Open Data Quality Is Poor but Slowly Improving*

Catalogue Quality Scores for Open Data Toronto, December 2019 to January 2021

Amy Farrow

29 January 2021

Abstract

This report uses the data collected by Open Data Toronto for its Data Quality Score. Significant limitations in the scoring system, which is still in beta testing, are discussed. The data show that scores vary across the five quality dimensions, and there is some improvement over time, primarily associated with the addition of new packages. While metrics are useful tools for guiding improvements, the Data Quality Score cannot accurately reflect the holistic value of the portal.

1 Introduction

The Toronto Open Data Portal was launched in 2009, following the increasing public interest in accessible and free data (Toronto, n.d.). The push for open data is a global trend, often linked to open government, democratic participation, and civic empowerment (Sayogo, Pardo, and Cook 2014). Ten years after the portal's launch, technical abilities and expectations for open data had risen significantly (Toronto, n.d.), leading the Open Data Toronto team to consider how the success of the portal could best be measured (Hernandez 2020a). Beginning in December 2019, A Data Quality Score system was developed. In keeping with the spirit of open data, complete scoring results are available on the portal.

More than a year since the Data Quality Score was first used, we can begin to see patterns: scoring is irregular and quality is poor overall, but there is some improvement over time. This report will first consider the scoring data relative to time to consider how different scoring dimensions perform relative to one another, finding that Metadata and Freshness scores are poor. Second, it will demonstrate that scoring has taken place over uneven intervals of time. Third, it will hypothesize that increases in average Quality scores may be associated with increases in number of packages. Fourth, it will compare the grade levels given in January 2021 to those given six months prior and find that the number of higher-grade packages is increasing. Finally, it will discuss the limits of the Data Quality Score, in terms of bias and in terms of measuring the societal value of the portal.

2 Data

Analysis for this project uses the R statistical programming language (R Core Team 2020), and more specifically, the `tidyverse` package for data manipulation (Wickham et al. 2019). Because the data is managed using R Projects, `here` is used to reference file locations (Müller 2020). The data is imported from the Open Data Toronto Portal using the `opendatatoronto` package, which imports data directly (Gelfand 2020); `haven` is used for reading and writing (Wickham and Miller 2020). `lubridate` helps with manipulating dates and times (Grolemund and Wickham 2011), while `scales` fixes date and time axes for visualizations (Wickham

*Code and data are available at: github.com/amycfarrow/metaopendatatoronto.

and Seidel 2020). `DescTools` formats graph labels (Signorell 2020), and `kableExtra` formats tables (Zhu 2020). `bookdown` is used to format the report (Xie 2020).

The data comes from the Data Quality Score project created by Open Data Toronto. The Open Data Toronto portal hosts datasets (referred to as packages) which contain files (referred to as resources) that are available to the public for free. The Data Quality Score project began in late 2019. Their goal was to create a measure other than number of packages that could be used to measure the portal's progress (Hernandez 2020a). This project is still in the beta testing phase. Thus far, the scoring model has been consistent (Hernandez 2021). The scoring model may change in the future, however, as the project adapts (Hernandez 2020a).

The scoring is done by querying the Open Data Portal via the CKAN API (Toronto 2020). They decided to use five dimensions, each with a number of corresponding metrics, selected partially based on what could be automated quickly (Hernandez 2020a). This choice of metrics biases the data: metrics were chosen because they were automatable, not necessarily because they best reflected the quality of the packages.

For each dimension, a package is given a score between zero and one (Hernandez 2020b):

- Accessibility: Is the data easy to access?
  - Metric: Can work with the DataStore API (True/False)
- Completeness: How much data is missing?
  - Metric: Percent of observations missing
- Freshness: How close to creation is publication?
  - Metric: Number of days from published refresh rate to last refreshed
  - Metric: Number of days between last refreshed to today
- Metadata: Is the data well-described?
  - Metric: Metadata fields filled out (True/False)
- Usability: How easy is it to work with the data?
  - Metric: Percent of columns with significant English words
  - Metric: Percent of valid features
  - Metric: Percent of columns with a constant value

These five dimensions are then weighted (Accessibility 7%, Completeness 12%, Freshness 18%, Metadata 25%, and Usability 38%) into a Quality score between zero and one (Hernandez 2020a). Before the Quality score is considered final, it is normalized (Hernandez 2020b). It is also worth noting that this Quality score only measures quality on the portal end. There are many aspects, like accuracy, coherence, precision, reliability, and non-redundancy, which are important to the quality of a package but are not included in the scoring. This is because they are considered to be on the data-provider side, not the portal side (Hernandez 2020b).

When scoring is done, all possible packages on the portal are scored at the same time. In theory, this should reduce bias, as the sample is the entire population. However, many of the packages on the portal are not eligible for scoring. Currently, only data that is in the CKAN Datastore API is scored, due to ease of access for scoring (Hernandez 2020b). This introduces an obvious source of bias: files that are not available through the API are often large (zip files) or non-ideal formats (Excel or PDF), meaning that the lower-quality packages may have been disproportionately excluded from the scoring. For this reason, the average scores can only be considered to reflect a specific subcategory of the whole data catalogue. Another issue is that Read Me files are scored and weighted exactly the same as data resources, despite having very different values and qualities (Toronto 2020).

This data is available in the resource titled 'catalogue-scorecard' in the package 'Catalogue quality scores'. There are 13 features: an ID number and name for the package; Accessibility, Completeness, Freshness,

Metadata, and Usability dimension scores for the package; Quality and normalized Quality scores for the package; grade and normalized grade for the package; the day and time the scoring was done; and the version of scoring that was used. I have used package name rather than ID to identify unique packages. I chose to use the non-normalized quality scores, because the scores are normalized using min-max scaling relative to the other packages on the portal at that moment in time, confusing any trends over time. I used the normalized grades as opposed to the non-normalized ones, because the normalized grades are considered the final measure. There are 126 unique times when scoring was done, and 143 unique packages that were scored. The data requires minimal cleaning (converting the date_scored feature to datetime), and has no missing values.

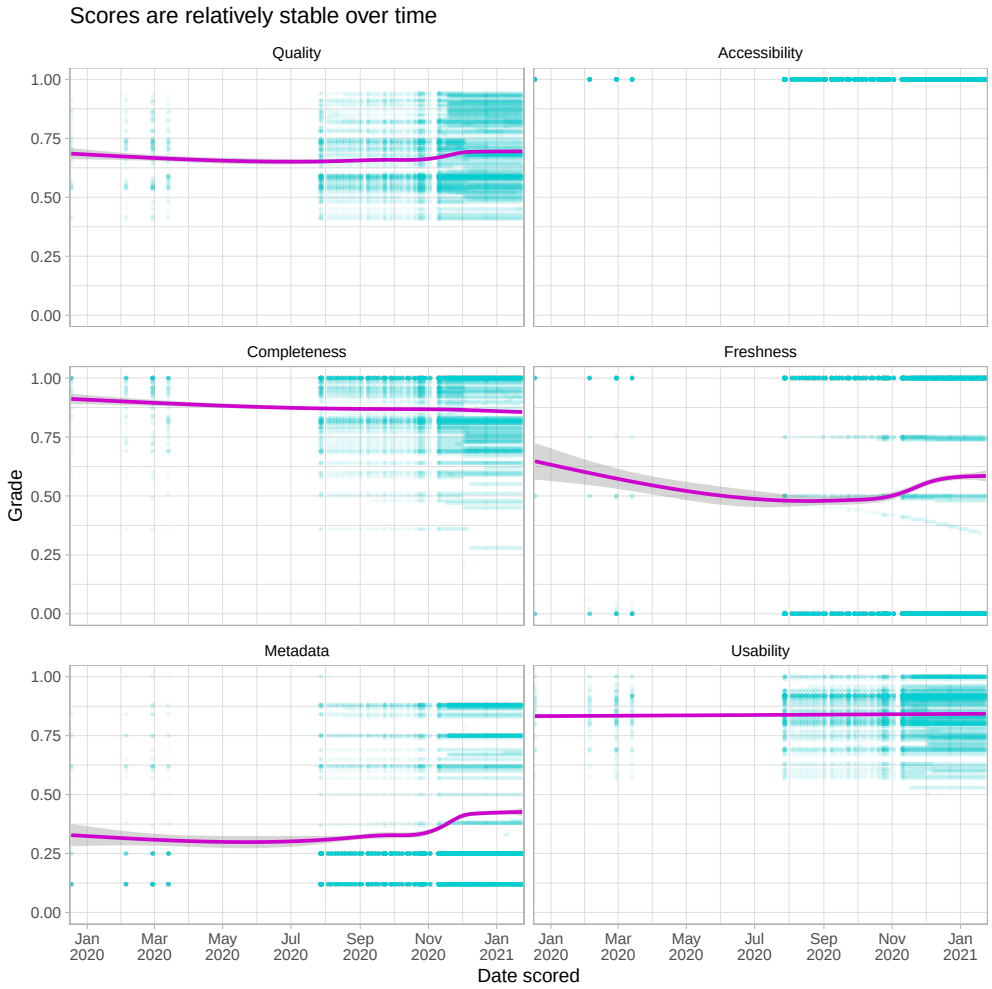

Figure 1: Scores over time

Figure 1 shows all five dimension scores and the final Quality score over time. Each point represents a scored package, and the line is fitted to the overall trend. In the case of the Accessibility score, no line has been fitted: this is because all packages at all dates received a score of 1. This is logical, because Accessibility is based on availability through the Datastore API, and packages are only scored if they are available through the Datastore API. We can see that some dimensions perform better than others: Completeness and Usability are relatively high and stable, while Freshness is lower and Metadata is lower still. There is some fluctuation in these last two dimension scores over time. The distributions of the dimension scores are wide, especially the Freshness and Metadata distributions, indicating that quality varies significantly depending on package. The overall performance is moderate: Quality scores trend just below the 0.7 mark for the entire time that packages have been scored.

Figure 1 also shows an interesting pattern with regards to the date the package was scored: there are erratic scores beginning in December 2019, but the density of scoring increases in July 2020 and again in November 2020.

Table 1 shows this trend more clearly. The DQS program was released in December 2019, and intermittently manually run in February and March 2020. It was not run at all in April, May, or June 2020. In July 2020, a system was put in place for automatic scoring, but the DQS team found that it was not reliable or consistent enough. Subsequently, in November 2020, they switched to another platform and increased the scoring frequency to daily. This may change again in the future, as the DQS team is finding daily scoring to be excessive considering the portal is not updated that frequently (Hernandez 2021).

Table 1: Scoring by month

	Manual scoring			Scheduled scoring						
	Dec 2019	Feb 2020	Mar 2020	Jul 2020	Aug 2020	Sept 2020	Oct 2020	Nov 2020	Dec 2020	Jan 2021
Counts										
Unique packages scored	57	73	101	103	106	110	111	136	138	138
Times the portal was scored	1	3	1	4	10	11	15	25	32	24
Average Scores										
Accessibility	1	1	1	1	1	1	1	1	1	1
Completeness	0.919	0.908	0.869	0.868	0.870	0.867	0.865	0.874	0.859	0.858
Freshness	0.560	0.651	0.493	0.461	0.481	0.490	0.490	0.523	0.576	0.580
Metadata	0.317	0.321	0.297	0.307	0.314	0.326	0.334	0.379	0.418	0.425
Usability	0.835	0.824	0.839	0.838	0.839	0.840	0.837	0.843	0.843	0.841
Quality	0.668	0.682	0.652	0.648	0.655	0.659	0.660	0.678	0.693	0.694

We can see a slight upwards trend in average Quality scores since the automated scoring began in July. However, we can also see that while Accessibility, Completeness, and Usability scores are strong, Freshness and Metadata scores are weak, indicating that the data needs to be refreshed more often and the metadata more carefully completed.

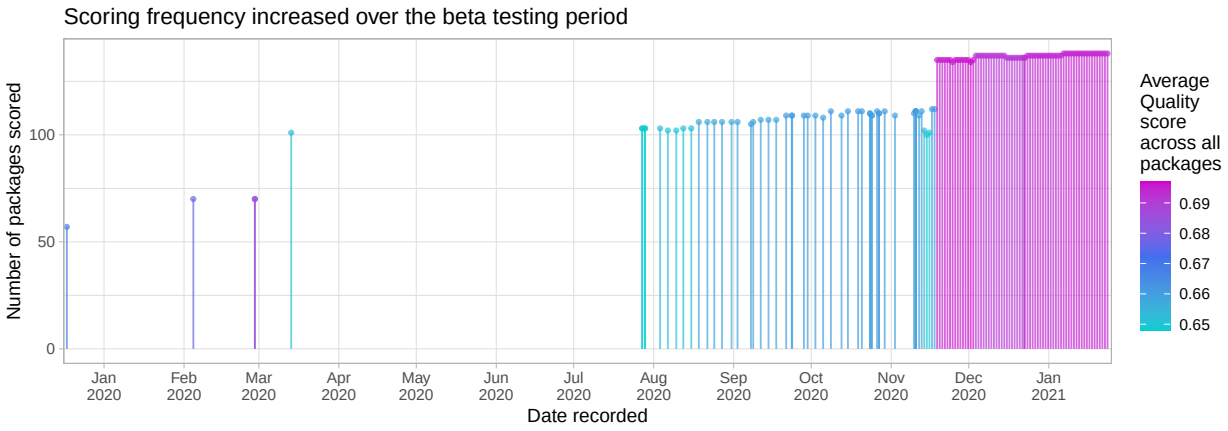

Figure 2: Scoring frequency over time

Figure 2 takes a more granular look at average quality, scoring frequency, and scoring volume over time. Every bar represents a time that the portal was scored. The height of the bar shows the number of packages that were scored, and the colour of the bar indicates the average Quality score that was given. We can see the regular scoring beginning July 27th, 2020, and the more frequent scoring beginning on November 10th, 2020.

In Figure 2, we can also see that the number of packages scored seems to change at the same time as the average Quality score. Most strikingly, the November 19th scoring shows that 23 more packages were scored than on the previous day, and the average Quality score increased from 0.4932143 to 0.5559259. This suggests that a change in average score over time may reflect the increasing quality of newly added packages, rather than the increasing quality of the existent packages.

Data portal users do not see the five dimension scores and final Quality scores. Because the Open Data Toronto team wanted the focus to be on the overall quality rather than minute changes in numerical scores, they decided to break the full spectrum of Quality scores into three medal grades (Hernandez 2020b):

- Bronze: normalized Quality score less than 0.6
- Silver: normalized Quality score 0.6 to 0.8
- Gold: normalized Quality score greater than 0.8

These medal grades are what is actually visible to the public, as we can see in the case of the 'Catalogue quality scores' package itself, which receives a silver grade as of January 24th 2021 (shown in Figure 3).

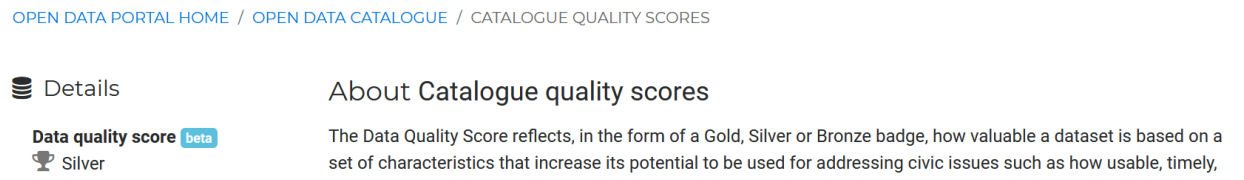

Figure 3: The 'Catalogue quality scores' package receives a silver grade

Since the grades are what the portal users see, we should consider the numbers of bronze, silver, and gold packages on the portal. Using the first systematized evaluation in July 2020 and the most recent evaluation in January 2021, we can see how the numbers have changed in the last half year. Figure 4 shows that while there are still proportionally more Bronze packages overall and the number of Bronze packages has increased, the number of Gold packages increased far more.

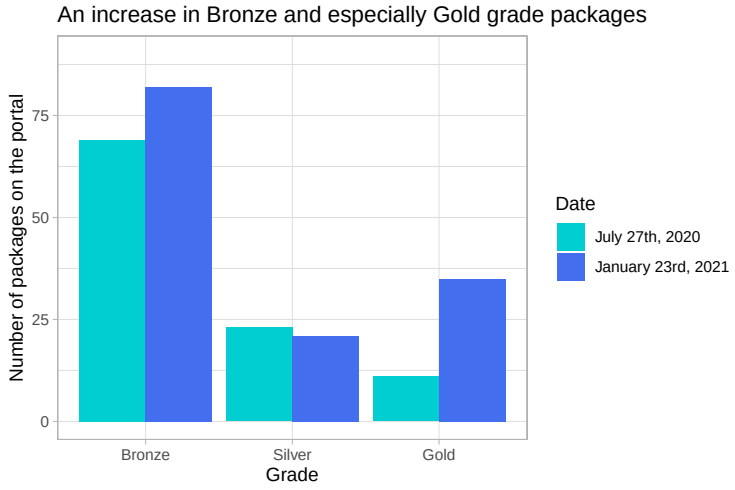

Figure 4: Comparing grade numbers in July and January

The change in numbers indicates that the portal is improving slightly over time, from a user's perspective. Users may also perceive improvement based on the total number of packages scored: from July 27th 2020 to January 24th 2021, it has increased from 103 to 138, which means that the number of API-accessible packages has increased.

While grades are an excellent way to communicate the likely overall quality of individual packages to portal users, looking at patterns in the scores is not the best way to assess the overall state of the portal. This is because there are limitations to these measures: not all packages are scored, there are significant factors that are not considered scorable (like accuracy and reliability), the scoring system is still being tested, and the scoring schedule is still in flux. More importantly, an open data portal is not only about the quality of packages. Sayogo, Pardo, and Cook (2014) reviewed government open data portals and created a model for assessing the value of an open data portal. Specifically, they were interested in the importance of data manipulation and engagement for user experience. A portal with extensive manipulation capabilities allows users to amass, sort, and analyze data on the portal, while advanced engagement means that users are collaborative creators who can work with the portal and other users. Both of these capabilities make it easier to engage with the open data, hopefully enhancing the impact that an open data portal has on the civic community. Whenever assessing an open data portal based on a score, especially one with significant limitations, we should remember that it represents only one part of the portal's societal value.

References

Gelfand, Sharla. 2020. *Opendatatoronto: Access the City of Toronto Open Data Portal.* https://CRAN.R-project.org/package=opendatatoronto.

Grolemund, Garrett, and Hadley Wickham. 2011. "Dates and Times Made Easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. https://www.jstatsoft.org/v40/i03/.

Hernandez, Carlos. 2020a. "Towards a Data Quality Score in Open Data (Part 1)." Medium. https://medium.com/open-data-toronto/towards-a-data-quality-score-in-open-data-part-1-525e59f729e9.

———. 2020b. "Towards a Data Quality Score in Open Data (Part 2)." Medium. https://medium.com/open-data-toronto/towards-a-data-quality-score-in-open-data-part-2-3f193eb9e21d.

———. 2021. "Re: Questions About Number of Packages Scored and Scoring Schedule in Package "Catalogue Quality Scores"." Personal communication via email.

Müller, Kirill. 2020. *Here: A Simpler Way to Find Your Files.* https://CRAN.R-project.org/package=here.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Sayogo, Djoko Sigit, Theresa A. Pardo, and Meghan Cook. 2014. "A Framework for Benchmarking Open Government Data Efforts." Hawaii International Conference on System Science.

Signorell, Andri et mult. al. 2020. *DescTools: Tools for Descriptive Statistics.* https://cran.r-project.org/package=DescTools.

Toronto, Open Data. 2020. "Catalogue Quality Scores." City of Toronto Open Data Portal. https://open.toronto.ca/dataset/catalogue-quality-scores/.

Toronto, City of. n.d. "About City of Toronto Open Data." City of Toronto Open Data Portal. https://open.toronto.ca/about/.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, and Evan Miller. 2020. *Haven: Import and Export 'Spss', 'Stata' and 'Sas' Files.* https://CRAN.R-project.org/package=haven.

Wickham, Hadley, and Dana Seidel. 2020. *Scales: Scale Functions for Visualization.* https://CRAN.R-project.org/package=scales.

Xie, Yihui. 2020. *Bookdown: Authoring Books and Technical Documents with R Markdown.* https://github.com/rstudio/bookdown.

Zhu, Hao. 2020. *KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax.* https://CRAN.R-project.org/package=kableExtra.