presented in Algorithm 2; in this algorithm, $\mathcal{N}(0, \sigma^2)$ denotes a random sample from a normal distribution of mean 0 and variance σ^2 . Step 1 uses only public data, Step 2 provides ρ -zCDP [20], and Step 3 is a post-processing step: the full algorithm satisfies ρ -zCDP.

Algorithm 2 Server-side algorithm for the current pageviews

Require: t : an ingestion threshold.

Require: τ : a suppression threshold.

Require: ρ : a privacy parameter for zCDP.

Require: $P = \langle p_1, b_1 \rangle, \langle p_2, b_2 \rangle, \dots$: a private dataset of annotated pageviews, such each user is at most associated with k unique pageviews $\langle p_i, b_i \rangle$ where $b_i = \text{true}$, and all of them have distinct p_i .

Require: $P_{\text{daily}} = \langle p_1, n_1 \rangle, \langle p_2, n_2 \rangle, \dots$: a public dataset listing the global number of pageviews for each page.

Require: C : a pre-defined list of countries.

Step 1: Collecting aggregation groups

```

1:  $G \leftarrow \{\}$ 
2: for  $\langle p, n \rangle$  in  $P_{\text{daily}}$  do
3:   if  $n \geq t$  then
4:     for  $c$  in  $C$  do
5:        $G \leftarrow G \cup \langle p, c \rangle$ 
6:     end for
7:   end if
8: end for

```

Step 2: Computing noisy counts

```

9:  $\sigma \leftarrow \sqrt{\frac{k}{2\rho}}$ 
10:  $O \leftarrow \{\}$ 
11: for  $g$  in  $G$  do
12:    $c \leftarrow |\{p \in P \mid p = g\}|$ 
13:    $\hat{c} \leftarrow c + \mathcal{N}(0, \sigma^2)$ 
14:    $O \leftarrow O \cup \langle g, \hat{c} \rangle$ 
15: end for

```

Step 3: Suppressing low counts

```

16: for  $\langle g, \hat{c} \rangle$  in  $G$  do
17:   if  $\hat{c} < \tau$  then
18:      $O \leftarrow O \setminus \langle g, \hat{c} \rangle$ 
19:   end if
20: end for
21: return  $O$ 

```

We use $k = 10$ as a per-user daily contribution bound, $t = 150$ as an ingestion threshold, and $\tau = 90$ as a suppression threshold. These values were chosen after extensive experimentation, for input dataset completeness and to optimize the utility metrics described in Section 3.4.

To select these algorithmic parameters, we computed metrics using the true data. Such metrics are, in principle, sensitive, and the parameters themselves are not differentially private. To mitigate the privacy risk from this tuning process, we kept fine-grained utility metrics confidential throughout the tuning process, minimizing data leakage. In addition to this consideration, we only publicly communicate approximate values of global utility metrics and the algorithmic parameters obtained from this tuning process.

Regardless, this remains a valid critique, and we would appreciate further research into the privacy loss entailed by confidentially tuning on sensitive metrics.

4.2 Historical pageviews

To compute differentially private counts using the historical pageview dataset as input data, we follow a similar process, with one key difference: since the data is pre-aggregated, it is impossible to perform per-user contribution bounding. Therefore, we do not use a client-side filtering step, and instead, use a different unit of privacy, as described in Section 3.3. We also have to sum the Count column of the pre-aggregated data, rather than simply counting the number of rows in each group. Another difference is the use of Laplace noise instead of Gaussian noise, motivated by the fact that we only have a bound on the L_1 sensitivity of the aggregation, and not L_2 like with the current pageviews data. The full process is otherwise similar to the previous one.

1. First, we collect the list of $\langle \text{page}, \text{country} \rangle$ tuples to aggregate over.
2. Second, we sum the pageview counts in each group, and we add Laplace noise to each sum.
3. Finally, we suppress low sums, and publish the data.

The full algorithm is provided as Algorithm 3; there, $\text{Lap}(0, \lambda)$ denotes a random sample from the Laplace distribution of mean 0 and scale λ . Its privacy analysis is straightforward: Step 1 uses only public data, Step 2 provides ϵ -DP guarantees [27], and Step 3 is a post-processing step, so the full algorithm satisfies ϵ -DP.

As mentioned in Section 3.3, we use $m = 300$ for the 2015–2017 data, and $m = 30$ for the 2017–2023 data. For the 2015–2017 data, we use $t = 150$ as ingestion threshold and $\tau = 3500$ as suppression threshold. For the 2017–2023 data, we use $t = 150$ as ingestion threshold and $\tau = 450$ as suppression threshold. These values were chosen to optimize the global utility metrics described in Section 3.4.

4.3 Implementation

The algorithms were implemented and deployed using Tumult Analytics [30, 18], a framework chosen for its robustness, production-readiness, compatibility with Wikimedia’s compute infrastructure, and support for advanced features like zCDP-based privacy accounting [5]. This incurs very slight differences in the mechanisms used: on integer-valued data, Tumult Analytics uses a two-sided geometric distribution instead of Laplace noise, and a discrete version of the Gaussian mechanism [21]. The data release based on the current input data required implementing a new notion of neighboring relation in the framework: rather than protecting a fixed number of rows, or an arbitrary number of rows associated with a single user identifier, it protects a fixed number of rows *associated with different aggregation groups*. This was made easier by the extensibility of the underlying framework, Tumult Core.

5 Outcomes

The deployment of this differentially private data publication project is now allowing the Wikimedia Foundation to release a much larger and much richer dataset about user visits to Wikimedia projects. The magnitude of this increase in published pageview data is summarized in Table 6.

More than 2,000 days of historical data from 2015 to 2021 were not previously published. The use of differential privacy in this project allowed the Wikimedia Foundation to release more than 135 million statistics about this data, encompassing 325 billion pageviews.

The output data had acceptable quality according to our success metrics.

- For the data based on the current pageviews dataset, more than 95% of the counts has a relative error below 50%, the drop rate above 150 is below 0.1%, the global spurious rate is below 0.01%, and below 3% for all but 3 countries.
- For the 2017–2023 data, the median top-1000 drop rate is below 8%, the drop rate above 450 is below 3%, and the global spurious rate is below 0.1%.

Algorithm 3 Algorithm for the historical pageviews

Require: m : the number of pageviews protected each day.

Require: t : an ingestion threshold.

Require: τ : a suppression threshold.

Require: ϵ : a privacy parameter for DP.

Require: $P_{hourly} = \langle p_1, c_1 \rangle, \langle p_2, c_2 \rangle, \dots$: a private dataset listing pre-aggregated hourly pageview counts.

Require: $P_{daily} = \langle p_1, n_1 \rangle, \langle p_2, n_2 \rangle, \dots$: a public dataset listing the global number of pageviews for each page.

Require: C : a pre-defined list of countries.

Require: t : A minimum pageview threshold for including pages in the output.

Step 1: Collecting aggregation groups

```
1:  $G \leftarrow \{\}$ 
2: for  $\langle p, n \rangle$  in  $P_{daily}$  do
3:   if  $n \geq t$  then
4:     for  $c$  in  $C$  do
5:        $G \leftarrow G \cup \langle p, c \rangle$ 
6:     end for
7:   end if
8: end for
```

Step 2: Computing noisy sums

```
9:  $\lambda \leftarrow \frac{m}{\epsilon}$ 
10:  $O \leftarrow \{\}$ 
11: for  $g$  in  $G$  do
12:    $s \leftarrow \sum_{\langle p, c \rangle \in P_{hourly} \text{ where } p=g} c$ 
13:    $\hat{s} \leftarrow s + \text{Lap}(0, \lambda)$ 
14:    $O \leftarrow O \cup \langle g, \hat{s} \rangle$ 
15: end for
```

Step 3: Suppressing low counts

```
16: for  $\langle g, \hat{s} \rangle$  in  $G$  do
17:   if  $\hat{s} < \tau$  then
18:      $O \leftarrow O \setminus \langle g, \hat{s} \rangle$ 
19:   end if
20: end for
21: return  $O$ 
```

	Before this project	After this project	Percentage change
Median number of data points released per day	9,000	360,000	+4,000%
Median number of pageviews released per day	50 million	120 million	+240%
Total number of data points released since 2021	8 million	120 million	+1,500%
Total number of pageviews released since 2021	47 billion	116 billion	+250%

Table 6: A comparison of the amount of data published before and after this project, as of June 29, 2023.

- For the 2015–2017 data, the top-1000 drop rate is below 40%, the drop rate above 3500 is below 3%, and the global spurious rate is below 20%.

These metrics show that the privacy-accuracy trade-offs are much better for recent data than for historical data: this is explained by the much tighter sensitivity bound from client-side filtering, allowing to take full advantage of the Gaussian mechanism and its fast-decaying tails.

6 Conclusion

In this paper, we described the process and mechanisms that allowed the Wikimedia Foundation to publish large-scale datasets about user behavior on Wikipedia and other Wikimedia projects. Multiple key factors made this launch possible.

- Tumult Labs’ systematic workflow for differential privacy publications, described in Section 2, provided the structure necessary to move the project forward from its inception to its deployment.
- Combining client-side filtering with server-side aggregation, as described in Section 4.1, was a key innovation that allowed us to obtain user-level differential privacy guarantees for the current pageview data without tracking user identifiers.
- Tumult Core, the privacy framework underlying Tumult Analytics, is designed for extensibility. This made it possible for us to add a novel neighboring definition to this framework to capture the properties of client-side filtering, while still being able to use tight privacy accounting techniques.
- Finally, the scalability offered by Tumult Analytics was essential in handling the massive datasets that were used as input in this project.

The data is now published online [9, 10, 11], along with the source code of the client-side filtering infrastructure [2] and the server-side algorithms [4, 8]. We look forward to seeing what use cases this data will enable!

7 Acknowledgements

We are grateful to Luke Hartman, Tomoko Kitazawa, Nuria Ruiz, and Xabriel J. Collazo Mojica for their help with this project, and to Leila Zia and the anonymous reviewers for their helpful comments and suggestions on this paper.

References

- [1] Academic studies about Wikipedia – Wikipedia. https://en.wikipedia.org/wiki/Academic_studies_about_Wikipedia.
- [2] Clie-side filtering code – Wikimedia Gerrit. <https://gerrit.wikimedia.org/r/plugins/gitiles/operations/puppet/+refs/heads/production/modules/varnish/templates/analytics.inc.vcl.erb#171>.
- [3] Country protection list – Wikitech. https://wikitech.wikimedia.org/wiki/Country_protection_list.
- [4] Differential Privacy – Wikimedia GitLab. <https://gitlab.wikimedia.org/repos/security/differential-privacy/>.
- [5] Differential privacy/Docs/Infrastructure and framework decision-making process – Wikimedia Meta-Wiki. https://meta.wikimedia.org/wiki/Differential_privacy/Docs/Infrastructure_and_framework_decision-making_process.
- [6] Legal:Data retention guidelines – Wikimedia Foundation. https://foundation.wikimedia.org/wiki/Legal:Data_retention_guidelines.
- [7] List of countries by the United Nations geoscheme – Wikipedia. https://en.wikipedia.org/wiki/List_of_countries_by_the_United_Nations_geoscheme.
- [8] Pageview historical notebooks – htriedman GitLab. https://gitlab.wikimedia.org/htriedman/stat-spark3/-/tree/main/pageview_historical/notebooks.
- [9] Pageviews Differential Privacy – Current – README. https://analytics.wikimedia.org/published/datasets/country_project_page/00_README.html.
- [10] Pageviews Differential Privacy – Historical – README. https://analytics.wikimedia.org/published/datasets/country_project_page_historical/00_README.html.
- [11] Pageviews Differential Privacy – Historical (pre-2017) – README. https://analytics.wikimedia.org/published/datasets/country_project_page_historical_pre_2017/00_README.html.
- [12] Policy:Open access policy – Wikimedia Foundation. https://foundation.wikimedia.org/wiki/Policy:Open_access_policy.
- [13] Policy:Privacy policy – Wikimedia Foundation. https://foundation.wikimedia.org/wiki/Policy:Privacy_policy.
- [14] Research:Projects – Wikimedia Meta-Wiki. <https://meta.wikimedia.org/wiki/Research:Projects>.
- [15] Research:Wikipedia clickstream – Wikimedia Meta-Wiki. https://meta.wikimedia.org/wiki/Research:Wikipedia_clickstream.
- [16] Salt (cryptography) – Wikipedia. [https://en.wikipedia.org/wiki/Salt_\(cryptography\)](https://en.wikipedia.org/wiki/Salt_(cryptography)).
- [17] Wikimedia Downloads: Analytics Datasets. <https://dumps.wikimedia.org/other/analytics/>.
- [18] Skye Berghel, Philip Bohannon, Damien Desfontaines, Charles Estes, Sam Haney, Luke Hartman, Michael Hay, Ashwin Machanavajjhala, Tom Magerlein, Gerome Miklau, Amritha Pai, William Sexton, and Ruchit Shrestha. Tumult Analytics: a robust, easy-to-use, scalable, and expressive framework for differential privacy. *arXiv preprint arXiv:2212.04133*, December 2022.

- [19] Nicola Luigi Bragazzi, Cristiano Alicino, Cecilia Trucchi, Chiara Paganino, Iliaria Barberis, Mariano Martini, Laura Sticchi, Eugen Trinka, Francesco Brigo, Filippo Ansaldi, et al. Global reaction to the recent outbreaks of zika virus: Insights from a big data analysis. *PloS one*, 12(9):e0185263, 2017.
- [20] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.
- [21] Clément L Canonne, Gautam Kamath, and Thomas Steinke. The Discrete Gaussian for Differential Privacy. In *Advances in Neural Information Processing Systems*, volume 33, pages 15676–15688. Curran Associates, Inc., 2020.
- [22] Aloni Cohen. Attacks on deidentification’s defenses. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1469–1486, 2022.
- [23] Damien Desfontaines. Demystifying the US Census Bureau’s reconstruction attack. <https://desfontain.es/privacy/us-census-reconstruction-attack.html>, 05 2021. Ted is writing things (personal blog).
- [24] Damien Desfontaines. A list of real-world uses of differential privacy. <https://desfontain.es/privacy/real-world-differential-privacy.html>, 10 2021. Ted is writing things (personal blog).
- [25] Travis Dick, Cynthia Dwork, Michael Kearns, Terrance Liu, Aaron Roth, Giuseppe Vietri, and Zhiwei Steven Wu. Confidence-ranked reconstruction of census microdata from published statistics. *Proceedings of the National Academy of Sciences*, 120(8):e2218605120, 2023.
- [26] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 486–503. Springer, 2006.
- [27] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.
- [28] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
- [29] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In *Proceedings of the 18th international conference on World wide web*, pages 171–180, 2009.
- [30] Tumult Labs. Tumult Analytics. <https://tmlt.dev>, December 2022.
- [31] Joseph Near and David Darais. Differential privacy: Future work & open challenges. *Cybersecurity insights*, 2022.
- [32] Royce J Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. Differentially private SQL with bounded user contribution. *Proceedings on Privacy Enhancing Technologies*, 2:230–250, 2020.
- [33] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web*, pages 1391–1399, 2017.
- [34] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th international conference on world wide web*, pages 1241–1250, 2017.