

APPLYING WEB TECHNOLOGIES FOR DATA MINING OF MODELS OF
ADVANCEMENT OVER TIME IN SPACE EXPLORATION TECHNOLOGY

A Thesis Submitted
to the Graduate School
University of Arkansas at Little Rock

in partial fulfillment of requirements
for the degree of

MASTER OF SCIENCE
in Information Science

in the Department of Information Science
of the Donaghey College of Science, Technology, Engineering, and Mathematics

December 2020

Peng-Hung Tsai

M.B.A., University of Arkansas at Little Rock, 2005

© Copyright by
Peng-Hung Tsai
2020

This thesis, "Applying Web Technologies for Data Mining of Models of Advancement Over Time in Space Exploration Technology," by Peng-Hung Tsai, is approved by:

Thesis Advisor:

Daniel Berleant
Professor of Information Science

Thesis Committee:

Richard S. Segall
Professor of Information Technology

Hyacinthe Aboudja
Associate Professor of Computer Science

Program Coordinator:

Daniel Berleant
Professor of Information Science

Graduate Dean:

Brian Berry
Professor of Chemistry

Fair Use

This thesis is protected by the Copyright Laws of the United States (Public Law 94-553, revised in 1976). Consistent with fair use as defined in the Copyright Laws, brief quotations from this material are allowed with proper acknowledgment. Use of this material for financial gain without the author's express written permission is not allowed.

Duplication

I authorize the Head of Interlibrary Loan or the Head of Archives at the Ottenheimer Library at the University of Arkansas at Little Rock to arrange for duplication of this thesis for educational or scholarly purposes when so requested by a library user. The duplication will be at the user's expense.

**APPLYING WEB TECHNOLOGIES FOR DATA MINING OF MODELS OF
ADVANCEMENT OVER TIME IN SPACE EXPLORATION TECHNOLOGY**
By Peng-Hung Tsai, December 2020

ABSTRACT

The development of web technologies has progressed rapidly in the past few decades. As a result, these technologies have increasingly gained popularity among researchers as an instrument for data analysis and visualization to aid their work. The purpose of this project is to pursue a new method for mining time-based data to perform curve fitting, using web client technology. This requires comparing the advantages and disadvantages of different JavaScript software designs to determine which is best. Based on the results, we emphasize developing a client-side web application and demonstrate its use through fitting a curve for a new metric to space mission data. The tool can act as a foundation for more advanced work using data from other domains.

TABLE OF CONTENTS

LIST OF FIGURES	vii
LIST OF TABLES.....	viii
1. INTRODUCTION.....	1
2. REVIEW OF TECHNOLOGIES	2
3. EVALUATION OF METHODS	14
4. INTRODUCTION TO MODELING ADVANCEMENT IN SPACE EXPLORATION TECHNOLOGY	16
5. INTENDED CONTRIBUTIONS	18
6. PREVIOUS WORK IN APPLYING MOORE’S LAW TO STUDY TECHNOLOGICAL ADVANCEMENT	19
7. CONSIDERATIONS OF FACTORS AFFECTING TECHNOLOGICAL ADVANCEMENT	20
8. METHODOLOGY	22
9. RESULTS	24
REFERENCES	37
APPENDIX.....	45

LIST OF FIGURES

Figure 1 – Evaluation results of web technologies	25
Figure 2 – Evaluation results of web technologies	25
Figure 3 – Evaluation results of web technologies	27
Figure 4 – Evaluation results of web technologies	28
Figure 5 – Evaluation results of web technologies	29
Figure 6 – Evaluation results of web technologies	30
Figure 7 – Evaluation results of web technologies	31
Figure 8 – Evaluation results of web technologies	32
Figure 9 – Evaluation results of web technologies	33
Figure 10 – Evaluation results of web technologies	34
Figure 11 – Evaluation results of web technologies	35
Figure 12 – Evaluation results of web technologies	36

LIST OF TABLES

Table 1 – Evaluation results of web technologies	16
Table 2 – Evaluation results of web technologies	26
Table 3 – Evaluation results of web technologies	27

1. INTRODUCTION

The term curve fitting describes a method that determines a model that provides the best fit curve to the specific values in a data set. It is widely used in various disciplines, such as economics, social sciences, data mining, etc. There are many software tools that can be used for curve fitting. Among the popular choices, a frequently used tool by researchers has been a spreadsheet program, such as Excel or Google Sheets, due to its simplicity and versatility in data analysis and visualization. Spreadsheets do a good job of presenting data clearly, which makes it easy to identify data quality problems, such as missing or duplicate values. Creating formulas to perform numerical calculations and manipulate data to generate the graphs in a spreadsheet is also very straightforward. However, one drawback of spreadsheets is their inflexibility in enabling us to add functionality we might need for advanced analysis without using plugins or supported programming languages. For example, Excel needs Visual Basic scripting knowledge while Google Sheets needs Google Apps Script in order to use their functionalities to the greatest extent possible. Figuring out how to use these tools can be quite challenging.

For the past 20 years, the evolution of web technologies has given rise to the widespread development of web applications (Wikipedia, 2020a). The growing trend toward client-side web applications is driving a more responsive and interactive user experience through an immediate content update from within the browser with fewer resources required from the server. It has been greatly increasing the possibility of performing curve fitting directly on the browser.

In this project, we will develop a client-side web application and demonstrate its use through fitting a curve to space mission data. This work will not only illustrate the use of a client-side web application in providing an interactive curve fitting tool but also act as a foundation for more advanced work using data from other domains such as economics, medical sciences, or other social sciences, etc. This study is organized in the following sequence. First, we investigate various modern web technologies that might be useful for our web application development. Second, we evaluate each technology using the metrics from ISO/IEC 25010 (Estdale & Georgiadou, 2018) to determine how it would perform to suit our needs. Third, we give an introduction to modeling advancement in space exploration technology and describe how the built tool can contribute to making conducting research more efficient and timely. Fourth, a review of relevant literature on the studies of modeling advancement in technologies will be provided. Next, we introduce our methodology of modeling advancement in space exploration technology. The last part presents the data fitting results and the extrapolation predictions for the progression of space exploration technology.

2. REVIEW OF TECHNOLOGIES

The overall research program requires that a set of data records be loaded and analyzed. Furthermore, it is desirable that the user could modify the data records in their own personal copy. It is also important that the system does not depend on maintenance of a special server running unique code, because that would present a permanent resource sink such that if the needed resources were

ever interrupted the system would become unavailable. In this, the system is unlike the popular commercial paradigm of cloud computing in which cloud based systems are maintainable because the services they provide yield a flow of funds that can be used for maintenance. With these considerations in mind we next examine a number of different strategies for storing the data records.

Screenshot

Among different methods to present data, taking a screenshot of the data and graphs in a spreadsheet and placing the images on the Web would take the least time to implement. Despite this, whenever there is an update in data or graphs, new copies of images will have to be uploaded to replace the old ones. Because images only serve the purpose of displaying the content, this method is restricted to presenting data only. Unless the data is also made available for download, users will not have access to the data for data editing or analysis.

HTML Table

HTML (HyperText Markup Language) (W3Schools, 2020a) can be used to create a table structure of columns and rows, an HTML table, to display data on the Web. Its basic structure is a table element, which uses the `<table>` and `</table>` tags to mark the start and end of the table and contains some number of rows between them. An HTML table can be used to present text, graphs, or sets of data in a tabular format. Basically, anything that could be put into a spreadsheet could go in a table.

We can control how content appears in an HTML table by linking a CSS

(Cascading Style Sheets) style sheet (W3Schools, 2020b) to the HTML code (W3Schools, 2020a). The size and position of the table and the size of each column and row can also be specified. Although HTML tables only allow for data presentation but not data editing or analysis, if users need data for their analysis, they could simply select and copy the data they want and then paste it into a spreadsheet, or alternatively use Excel to directly extract data from the web page. This is an advantage compared to a screenshot, which does not allow convenient access to the data. Such access is needed, however, to visualize the data with charts and graphs. We will need to use some JavaScript graphing libraries for this, which will be described later.

Alternatively, users could edit the HTML table directly in the HTML file, or edit it using an editor made for editing HTML files. Then they would have to save the HTML file on their local machine for future use. However it would be a burden on the user to have to do this.

HTML Form

An HTML form, defined by the `<FORM>` and `</FORM>` tags in a web page, is used to collect data inputted by users and thus, unlike screenshot and HTML Tables, allows for interactivity within the page. The process of creating an HTML Form set up with initial data is similar to the process of creating an HTML Table. However, as HTML only builds a form's structure, a form on its own is not especially useful but often used with HTML Tables and some programming languages, such as JavaScript and PHP, to create a dynamic web page that

enables users to input data, which is then sent to a web server for processing or used to add new data on the client-side. A form's appearance can also be controlled by adding some CSS (W3Schools, 2020b) to its HTML code (W3Schools, 2020a).

A possible disadvantage of this method is that it is not clear if the user would have trouble saving the edited page on their machine for later use. However, a potential advantage of this approach is that the form submit button could be set up to tell the server about the updates to the form data and the server could then, if it was constructed properly, make a record of the changes that the research team could use to curate and perhaps update the default data in the form in the web page provided by the server.

JSON

JSON, or JavaScript Object Notation, is a language-independent, text-based syntax that is derived from JavaScript for storing and exchanging data (W3Schools, 2020d). JSON is commonly used as a data interchange format with most programming languages to transport data between browsers and servers. JSON data can be saved in its own file with the .json extension or stored in an HTML file and accessed using the dot notation (.). It can also be retrieved from a server through JavaScript codes (W3Schools, 2020c) into a web application and converted to an HTML table in the browser. The following example shows a simple JSON format containing two name/value pairs:

```
{  
  "Make": "Nissan",  
  "Price": 10000  
}
```

JSON has some advantages. JSON syntax is simple and easy to work with as it includes no more than commas, curly braces, square brackets, and name/value pairs data. Machines also can handle JSON data easily, so if JSON is used on the server side, users can get a fast response from the server. Moreover, the fact that JSON is recognized natively by JavaScript makes it well-suited to being used in JavaScript-based web applications. However, its flexibility and lack of schemas limit the ability to verify that the data is correct. If the data is of poor quality, the application using the data can fail. Another issue with the JSON approach is the situation where the user wishes to edit and save the data, then reload it later. The JSON data would need to be stored on the user's local machine, for example in the JavaScript localStorage data repository. This is certainly possible, but the stored data is relatively inaccessible to the user except through the web page that it is stored/read within, making it unclear how the user might move the stored data to another machine.

Note that JSON for storing data is compatible with the HTML table and form methods of presenting it in a web page, and so is not an alternative to them, but rather an option to use or not use with them. Similarly, JSON is not a full solution but needs to be used in conjunction with presenting the data using an HTML table or form or perhaps some other presentation method. Thus, JSON would be

potentially useful for the research team as a way to edit and view the raw data, perhaps preferable to reading raw HTML or depending on a commercial HTML editor with the necessary functionality. The JSON file would probably be invisible to the user, unless they tried to change machines in which case they would be stuck or need to be sophisticated users with knowledge of browser-specific intricacies (and they still would likely not need to see the actual JSON data).

Google Sheets

Google Sheets is a cloud-based spreadsheet software that has very similar functionalities to Excel. Anyone who has a Google account can access it easily with any Internet browser and work quickly in entering and analyzing data with little to no programming required. Users can also conduct complicated calculations, such as solving what-if types questions, without redoing the calculations. A distinct feature of Google Sheets is that it enables a user to work on it simultaneously with other users and decide whether they can just view or edit. This ensures that data security is kept in check.

Despite being limited in customizability, its graphing functionality is easy to operate for simple visualizations with various chart types, which can be configured for automatic updates from source data. Making further customizations to the chart can be done with some JavaScript coding and functions.

Sophisticated processing of data could be done in the web page that serves as a front end to the Google sheet, but that processing could also be done directly in the Google sheet. Comparing these two options, many users are likely

to be more comfortable seeing the space mission data in a web page than in a Google sheet. One reason is that the web page approach better supports explanatory text and content that places the data in context. Also, the web page approach gives users the ability to perform advanced data simulations or modeling without any knowledge of coding or expertise in using plugin tools.

Open-source software libraries

Tables display tabular data with multiple rows and columns. Nearly all websites use tables to present data. In addition to using HTML Tables, various open-source JavaScript libraries are available to achieve this purpose. However, unlike standard HTML tables that do no more than providing a basic layout of data, many of them provide advanced features for direct data manipulation, like filtering, sorting, dynamic editing, etc., in a table-like interface. Using these libraries can quickly transform almost any data source to a beautiful interactive table with a short length of JavaScript code.

Since comparing each of them is difficult and not a part of this project, we look at two libraries, DataTables and Handsontable, that would best fit our project as both are easy to use and enable responsive interaction. They also have detailed documentation on using their functionalities. They are compatible with major JavaScript libraries including jQuery (Wikipedia, 2020d), which is used to make DOM (Document Object Model) manipulation easier. Best of all, they both support client-side processing, which allows us to implement curve fitting directly in the browser.

DataTables is a very flexible library (DataTables, 2020). It supports inline

editing and allows us to easily add filtering, sorting, searching, and pagination to pure HTML tables. Additionally, it offers a variety of extensions to add more advanced interaction controls.

Handsontable has a very intuitive interface because it provides Excel-like table editing (Handsontable, 2020). Users can simply double-click on a cell via a mouse to edit its value and also perform all other major Excel tasks, such as copy, remove, merge cells, insert a row or column, etc. With Handsontable, anyone with experience working with Excel or Google Sheets will find similar spreadsheet experience. It is free for non-commercial purposes.

Graphing libraries

Visualizing data via charts or graphs helps people analyze data and understand data more easily. It also enables identifying patterns within data sets that we could not see if simply looking at a table. Our goal is to find the most appropriate visualization tool for our curve-fitting method development needs.

There are many different JavaScript graphing libraries that can be utilized to create versatile charts and graphs (Wikipedia, 2020b). Some are simpler or faster, while some are more difficult to learn with a steep learning curve. Some are free while others are paid for. It is outside our present scope to compare all of them comprehensively, and sometimes it may be best to complement one with another. For our project, we consider these factors:

1. Is the line chart type available to choose from?
2. Is the library compatible with major JavaScript frameworks such as

jQuery?

3. Is the chart responsive and can it be customized?
4. Is the library available free of charge for research use?

The assessment results of these factors lead us to two options, amCharts and Chart.js, both of which would work best for our project. We will see next how they might be integrated with our design.

amCharts is a simple yet flexible graphing library that has great customization functionality (amCharts, 2019). It is compatible with major JavaScript frameworks such as jQuery, Angular, React, etc. In addition, it can work with Handsontable, an Excel-like interface, so that users who are comfortable with using spreadsheets can easily enter and manipulate data for their charts on the web page. It is free for any use as well although a small mark "JS chart by amCharts" will be added onto the graph.

Chart.js is known for its simplicity and user friendliness. It allows users to create responsive, interactive, and customizable charts and graphs in a short time (Chart.js, 2019). It also provides integrations with jQuery, Angular, React, etc. It is a free open-source library.

Server-side vs. client-side web applications

According to Wikipedia, a web application is a client–server computer program that the client (the user) runs in a web browser (Wikipedia, 2020a). It can include scripts that are run on the client's browser or on the web server. Server-side scripts, such as PHP and ASP.NET, are the codes that are executed

by servers and used to handle the processing and storage of data, while client-side scripts, such as HTML and JavaScript, are codes that are executed by browsers and used to display data to users. This allows users to interact with a web page. Some examples of web applications include interactive games, online surveys, shopping carts, online banking, etc.

Conventionally in a web application, the user sends a request to the server, and then the server processes the data before sending a response back to the user. In this flow, the majority of the processing of the data is done at the server end. Nowadays, JavaScript frameworks have brought a trend toward client-side web application development, which shifts much processing to the browser. When the flow simply occurs on the browser, the response to the user's action becomes faster as a trip to the server is not needed. Also, resources on the server are much less used. However, because client-side scripts are mostly written in JavaScript (W3Techs, 2020) along with HTML and CSS, the browser must support or not block the use of these languages in order to run client-side scripts.

Custom client-side applications with JavaScript

The aforementioned investigations of various web technologies have led us to propose a web tool by taking advantage of the strengths of each of them. Because Handsontable has been incorporated by amCharts into its product design and they work together very smoothly, we chose to use both as our table and graphing library respectively for our development work. One potential drawback to note, however, is that if Handsontable or amCharts goes out of

business, our application will stop working.

We also use the jQuery library to employ a useful method, `getJSON()`, that allows us to easily retrieve JSON format data from a Google sheet for storing space mission data (Boggarapu, 2016). Given that we prefer not to need a special physical server to store our data and to execute code, and the tool that we develop requires a fast response from the browser, we determined to develop a client-side rather than server-side application.

The reasons that we use Google Sheets as the backend to store our data are many and obvious. It is available for free in the cloud and can be easily accessed with any web browser. Users on Excel can find themselves comfortable manipulating data using Google Sheets because of its similarity with Excel and user-friendly interface. Data security can also be assured by limiting access to the data to appropriate people while keeping the data public. If users want to create their own copy or download the data that they can then edit, it will only require several clicks to achieve it. Above all, without any backend programming knowledge, we can pull the data in JSON format from Google Sheets simply using a few JavaScript scripts.

Before we extract the JSON format data from our Google spreadsheet, we need to change the document settings and create the spreadsheet URL by taking the following steps:

1. Open the Google spreadsheet file, and then click on the "Share" button on the top right corner. Click "Advanced", and then click "Change..." under "Who has access". Select "On - Public on the web", and then click "Save". On the pop-up window of "Sharing settings", make sure "Public on the web - Anyone

on the Internet can find and view" is shown under "Who has access", and then click "Done".

2. Go to "File". Click "Publish to the web", and then click "Publish".
3. Get the spreadsheet ID (bolded below) of the spreadsheet from the URL in the address bar.

<https://docs.google.com/spreadsheets/d/1ILB9IUsCnZ7bh3UX9nJ-enyfr-Cqjj543xfXPmdG9Zc/edit#gid=846789546>

4. Replace the spreadsheetID in the following URL string with the ID acquired in the previous step.

<https://spreadsheets.google.com/feeds/worksheets/spreadsheetID/public/basic?alt=json>

The resulted URL will look like this:

<https://spreadsheets.google.com/feeds/worksheets/1ILB9IUsCnZ7bh3UX9nJ-enyfr-Cqjj543xfXPmdG9Zc/public/basic?alt=json>

5. Open a new browser using the new URL above to get the JSON format data, and find the child element "entry" to get the worksheet ID: oe05mr4, as bolded below.

```
"entry": [
  {
    "id": {
      "$t":
        "https://spreadsheets.google.com/feeds/worksheets/1ILB9IUsCnZ7bh3UX9nJ-enyfr-Cqjj543xfXPmdG9Zc/public/basic/oe05mr4"
    },...
```

6. Replace spreadsheetID and worksheetID in the following URL string with the spreadsheet ID and worksheet ID respectively.

`https://spreadsheets.google.com/feeds/list/spreadsheetID/worksheetID/public/values?alt=json`

The resulted URL will look like this:

`https://spreadsheets.google.com/feeds/list/1ILB9IUsCnZ7bh3UX9nJ-enyfr-Cqjj543xfXPmdG9Zc/oe05mr4/public/values?alt=json`

7. Place this URL inside the `getJSON()` method in our application script as shown below.

```
$.getJSON('https://spreadsheets.google.com/feeds/list/1ILB9IUsCnZ7bh3
UX9nJ-enyfr-Cqjj543xfXPmdG9Zc/oe05mr4/public/values?alt=json',
function(data) {}
```

The specific steps that we took and the corresponding scripts that we used to build our application are shown in the Appendix.

3. EVALUATION OF METHODS

In this section, we assess each method according to the criteria adapted from ISO/IEC 25010 (International Organization for Standardization, 2011), which is a widespread international quality standard derived from ISO/IEC 9126 (Wikipedia, 2020c). Both standards were prepared by the joint technical committee from ISO (the International Organization for Standardization) and IEC (the International

Electrotechnical Commission) to evaluate software quality.

The standard defines eight quality attributes, which are further broken into sets of sub-attributes, in its product quality model: functional suitability, performance efficiency, compatibility, maintainability, portability, reliability, security, and usability. Among them, we consider six attributes that are suitable for the evaluation of these methods and specify these six attributes as follows:

1. Functional suitability: The level to which the method meets functional requirements.
2. Performance efficiency: The duration of the response and processing times of the method when performing its functions.
3. Compatibility: The level to which the method can work with other technologies to perform its function.
4. Maintainability: How easily the method can be modified or updated.
5. Reliability: How well the method performs desired functions for a specified period of time.
6. Usability: How effectively and efficiently users can use the method to complete the task.

We use the following indicators to assess each method.

3 – Effective

2 – Moderately Effective

1 – Not Very Effective or Ineffective

The evaluation results are shown in Table 1 below.

Table 1.

	Screenshot	HTML Table	HTML Form	JSON	Google Sheets	Open-Source Software Library	Graphing Library	Server-Side Web Application	Client-Side Web Application
	Constituents of Web Application Architecture								
Functional suitability	1	1	1	1	3	1	1	3	3
Performance efficiency	1	3	3	3	3	3	3	2	3
Compatibility	2	3	2	2	3	2	2	3	3
Maintainability	1	1	1	1	2	1	1	3	3
Reliability	3	3	3	3	3	3	3	3	3
Usability	1	2	1	1	3	2	1	3	3
Total	9	13	11	11	17	12	11	17	18

4. INTRODUCTION TO MODELING ADVANCEMENT IN SPACE

EXPLORATION TECHNOLOGY

Many technologies have been proved to conform to the general form of

Moore's law, which states that the rate of advances in technologies will increase

exponentially over time, despite at different rates of change (Farmer & Lafond, 2016). Another law, Wright's law (also called the power law or experience curve), which has also been proposed by many researchers, provides approximations of comparable quality for the pace of technological improvement (Nagy et al., 2013; Magee et al., 2016; Lafond et al., 2018; Berleant et al., 2019). While these laws have served as guides for technology progress in various domains, only a few studies have demonstrated that space exploration technology follows similar patterns (Howell et al., 2019a). The fact that the human population of space has historically exhibited a general upward growth trend (Elliott, 2014; SpacePunx, 2016) seems to make it promising that the curve may also follow an exponential growth pattern, but even so, some have pessimistically argued that the progress of human space exploration has come to a stop (Hicks, 2015; Bardi, 2016). Despite these arguments, it remains debatable what the right proxy is for the performance measure of space exploration technology since space exploration progress cannot be directly measured.

Recently, Berleant et al. (2019) attempted to employ spacecraft lifespan as a proxy for advancement in space exploration technology and found an exponential trend similar to Moore's law and Wright's law. Howell et al. (2019b) later compared the spacecraft technological progress of the United States to that of the Soviet Union in the Space Race age and concluded that the former had a higher rate of improvement than the latter. Both of these studies chose to use a single proxy rather than multiple variables, thus helping to lower the risk of overfitting. Ultimately an extrapolatable curve with a perfect fit to the data will have limited value if it has no predictive power (Kodali et al., 2018). Our study

extends their research to incorporate an additional variable into the general form of Moore's law for a curve that models technological advancement to pursue a more detailed model of advancement in interplanetary exploration vessels.

Specifically, we utilize NASA's budget along with the time variable to model their relationship with spacecraft lifespans. Our goal is to identify a model that not only provides a good fit and forecast but is intuitive and simple to apply.

5. INTENDED CONTRIBUTIONS

This project will provide a new approach to curve fitting using a custom client-side web application. The application will allow users with any technical background to easily explore and manipulate data for curve fitting by changing a set of parameters and comparing different fit results graphically and numerically on the same screen in real time. These accessibility and interactive features will greatly improve how users understand and analyze data and ultimately aid research. Moreover, by demonstrating its use through fitting a curve for a new metric to space mission data, our results will contribute to the literature related to the advancement in space exploration. After all, understanding how technological performance in space exploration improves over time will provide better foresight in formulating technology policy as well as planning unmanned exploration missions to distant objects (Hall et al., 2017). Although we only focus on using space mission data in this research, the new tool and the methodology for curve fitting could be applied to different data sets from other domains.

6. PREVIOUS WORK IN APPLYING MOORE'S LAW TO STUDY TECHNOLOGICAL ADVANCEMENT

In 1965, Gordon Moore, the co-founder of Intel and Fairchild Semiconductor, described in his article published in *Electronics* that the number of components per integrated circuit had doubled approximately every year from 1959 until 1965 (Moore, 1965). He predicted that this growth would continue at around the same pace for at least the next ten years. He then in 1975 revised the growth rate to a doubling time of every two years (Mollick, 2006). Such exponential progress in the number of components on an integrated circuit eventually became known as Moore's law. Over the years, a generalized version of Moore's Law has taken on a meaning referring to an exponential growth trend (Sood et al., 2012; Nagy et al., 2013; Singularity Symposium, 2020).

The generalized Moore's Law has been observed in many other technologies and technology domains than the semiconductor sector since Moore's article. For instance, Nielsen (1998) looked into internet bandwidth growth rates and projected that "a high-end user's connection speed grows by 50% per year", which suggests that Internet bandwidth doubles almost every 21 months. Mark Kryder proposed that disk drive areal density increases 1,000-fold every 10.5 years (Walter 2005), which is approximately equivalent to a doubling of storage capacity every 13 months. Additionally, Robert Carlson found evidence that the doubling rate of DNA sequencing and synthesis technologies has followed an even faster Moore's law (Specter, 2009). Also referred to as the Carlson Curve, this exponentially growing progress has come to be known as the

biotechnological equivalent of Moore's Law. More recently, Bailey et al. (2012) and Nagy et al. (2013) examined the cost and production data of 62 different technologies and found exponential relationships between performance and time.

Different technology domains have also been found to advance following an exponential function of time. For example, Koh and Magee (2006, 2008) studied the technological development of information and energy technology with a broad functional category approach. They assessed three different functional categories including storage, transportation, and transformation for both technology domains and developed associated non-device functional performance metrics, which are the generic function measures for a technology domain (Benson & Magee, 2015), to analyze their technological improvement trends. They found exponential growth for all of these in technological progress throughout the years. Similarly, Magee et al. (2016) looked at 28 different technology domains and employed 71 metrics to study how their performance evolves with time. The results show that these 28 technology domains also seem to exhibit an exponential rate of progress, similar to Moore's Law for electronic chips.

7. CONSIDERATIONS OF FACTORS AFFECTING TECHNOLOGICAL ADVANCEMENT

As discussed above, considerable research has demonstrated that the performance of a broad range of technologies increases exponentially over time.

But it remains inconclusive whether the generalized form of Moore's law provides the most accurate possible predictions for technological performance. Moreover, the generalized Moore's law alone cannot explain what influences technological improvements. As MacDonald and Schrattenholzer (2001) stated: "technology design that is left on the shelf does not become better the longer it sits unused." This suggests the need for knowing the factors that shape technology performance in order to build a better model for predicting technological performance.

Prior research has explored various explanatory variables that affect technological progress. Some studies have measured the technological improvement rates based on insights obtained from patents (Benson & Magee, 2014; Benson & Magee, 2015; Magee et al., 2016; Triulzi et al., 2018; Singh et al., 2020), while others suggested using production, R&D spending, sales revenue, etc. (Bailey et al., 2012; Farmer & Lafond, 2016; Magee et al., 2016). Some scholars took a broader view, looking at the driving forces behind technological innovation, which comprise pull drivers, such as market demand and profitability potential, and push drivers, such as new technology and innovative manufacturing processes (Dosi, 1982; Mack, 2011). They argued that these two drivers are indispensable for technology evolution and a lack of either would change the path of technological advancement.

A number of researchers have emphasized the importance of public policies and government funding for the evolution of certain technologies (Ball, 2013; Trancik & Cross-Call, 2013). For many such technologies, the private sector may have limited incentive or ability to support research because the time needed to

obtain benefit from the research results is often measured in years, if not decades, and they cannot assume the risks inherent in such research, nor can they continue funding research in a particular field over a long period of time if the payback is unclear. In our view, the importance of continued government funding in space exploration technology cannot be emphasized enough as space exploration is a long-term endeavor and requires steady funding to stay on schedule. A decreased level of funding would certainly result in delays in the development of new technology. This motivates our consideration of NASA's budget in modeling advancement in space exploration technology. Additionally, NASA's budget acts as both push and pull drivers for the advancement of space exploration technology. Not only does NASA funded research facilitate the creation of new technology (a push driver), but NASA funded missions provide the applications (a pull driver) that the newly created technology enables.

8. METHODOLOGY

The space mission data for the period 1959 to 2020 is updated from the work of Hall et al. (2017). NASA's annual budget data, measured in US\$ millions, was extracted from Wikipedia (Wikipedia, 2020e). We use the data from the column of 2014 Constant Dollars, which takes inflation into account and will provide a more meaningful assessment than the data from the column of Nominal Dollars. Our study fits the following log linear model to the data:

$$\log_2 L_t = \alpha + \beta_1 t + \beta_2 \log_2 S_t + \varepsilon$$

The dependent variable L_t represents the maximum lifespan of the spacecraft launched in a particular year. A lifespan is calculated by subtracting the launch date of a spacecraft from its failure date and is measured in years. Variable t represents the year that a spacecraft launched. It ranges from 1959 to 1999 to account for the years for which all spacecraft launched are no longer operating (except for Voyager 1 and 2 launched in 1977). S_t represents the moving averages of NASA's annual budget calculated at year t and is measured in US\$ millions. A moving average length of 3 (a window size of 3), for instance, would place the first moving average value at period 3, which is year 1961, the next at period 4, which is 1962, and so on. The parameters α , β_1 , and β_2 are fitted using the method of ordinary least squares. The error terms, ε , are generated about the fitted curve to be used for measuring the goodness-of-fit once the curve is fitted. We assume that the error terms exhibit a mean of zero and constant variance.

To illustrate how we calculate the moving averages of NASA's annual budget, suppose that the data begins with the numbers 10, 20, 30, 40, 50 and we use a window size of 3. The first value of the moving average is supposed to be missing but we replace it with the average of the first value, which is 10 itself; the second value of the moving average again is supposed to be missing but we replace it with the average of the first two values, which is 15; the third value of the moving average is the average of 10, 20, 30, where we start to apply the

window size of 3; the fourth value is then the average of 20, 30, 40 and the fifth uses 30, 40, 50.

We estimated the fit for each window size from 1 to 20 and used the root-mean-square error (RMSE) to measure the goodness-of-fit of the model. Then the resulted RMSE values are compared and the best fit is determined based on the lowest RMSE, which also determines the optimal window size.

9. RESULTS

The plots of RMSE vs. window size allow us to find the best fit to the data when the lowest RMSE is identified. Figure 1 shows the minimum RMSE over various window sizes. The minimum RMSE starts at 2.2342 and decreases as the window size increases to a low of 2.1383 at a window size of 9 before increasing again to 2.1608 at a window size of 20. For comparisons, we also include the two Voyager spacecraft launched in 1977 that are still operating into our analysis. According to NASA's prediction (NASA Jet Propulsion Laboratory, 2020), the Voyagers may continue operation up to year 2025, which would enable both spacecraft a lifespan of 48 years. Figure 2 shows the RMSE over various window sizes for the data with the Voyager mission included assuming both spacecraft have a lifespan of 48 years. It can be seen that the optimal window size changes to 13 with the lowest RMSE of 2.2230.

Figure 1. Minimum RMSE vs. Window Size (Exclusive of the Voyagers)

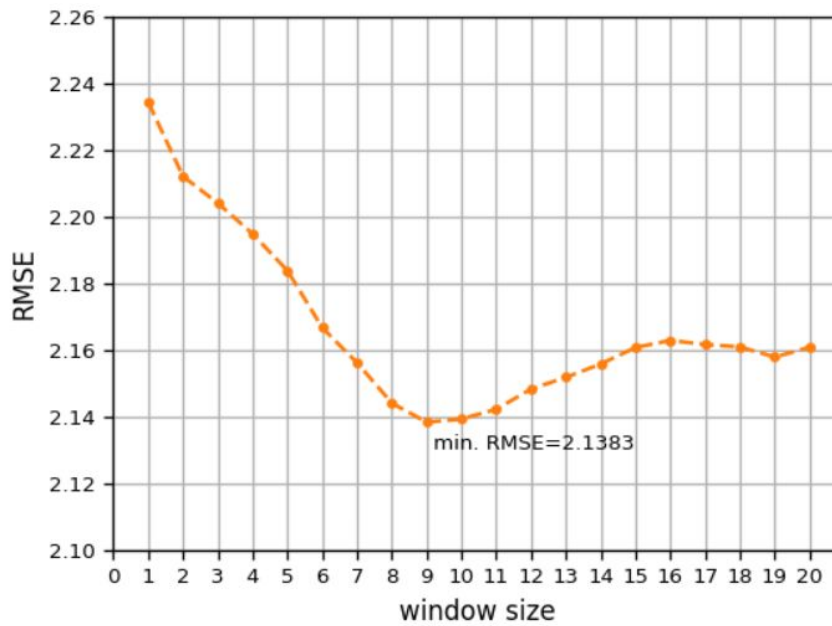
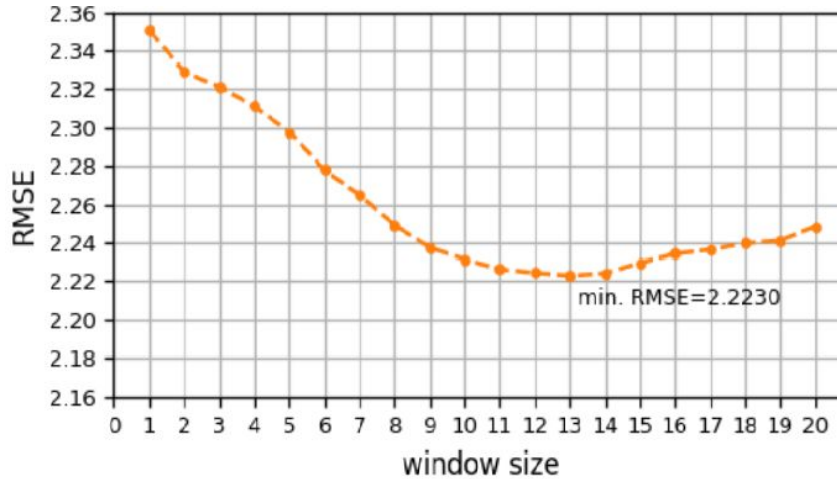


Figure 2. Minimum RMSE vs. Window Size (Inclusive of the Voyagers)



The figures above show that applying a moving average with window size 13 and 9 on NASA's annual budget (Variable S_t) would best fit the data with and without the Voyagers included respectively. Our next step is, therefore, regress

the logarithm of the maximum lifespan of the spacecraft on both the launch year and the logarithm of the moving average of NASA's annual budget in 2014 constant dollars using a window size of 13 on the data inclusive of the Voyagers and using a window size of 9 on the data exclusive of the Voyagers separately.

Table 2 and Table 3 show the fitting results. Without the Voyagers included, a 1% increase in the moving average (window size = 9) of NASA's annual budget in 2014 constant dollars would lead to a 0.9365% rise in the maximum lifespan of the spacecraft, while with the Voyagers included, a 1% increase in the moving average (window size = 13) of NASA's annual budget in 2014 constant dollars would lead to a 1.0593% increase, holding all else constant. All the coefficients for the two fitted curves are significantly different from zero, and the F-statistic indicates that in each model the explanatory variables are jointly significant in the determination of the logarithm of the maximum lifespan of the spacecraft. The results support the view that continued government funding has a significant impact on the progression of space exploration technology.

Table 2. Results of Curve Fitting (Exclusive of the Voyagers)

Variable	Coefficient	t-stat.	p-value
β_1	0.0892	2.343	0.027
β_2	0.9365	2.307	0.029
α	-188.3686	-2.585	0.015

Prob (F-statistic): 0.000583; R-squared: 0.424, Adj R-squared: 0.381

Table 3. Results of Curve Fitting (Inclusive of the Voyagers)

Variable	Coefficient	t-stat.	p-value
β_1	0.0823	2.050	0.050
β_2	1.0593	2.401	0.023
α	-176.3800	-2.301	0.029

Prob (F-statistic): 0.000662; R-squared: 0.407, Adj R-squared: 0.365

Figure 3 and 4 show the residuals plotted against the fitted values. The random scatter in both plots is consistent with the assumption of the models that the error terms exhibit a mean of zero and constant variance.

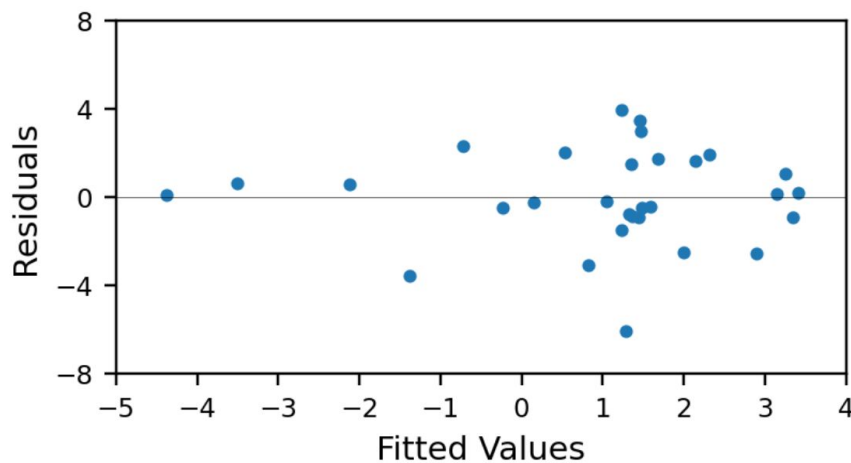
Figure 3. Residual vs. Fitted Value (Exclusive of the Voyagers)

Figure 5. Fitted Curve vs. Moore's Law (Exclusive of the Voyagers)

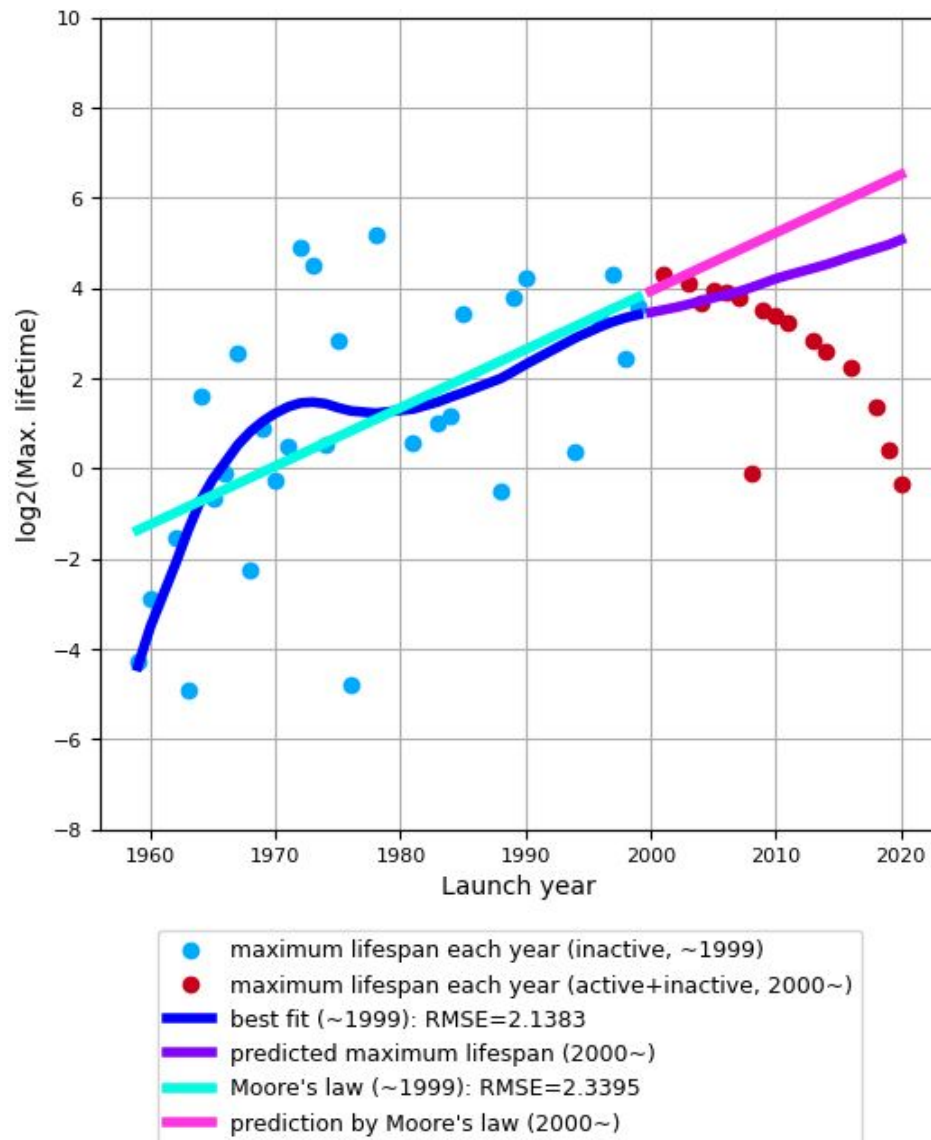


Figure 6. Fitted Curve vs. Moore's Law (Inclusive of the Voyagers)

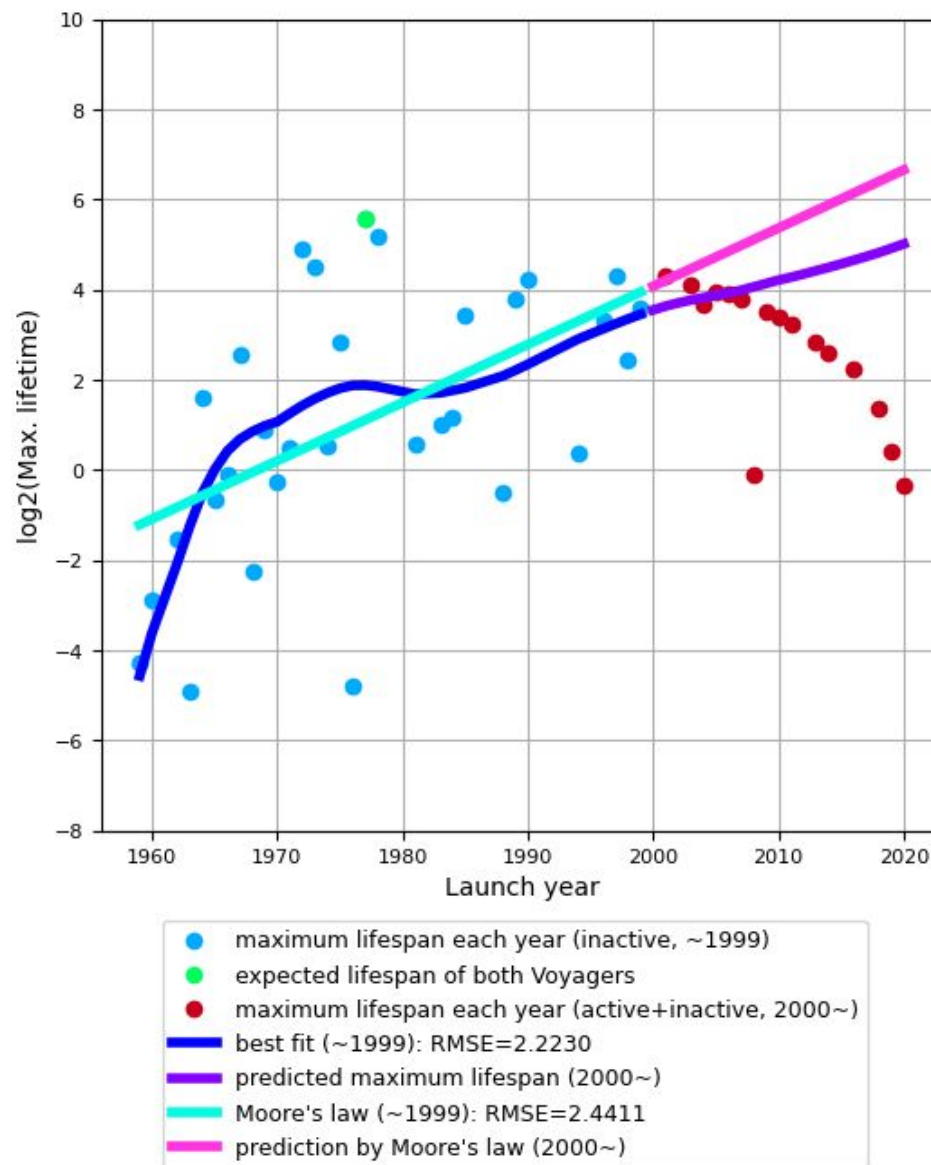


Figure 7. Fitted Curve vs. Moore's Law in linear space

(Exclusive of the Voyagers)

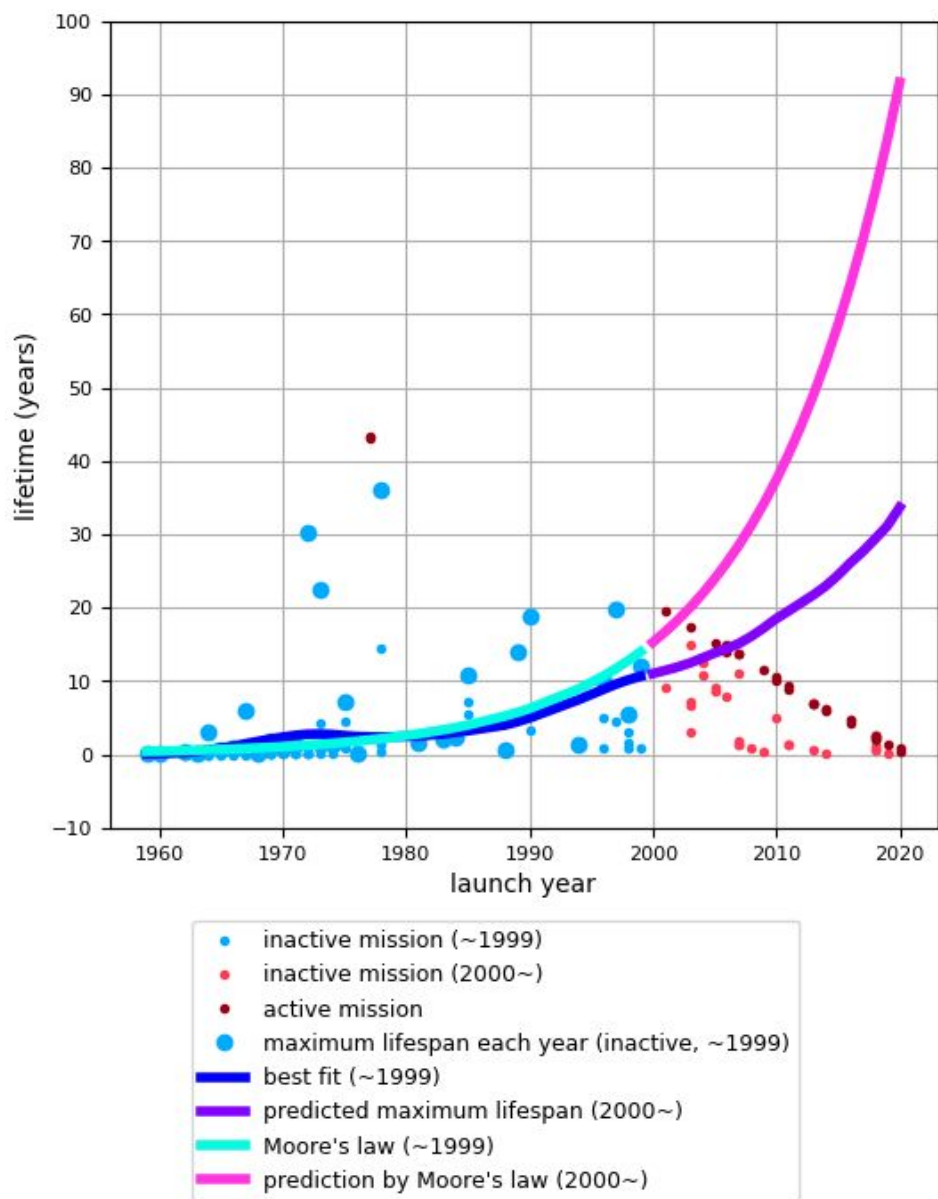
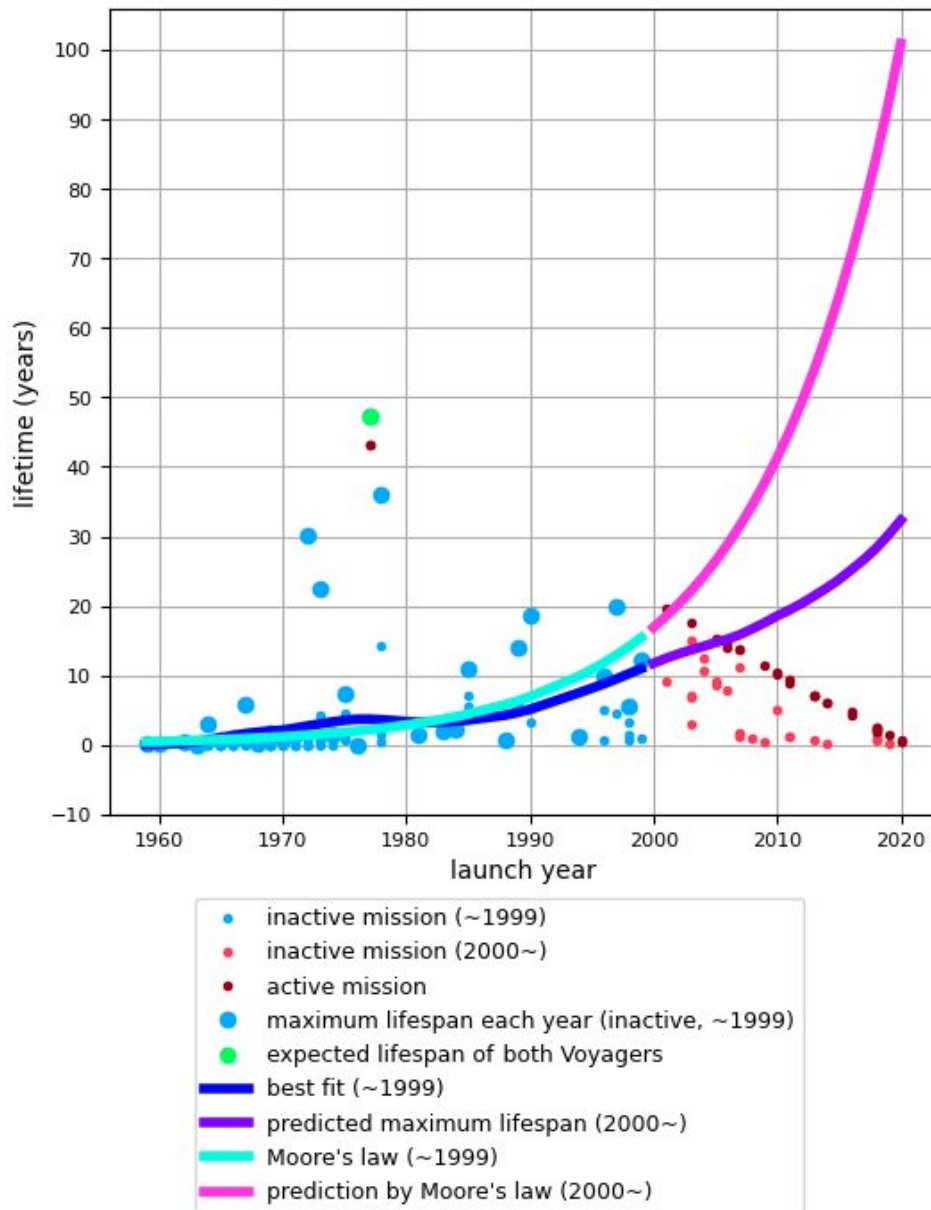


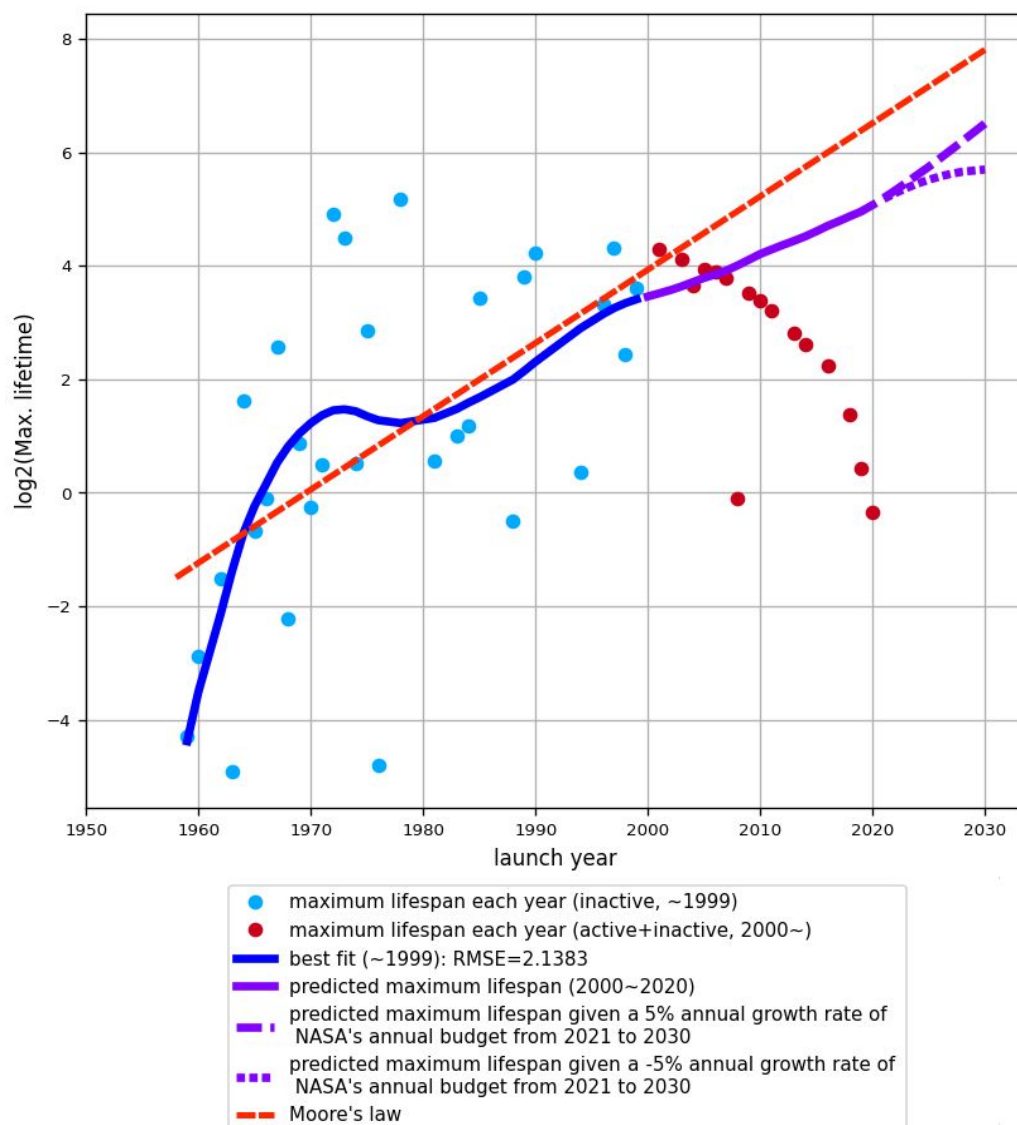
Figure 8. Fitted Curve vs. Moore's Law in linear space
(Inclusive of the Voyagers)



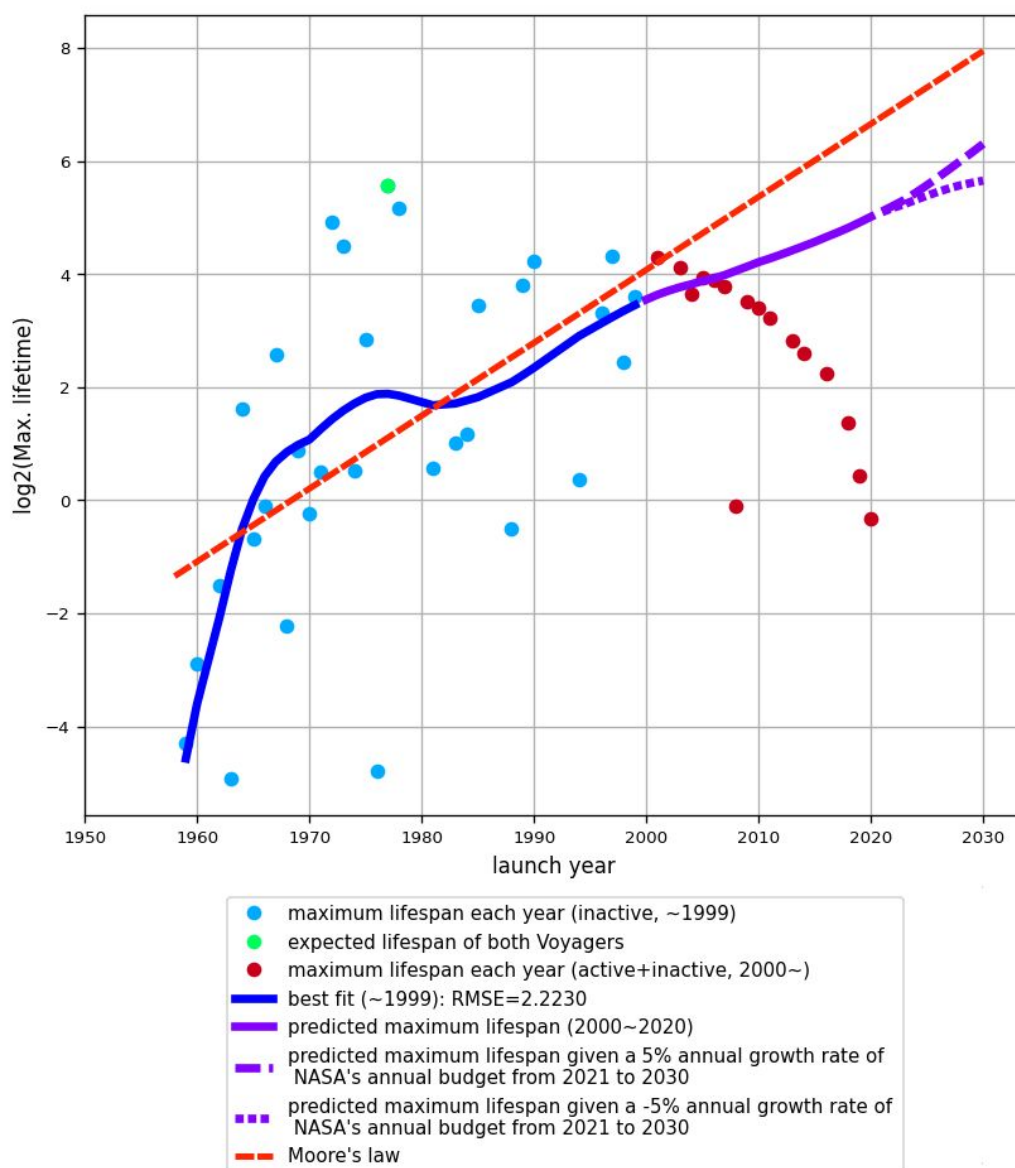
We also estimated NASA's annual budget for the years 2021-2030 using a 5% and -5% annual growth rate respectively to extrapolate the future performance of the maximum lifespan of the spacecraft during the same period

with the fitted models. The extrapolation predictions are illustrated in Figure 9 and Figure 10. The plots demonstrate that the changes in NASA's budget have a significant impact on the advancement of the maximum lifespan of the spacecraft. Figure 11 and Figure 12 show the predictions along with the exponential curve of the generalized Moore's Law in linear space.

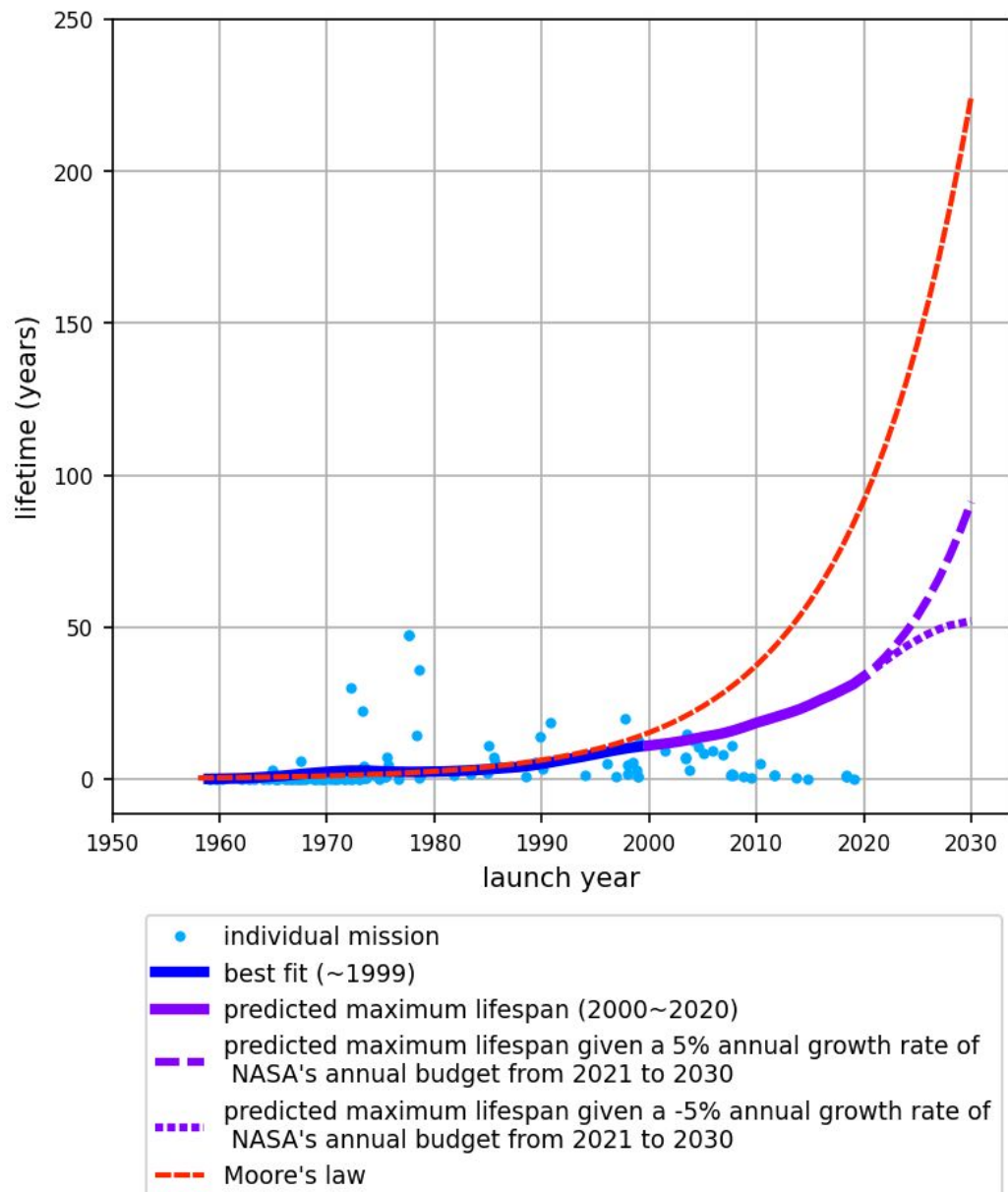
**Figure 9. Predicted maximum lifespans of the spacecraft during 2021-2030
(Exclusive of the Voyagers)**



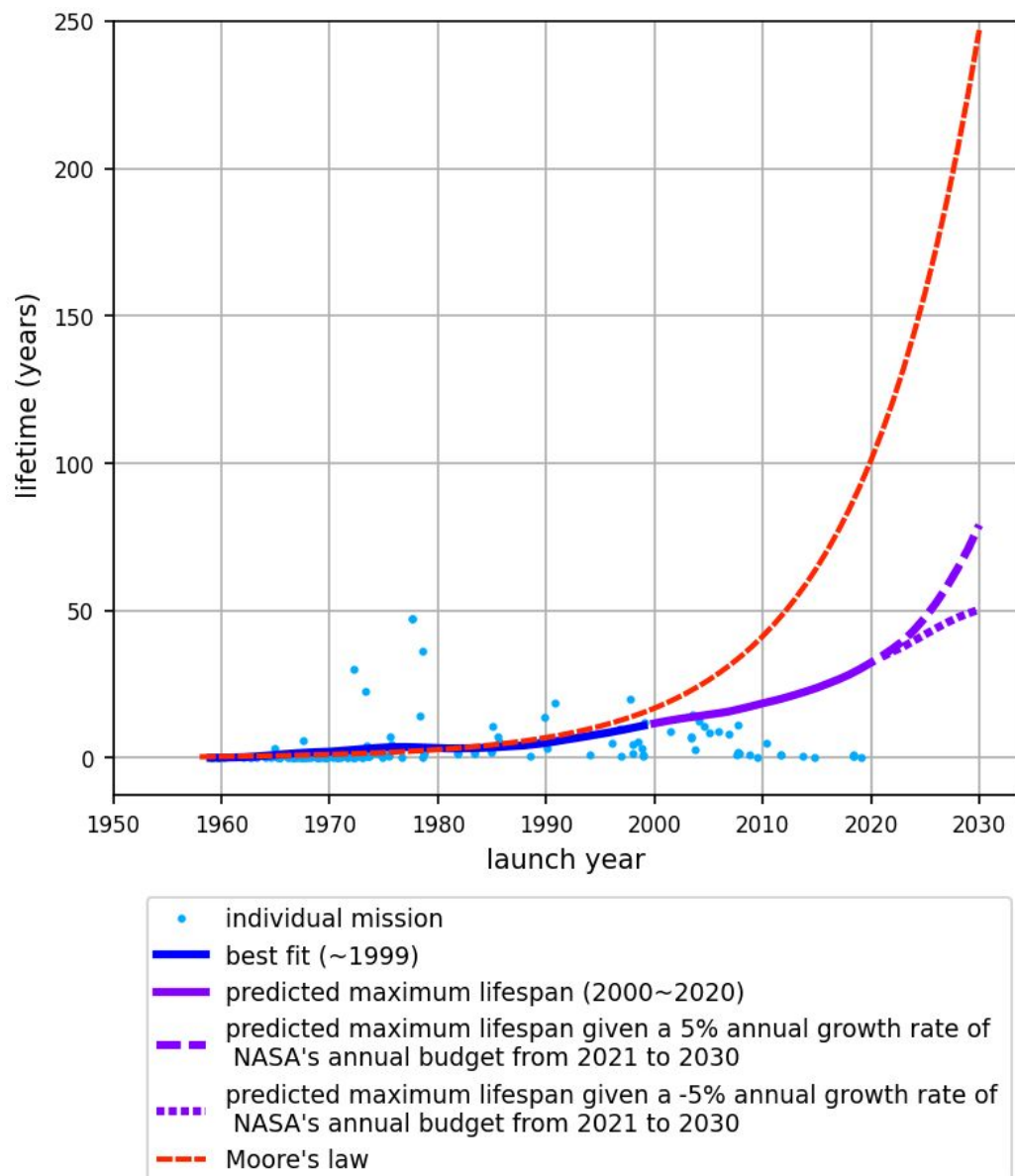
**Figure 10. Predicted maximum lifespans of the spacecraft during
2021-2030 (Inclusive of the Voyagers)**



**Figure 11. Predicted maximum lifespans of the spacecraft during
2021-2030 in linear space (Exclusive of the Voyagers)**



**Figure 12. Predicted maximum lifespans of the spacecraft during
2021-2030 in linear space (Inclusive of the Voyagers)**



REFERENCES

- amCharts (2019). *amCharts V4 Tutorial*. Retrieved 2020-03-02, from
<https://www.amcharts.com/docs/v4/>
- Bailey, A., Bui, Q.M., Farmer, J., Margolis, R., & Ramesh, R. (2012). Forecasting technological innovation. *ARCS Workshops (ARCS)*, pp. 1–6
- Ball, P. (2013). Moore's Law Found to Apply to Evolution of Technologies Beyond Transistors. *Scientific American*. Retrieved 2020-10-20, from
<https://www.scientificamerican.com/article/moores-law-found-to-apply-beyond-transistors>
- Bardi, U. (2016). John Glenn (1921-2016): the end of an era. *Cassandra's Legacy*. Retrieved 2020-10-20, from
<https://cassandralegacy.blogspot.com/2016/12/john-glenn-1921-2016-end-of-era.html>
- Benson, C.L., & Magee, C.L. (2014). On improvement rates for renewable energy technologies: Solar PV, wind turbines, capacitors, and batteries. *Renewable Energy*, 2014; 68, 745–751
- Benson, C.L., & Magee, C.L. (2015). Quantitative Determination of Technological

Improvement from Patent Data. *PlosS ONE*, 10(4):e0121635

doi:10.1371/journal.pone.0121635

Berleant, D., Kodali, V., Segall, R., Aboudja, H., & Howell, M. (2019). Moore's law, Wright's law, and the countdown to exponential space. *The Space Review*, 2019, Paper 3632/1.

Boggarapu, R. (2016). *Interactive JavaScript Charts Using Data from Google Sheets*. Retrieved 2020-02-01, from <https://www.sitepoint.com/interactive-javascript-charts-using-data-from-google-sheets/>

Chart.js (2019). *Chart.js V2.9.3 Tutorial*. Retrieved 2020-02-24, from <https://www.chartjs.org/docs/latest/>

DataTables (2020). *DataTables 1.10+ Tutorial*. Retrieved 2020-02-08, from <https://datatables.net/manual/>

Dosi, G. (1982). Technological paradigms and technological trajectories. *Research Policy*, 11 (3), 147–162.

Elliott, G. (2014). Figure 4: the population of space. *Nefarious Plots*. Retrieved 2020-10-08, from <http://www.nefariousplots.com/figures/4>

Estdale, J., & Georgiadou, E. (2018). Applying the ISO/IEC 25010 Quality Models to Software Product. *European Conference on Software Process Improvement*. Springer, Cham, 896, 492–503. doi: https://doi.org/10.1007/978-3-319-97925-0_42.

Farmer, J.D., & Lafond, F. (2016). How Predictable is Technological Progress? *Research and Policy*, 45(3), pp. 647–665.

Hall, C., Berleant, D., Segall, R., & Lu, S. (2017). Steps toward quantifying advancement in space exploration. *Proceedings of WMSCI*.

Handsontable (2020). *Handsontable 7.4.2 Tutorial*. Retrieved 2020-03-02, from <https://handsontable.com/docs/7.4.2/tutorial-introduction.html>

Hicks, K. (2015). Astronomy: It's sad that human space exploration has stalled. *Columbus Dispatch*. Retrieved 2020-10-20, from <https://www.dispatch.com/article/20151106/NEWS/311069676>

Howell, M., Kodali, V., Kreinovich, V., Aboudja, H., Batthula, V.J.R., Segall, R., & Berleant, D. (2019a). Trends in technology development in the US and USSR during the Space Race. *The Space Review*, 2019, Paper 3840/1.

Howell, M., Kodali, V., Segall, R., Aboudja, H., & Berleant, D. (2019b). Moore's Law and Space Exploration: New Insights and Next Steps. *Journal of the Arkansas*

Academy of Science, 73(Article 6):13-17.

International Organization for Standardization. (2011). *_Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models_(ISO Standard No. 25010:2011(en))*. Retrieved 2020-04-10, from <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>

Kodali, V., Duggirala, R., Segall, R., Aboudja, H., & Berleant, D. (2018). Travel to extraterrestrial destinations over time: some exploratory analyses of mission data. *Journal of the Arkansas Academy of Science*, 72(Article 13):58-66.

Koh, H., & Magee, C.L. (2006). A functional approach for studying technological progress: application to information technology. *Technological Forecasting and Social Change*, 73:1061–1083

Koh, H., & Magee, C.L. (2008). A functional approach for studying technological progress: extension to energy technology. *Technological Forecasting and Social Change*, 75:735–758

Lafond, F., Bailey, A.G., Bakker, J.D., Rebois, D., Zadourian, R., McSharry, P., & Farmer, J.D. (2018). How well do experience curves predict technological progress? A method for making distributional forecasts. *Technological Forecasting and Social Change*, 128:104-117.

- MacDonald, A., & Schrattenholzer, L. (2001). Learning rates for energy technologies. *Energy Policy*, Vol. 29, Issue 4, 255–261.
- Mack, C.A. (2011). Fifty years of Moore's law. *IEEE Transactions on Semiconductor Manufacturing*, 24(2), 202–207.
- Magee, C.L., Basnet, S., Funk, J.L., & Benson, C.L. (2016). Quantitative empirical trends in technical performance. *Technological Forecasting and Social Change*, 104:237-246.
- Mollick, E.R. (2006). Establishing Moore's Law. *IEEE Annals of the History of Computing*, 28: 62-75.
- Moore, G.E. (1965). Cramming More Components onto Integrated Circuits. *Electronics*, Vol. 38, No. 8, 114-117.
- Nagy, B., Farmer, J.D., Bui, Q.M., & Trancik, J.E. (2013). Statistical basis for predicting technological progress. *PLoS One*, 8(2):e52669.
- NASA Jet Propulsion Laboratory (2020). *Did You Know?* Retrieved 2020-10-28, from <https://voyager.jpl.nasa.gov/mission/did-you-know>
- Nielsen, J. (1998). *Nielsen's Law of Internet Bandwidth*. Retrieved 2020-10-21, from

<http://www.nngroup.com/articles/law-of-bandwidth>

Singh, A., Triulzi, G., & Magee, C.L. (2020). Technological improvement rate estimates for all technologies: Use of patent data and an extended domain description. *arXiv:2004.13919 [econ.GN]*.

Singularity Symposium (2020). *What is Moore's Law?* Retrieved 2020-02-12, from <http://www.singularitysymposium.com/moores-law.html>

SpacePunx (2016). *File:Space Population.png (Population of space over time)*.

Retrieved 2020-10-08, from

https://en.wikipedia.org/wiki/File:Space_Population.png

Specter, M. (2009). A life of its own: Where will synthetic biology lead us? *The New Yorker*. Retrieved 2020-10-20, from <https://math.dartmouth.edu/~rockmore/LimitsSupplementalReading/NewYorkerSyntheticBiology.pdf>

Sood, A., James, G.M., Tellis, G.J., & Zhu, J. (2012). Predicting the path of technological innovation: SAW vs. Moore, Bass, Gompertz, and Kryder. *Marketing Science*, 2012;31(6):964–979.

Trancik, J.E., & Cross-Call, D., (2013). Energy technologies evaluated against climate targets using a cost and carbon trade-off curve. *Environmental Science &*

Technology, 47, 6673–6680.

Triulzi, G., Alstott, J., & Magee, C.L. (2018). Estimating Technology Performance Improvement Rates by Mining Patent Data. *SSRN Electronic Journal*, 2018:1-54.

W3Schools (2020a). *HTML Tutorial*. Retrieved 2020-02-05, from
<https://www.w3schools.com/html/default.asp>

W3Schools (2020b). *CSS Tutorial*. Retrieved 2020-02-12, from
<https://www.w3schools.com/css/default.asp>

W3Schools (2020c). *JavaScript Tutorial*. Retrieved 2020-02-08, from
<https://www.w3schools.com/js/default.asp>

W3Schools (2020d). *JS JSON Tutorial*. Retrieved 2020-02-07, from
https://www.w3schools.com/js/js_json_intro.asp

W3Techs (2020). *Usage statistics of client-side programming languages for websites*. Retrieved 2020-02-04, from
https://w3techs.com/technologies/overview/client_side_language

Walter, C. (2005). Kryder's law. *Scientific American*. 293(2):32-33

Wikipedia (2020a). *Web application*. Retrieved 2020-02-22, from

https://en.wikipedia.org/wiki/Web_application

Wikipedia (2020b). *Comparison of JavaScript charting libraries*. Retrieved 2020-02-24, from

https://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_libraries

Wikipedia (2020c). *ISO/IEC 9126*. Retrieved 2020-04-10, from

https://en.wikipedia.org/wiki/ISO/IEC_9126

Wikipedia (2020d). *jQuery*. Retrieved 2020-02-22, from

https://en.wikipedia.org/wiki/JQuery#cite_note-:0-4

Wikipedia (2020e). *Budget of NASA*. Retrieved 2020-10-30, from

https://en.wikipedia.org/wiki/Budget_of_NASA

APPENDIX

```
// HTML code

<!doctype html>

<html lang="en-US">


    <div class="loading">

        </div>

    <div class="app">

</div>


<head>


    <style>

        @import url(https://fonts.googleapis.com/css?family=Lato);

        body, html {

            height: 100%;

            margin: 0;

            font-family: Lato;

        }
```

```
.app {  
    display:none;  
}  
  
.loading {  
    width: 100%;  
    height: 100%;  
    margin: 0;  
    position: fixed;  
    top: 35%;  
    left: 45%;  
}  
  
.tabs {  
    overflow: auto;  
    height: 35px;  
    background-color: #d4d4d4;  
}  
  
.stripe {  
    overflow: auto;  
    height: 40px;  
    background-color: #e0e0e0;  
}
```

```
.mr-1 {  
    margin-right: 0.25rem !important;  
}
```

```
/* Style individual tabs (class="tabsection") */
```

```
.tabsection {  
    background-color: inherit;  
    border: 1px solid #f0f0f0;  
    outline: none;  
    cursor: pointer;  
    padding: 4px 12px;  
    font-size: 16px;  
    width: 20%;  
}
```

```
/* Change background color of the tab on hover */
```

```
.tabsection:hover {  
    background-color: #cfcfcf;  
}
```

```
/* Style the tab content */
```

```
.tabcontent {  
    display: none;  
    background-color: #f0f0f0;
```

```
padding-top: 5px;  
}
```

```
.card-header {  
padding: 0.75rem 1.25rem;  
font-size: 8px;  
margin-bottom: 0;  
background-color: rgba(0, 0, 0, 0.03);  
border-bottom: 1px solid rgba(0, 0, 0, 0.125);  
}
```

```
.card-body {  
flex: 1 1 auto;  
min-height: 1px;  
padding: 1.25rem;  
}
```

```
table, tr, th, td {  
margin: 2px;  
}
```

```
#chartdiv11 {  
width: 100%;  
height: 380px;
```



```
        border: 1px solid #cccccc;  
    }  
  

```

```
#chartdiv12 {  
    width: 100%;  
    height: 380px;  
    border: 1px solid #cccccc;  
}  
  

```

```
#chartdiv13 {  
    width: 100%;  
    height: 380px;  
    border: 1px solid #cccccc;  
}  
  

```

```
#chartdiv10 {  
    width: 100%;  
    height: 380px;  
    border: 1px solid #cccccc;  
}  
  

```

```
#chartdiv41 {  
    width: 50%;  
    height: 400px;  
  

```

```
        border: 1px solid #cccccc;
    }

    #chartdiv42 {
        width: 80%;
        height: 800px;
        border: 1px solid #cccccc;
    }

    #chartdiv43 {
        width: 40%;
        height: 700px;
        border: 1px solid #cccccc;
    }

    #chartdiv44 {
        width: 55%;
        height: 500px;
        border: 1px solid #cccccc;
    }
</style>

<meta charset="utf-8">

<title>Curve Fitting App.</title>
```

```

<!-- jQuery -->

<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>

<!-- Handsontable -->

<script
src="https://cdnjs.cloudflare.com/ajax/libs/handsontable/0.20.2/handsontable.full.
min.js"></script>

<link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/handsontable/0.20.2/handsontable.full
.min.css">

<link rel="stylesheet" type="text/css"
href="https://handsontable.com/static/css/main.css">

<!-- amCharts -->

<script src="https://www.amcharts.com/lib/3/amcharts.js"></script>

<script src="https://www.amcharts.com/lib/3/xy.js"></script>

<script src="https://www.amcharts.com/lib/3/serial.js"></script>

<script src="https://www.amcharts.com/lib/3/themes/light.js"></script>

<script
src="https://www.amcharts.com/lib/3/plugins/dataloader/dataloader.min.js"></scri
pt>

<script
src="//www.amcharts.com/lib/3/plugins/tools/datePadding/datePadding.min.js"><
/script>

<script

```

```
src="https://www.amcharts.com/lib/3/plugins/tools/bestFitLine/bestFitLine.min.js"
```

```
></script>
```

```
    <script src="https://www.amcharts.com/lib/3/plugins/export/export.js"
```

```
type="text/javascript"></script>
```

```
    <link href="https://www.amcharts.com/lib/3/plugins/export/export.css"
```

```
rel="stylesheet" type="text/css">
```

```
</head>
```

```
<body>
```

```
    <h1 style="font-size:22px;">&nbsp;Curve Fitting<a class="button butn"
```

```
href="https://docs.google.com/spreadsheets/d/1SwzNQ2rWdOnDPPc40LhVc0D
```

```
NBXpOcCtn0qQz2C991VQ/edit#gid=846789546" target="_blank">
```

```
    &nbsp;Click here for the Data Sets.</a>
```

```
</h1>
```

```
<section>
```

```
    <div class="tabs">
```

```
        <button class="tabsection" onclick="openPage('Launch',
this, '#fcfcfc')" id="defaultOpen">Exploring different curve fitting</button>
```

```
        <button class="tabsection" onclick="openPage('Project',
this, '#fcfcfc')">Results of fitting</button>
```

```
    </div>
```

```
</section>
```

```
<div id="Launch" class="tabcontent">
```

```
  <table>
```

```
    <th><div id="chartdiv11"></div></th>
```

```
    <th><div id="chartdiv12"></div></th>
```

```
    <th><div id="chartdiv13"></div></th>
```

```
    <th><div id="chartdiv10"></div></th>
```

```
  </table>
```

```
  <section class="stripe">
```

```
    <h2 style="font-size:15px;">&nbsp;RMSE:</h2>
```

```
  </section>
```

```
  <div id="hTable1"></div>
```

```
  <section class="stripe">
```

```
    <h2 style="font-size:15px;">&nbsp;Source Data:</h2>
```

```
  </section>
```

```
  <div id="hTable2"></div>
```

```
</div>
```

```
<div id="Project" class="tabcontent">
```

```

<section class="stripe">
    <h2 style="font-size:15px;">&nbsp;Lowest RMSE (the
best fit)</h2>
</section>
<div class="card-body">
    
</div>
<section class="stripe">
    <h2 style="font-size:15px;">&nbsp;Fitted curve v.s
Moore's law in log space</h2>
</section>
<div class="card-body">
    
</div>
<section class="stripe">
    <h2 style="font-size:15px;">&nbsp;Fitted curve v.s
Moore's law in linear space</h2>
</section>
<div class="card-body">
    

</div>

<section class="stripe">

<h2 style="font-size:15px;">&nbsp;Model summary</h2>

</section>

<div class="card-body">



</div>

<section class="stripe">

<h2 style="font-size:15px;">&nbsp;Click the image to  
access the curve fitting algorithms in Python on Google Colab (Sign-in ID:  
spacemissions20@gmail.com / Password: spacemissions)</h2>

</section>

<div class="card-body">

<iframe

src="https://drive.google.com/embeddedfolderview?id=1RurleTzpqqJiB9DqUbrOIKXdvj0CtpdO#grid" frameborder="0" width="240" height="280"  
marginheight="0" marginwidth="0"></iframe>

</div>

</div>

```
<script>
```

```

$(function() {
 $(".loading").fadeOut(1000, function() {
 $(".app").fadeIn(1000);
 });
});

```

```

function openPage(tabName, buttOpen, color) {
 // Hide all class="tabcontent" elements
 var i, tabcontent, tabsection;
 tabcontent =
document.getElementsByClassName("tabcontent");
 for (i = 0; i < tabcontent.length; i++) {
 tabcontent[i].style.display = "none";
 }

 // Remove the background color of the tabs
(class="tabsection")
 tabsection =
document.getElementsByClassName("tabsection");
 for (i = 0; i < tabsection.length; i++) {
 tabsection[i].style.backgroundColor = "";
 }

```



```

// Show the specific tab content
document.getElementById(tabName).style.display =
"block";

// Add the specific color to the button used to open the
tab content
buttOpen.style.backgroundColor = color;
}

// Get the element with id="defaultOpen" and click on it
document.getElementById("defaultOpen").click();

// Build functions to create the interactive charts.

function createChart10(inputdata) {

 var chart = AmCharts.makeChart("chartdiv10", {
 "type": "serial",
 "theme": "light",
 "dataProvider": inputdata,
 "resizeCategoryWidth": 2.72,
 "marginTop": 5,
 "marginBottom": 0,

```

```
"marginRight": 15,
"marginLeft": 0,
"titles": [{
 "text": "New Curve Fitting",
 "size": 10
}],
"categoryField": "launchdate",
"categoryAxis": {
 "title": 'launch date',
 "titleFontSize": 10,
 "fontSize": 9,
 "autoGridCount": false,
 "gridCount": 20,
 "parseDates": true,
 "axisThickness": 2,
 "labelRotation": 30
},
"valueAxes": [{
 // "logarithmic": true,
 "titleFontSize": 10,
 "fontSize": 9,
 "gridAlpha": 0.07,
 "position": "left",
 "title": "log2(lifetime)",
```

```

 "axisThickness": 2
 }},
 "graphs": [{
 "id": "g1",
 "title": "log2(Lifetime)",
 "valueField":
"log2lifetimeinlaunchdateorder",
 "lineThickness": 0,
 "bullet": "round",
 "bulletSize": 4,
 },
 {
 "title": "New Fit 1",
 "valueField": "newfit1",
 "type": "line",
 "lineThickness": 2,
 "lineColor": "#3d0101",
 "fillAlphas": 0
 },
 {
 "title": "New Fit 2",
 "valueField": "newfit2",
 "type": "line",
 "lineThickness": 2,

```

```

 "lineColor": "#001773",
 "fillAlphas": 0
 },
 {
 "title": "New Fit 3",
 "valueField": "newfit3",
 "type": "line",
 "lineThickness": 2,
 "lineColor": "#065700",

 "fillAlphas": 0
 }
}],
"chartScrollbar": {
 "graph": "g1",
 "oppositeAxis": true,
 "offset": 8,
 "scrollbarHeight": 20,
 "backgroundAlpha": 0,
 "selectedBackgroundAlpha": 0,
 "selectedBackgroundColor": "#000000",
 "graphFillAlpha": 0,
 "graphLineAlpha": 0.5,
 "selectedGraphFillAlpha": 0,
 "selectedGraphLineAlpha": 0.5,

```

```

 "autoGridCount": false,
 "gridCount": 8,
 "color": "#7a7a7a"
 },
 "valueScrollbar": {
 "graph": "g1",
 "oppositeAxis": true,
 "offset": 8,
 "scrollbarHeight": 10,
 "backgroundAlpha": 0,
 "selectedBackgroundAlpha": 0,
 "selectedBackgroundColor": "#000000",
 "graphFillAlpha": 0,
 "graphLineAlpha": 0.5,
 "selectedGraphFillAlpha": 0,
 "selectedGraphLineAlpha": 0.5,
 "autoGridCount": false,
 "gridCount": 5,
 "color": "#7a7a7a"
 },
 "chartCursor": {
 "categoryBalloonDateFormat":
"YYYY-MM-DD",
 "cursorAlpha": 0.1,

```

```

 "cursorColor": "#000000",
 "fullWidth": true,
 "valueLineEnabled": true,
 "valueLineBalloonEnabled": true,
 "valueLineAlpha": 0.5,
 "valueBalloonsEnabled": false,
 "zoomable": true
 },
 "legend": {
 "enabled": true,
 "fontSize": 9,
 "position": "bottom",
 "horizontalGap": 0,
 "markerSize": 10,
 "labelWidth": 0,
 "markerLabelGap": 2,
 "valueWidth": 30,
 "verticalGap": 0.5,
 "useGraphSettings": true //Set the markers
 }
 to have similar-look shapes as in the graph.
 },
 "export": {
 "enabled": true
 },
 },

```

```
 });

 return chart;
}

function createChart11(inputdata) {

 var chart = AmCharts.makeChart("chartdiv11", {
 "type": "serial",
 "theme": "light",
 "dataProvider": inputdata,
 "resizeCategoryWidth": 2.72,
 "marginTop": 5,
 "marginBottom": 0,
 "marginRight": 15,
 "marginLeft": 0,
 "titles": [{
 "text": "Exponential Fitting of Lifetimes",
 "size": 10
 }],
 "categoryField": "launchdate",
 "categoryAxis": {
 "title": 'launch date',
 "titleFontSize": 10,
```

```

 "fontSize": 9,
 "autoGridCount": false,
 "gridCount": 20,
 "parseDates": true,
 "axisThickness": 2,
 "labelRotation": 30
 },
 "valueAxes": [{
 // "logarithmic": true,
 "titleFontSize": 10,
 "fontSize": 9,
 "gridAlpha": 0.07,
 "position": "left",
 "title": "lifetime",
 "axisThickness": 2
 }],
 "graphs": [{
 "id": "g1",
 "title": "Lifetime",
 "valueField": "lifetime",
 "lineThickness": 0,
 "bullet": "round",
 "bulletSize": 4,
 },

```



```

{
 "title": "Exponential Fitting",
 "valueField":
"exponentialfittingonlaunchdate",
 "type": "line",
 "lineThickness": 2,
 "lineColor": "#3d0101",
 "fillAlphas": 0
}],
"chartScrollbar": {
 "graph": "g1",
 "oppositeAxis": true,
 "offset": 8,
 "scrollbarHeight": 20,
 "backgroundAlpha": 0,
 "selectedBackgroundAlpha": 0,
 "selectedBackgroundColor": "#000000",
 "graphFillAlpha": 0,
 "graphLineAlpha": 0.5,
 "selectedGraphFillAlpha": 0,
 "selectedGraphLineAlpha": 0.5,
 "autoGridCount": false,
 "gridCount": 8,
 "color": "#7a7a7a"

```

```

 },
 "valueScrollbar": {
 "graph": "g1",
 "oppositeAxis": true,
 "offset": 8,
 "scrollbarHeight": 10,
 "backgroundAlpha": 0,
 "selectedBackgroundAlpha": 0,
 "selectedBackgroundColor": "#000000",
 "graphFillAlpha": 0,
 "graphLineAlpha": 0.5,
 "selectedGraphFillAlpha": 0,
 "selectedGraphLineAlpha": 0.5,
 "autoGridCount": false,
 "gridCount": 5,
 "color": "#7a7a7a"
 },
 "chartCursor": {
 "categoryBalloonDateFormat":
"YYYY-MM-DD",
 "cursorAlpha": 0.1,
 "cursorColor": "#000000",
 "fullWidth": true,
 "valueLineEnabled": true,

```

```

 "valueLineBalloonEnabled": true,
 "valueLineAlpha": 0.5,
 "valueBalloonsEnabled": false,
 "zoomable": true
 },
 "legend": {
 "enabled": true,
 "fontSize": 9,
 "position": "bottom",
 "horizontalGap": 0,
 "markerSize": 10,
 "labelWidth": 0,
 "markerLabelGap": 2,
 "valueWidth": 30,
 "verticalGap": 0.5,
 "useGraphSettings": true //Set the markers
to have similar-look shapes as in the graph.
 },
 "export": {
 "enabled": true
 },
});

return chart;

```

```
}
```

```
function createChart12(inputdata) {
```

```
 var chart = AmCharts.makeChart("chartdiv12", {
```

```
 "type": "serial",
```

```
 "theme": "light",
```

```
 "dataProvider": inputdata,
```

```
 "resizeCategoryWidth": 2.72,
```

```
 "marginTop": 5,
```

```
 "marginBottom": 0,
```

```
 "marginRight": 15,
```

```
 "marginLeft": 0,
```

```
 "titles": [{
```

```
 "text": "Moving Window Average of
```

```
Lifetimes",
```

```
 "size": 10
```

```
 }],
```

```
 "categoryField": "launchdate",
```

```
 "categoryAxis": {
```

```
 "title": 'launch date',
```

```
 "titleFontSize": 10,
```

```
 "fontSize": 9,
```

```
 "autoGridCount": false,
```

```

 "gridCount": 20,
 "parseDates": true,
 "axisThickness": 2,
 "labelRotation": 30
 },
 "valueAxes": [{
 // "logarithmic": true,
 "titleFontSize": 10,
 "fontSize": 9,
 "gridAlpha": 0.07,
 "position": "left",
 "title": "lifetime",
 "axisThickness": 2
 }],
 "graphs": [{
 "id": "g1",
 "title": "Lifetime",
 "valueField": "lifetime",
 "lineThickness": 0,
 "bullet": "round",
 "bulletSize": 4,
 },
 {
 "title": "Moving Window Average of

```

Lifetimes",

"valueField":

"movingwindowaverageoflifetimeinlaunchdateorder",

"type": "line",

"lineThickness": 2,

"lineColor": "#3d0101",

"fillAlphas": 0

}},

"chartScrollbar": {

"graph": "g1",

"oppositeAxis": true,

"offset": 8,

"scrollbarHeight": 20,

"backgroundAlpha": 0,

"selectedBackgroundAlpha": 0,

"selectedBackgroundColor": "#000000",

"graphFillAlpha": 0,

"graphLineAlpha": 0.5,

"selectedGraphFillAlpha": 0,

"selectedGraphLineAlpha": 0.5,

"autoGridCount": false,

"gridCount": 8,

"color": "#7a7a7a"

},

```

"valueScrollbar":{
 "graph": "g1",
 "oppositeAxis": true,
 "offset": 8,
 "scrollbarHeight": 10,
 "backgroundAlpha": 0,
 "selectedBackgroundAlpha": 0,
 "selectedBackgroundColor": "#000000",
 "graphFillAlpha": 0,
 "graphLineAlpha": 0.5,
 "selectedGraphFillAlpha": 0,
 "selectedGraphLineAlpha": 0.5,
 "autoGridCount": false,
 "gridCount": 5,
 "color": "#7a7a7a"
},
"chartCursor": {
 "categoryBalloonDateFormat":
"YYYY-MM-DD",
 "cursorAlpha": 0.1,
 "cursorColor": "#000000",
 "fullWidth": true,
 "valueLineEnabled": true,
 "valueLineBalloonEnabled": true,

```

```

 "valueLineAlpha":0.5,
 "valueBalloonsEnabled": false,
 "zoomable": true
 },
 "legend": {
 "enabled": true,
 "fontSize": 9,
 "position": "bottom",
 "horizontalGap": 0,
 "markerSize": 10,
 "labelWidth": 0,
 "markerLabelGap": 2,
 "valueWidth": 30,
 "verticalGap": 0.5,
 "useGraphSettings": true //Set the markers
to have similar-look shapes as in the graph.
 },
 "export": {
 "enabled": true
 },
});

return chart;
}

```



```

function createChart13(inputdata) {

 var chart = AmCharts.makeChart("chartdiv13", {
 "type": "serial",
 "theme": "light",
 "dataProvider": inputdata,
 "resizeCategoryWidth": 2.72,
 "marginTop": 5,
 "marginBottom": 0,
 "marginRight": 15,
 "marginLeft": 0,
 "titles": [{
 "text": "Moving Window Average of
log2(Lifetime)",
 "size": 10
 }],
 "categoryField": "launchdate",
 "categoryAxis": {
 "title": 'launch date',
 "titleFontSize": 10,
 "fontSize": 9,
 "autoGridCount": false,
 "gridCount": 20,

```

```

 "parseDates": true,
 "axisThickness": 2,
 "labelRotation": 30
 },
 "valueAxes": [{
 // "logarithmic": true,
 "titleFontSize": 10,
 "fontSize": 9,
 "gridAlpha": 0.07,
 "position": "left",
 "title": "log2(lifetime)",
 "axisThickness": 2
 }],
 "graphs": [{
 "id": "g1",
 "title": "log2(Lifetime)",
 "valueField":
"log2lifetimeinlaunchdateorder",
 "lineThickness": 0,
 "bullet": "round",
 "bulletSize": 4,
 "legendValueText": "[[value]]"
 },
 {

```

```

 "title": "Moving Window Average of
log2(Lifetime)",

 "valueField":

"movingwindowaverageoflog2lifetimeinlaunchdateorder",

 "type": "line",

 "lineThickness": 2,

 "lineColor": "#3d0101",

 "fillAlphas": 0

 }},

 "chartScrollbar": {

 "graph": "g1",

 "oppositeAxis": true,

 "offset": 8,

 "scrollbarHeight": 20,

 "backgroundAlpha": 0,

 "selectedBackgroundAlpha": 0,

 "selectedBackgroundColor": "#000000",

 "graphFillAlpha": 0,

 "graphLineAlpha": 0.5,

 "selectedGraphFillAlpha": 0,

 "selectedGraphLineAlpha": 0.5,

 "autoGridCount": false,

 "gridCount": 8,

```

```

 "color": "#7a7a7a"
 },
 "valueScrollbar": {
 "graph": "g1",
 "oppositeAxis": true,
 "offset": 8,
 "scrollbarHeight": 10,
 "backgroundAlpha": 0,
 "selectedBackgroundAlpha": 0,
 "selectedBackgroundColor": "#000000",
 "graphFillAlpha": 0,
 "graphLineAlpha": 0.5,
 "selectedGraphFillAlpha": 0,
 "selectedGraphLineAlpha": 0.5,
 "autoGridCount": false,
 "gridCount": 5,
 "color": "#7a7a7a"
 },
 "chartCursor": {
 "categoryBalloonDateFormat":
"YYYY-MM-DD",
 "cursorAlpha": 0.1,
 "cursorColor": "#000000",
 "fullWidth": true,

```

```

 "valueLineEnabled":true,
 "valueLineBalloonEnabled":true,
 "valueLineAlpha":0.5,
 "valueBalloonsEnabled": false,
 "zoomable": true
 },
 "legend": {
 "enabled": true,
 "fontSize": 9,
 "position": "bottom",
 "horizontalGap": 0,
 "markerSize": 10,
 "labelWidth": 0,
 "markerLabelGap": 2,
 "valueWidth": 30,
 "verticalGap": 0.5,
 "useGraphSettings": true //Set the markers
to have similar-look shapes as in the graph.
 },
 "export": {
 "enabled": true
 },
});

```

```

 return chart;
 }

 AmCharts.ready(function() {

$.getJSON('https://spreadsheets.google.com/feeds/list/1SwzNQ2rWdOnDPPc40
LhVc0DNBXpOcCtn0qQz2C991VQ/1/public/values?alt=json', function(data) {

 var sheetData = data.feed.entry;

 var array1 = [];

 for (var i = 0; i < sheetData.length; i++) {
 var objson = [

sheetData[i]['gsx$rmseexponentialfitting']['$t'],

sheetData[i]['gsx$rmsemovingwindowaverageoflifetime']['$t'],

sheetData[i]['gsx$rmsemovingwindowaverageoflog2lifetime']['$t'],

sheetData[i]['gsx$rmsevenewfit1']['$t'],

```

```
sheetData[i]['gsx$rmsenewfit2']['$t'],
```

```
sheetData[i]['gsx$rmsenewfit3']['$t'],
```

```
];
```

```
array1.push(objson);
```

```
}
```

```
var array2 = [];
```

```
for (var i = 0; i < sheetData.length; i++) {
```

```
var objson = [
```

```
sheetData[i]['gsx$ordinality']['$t'],
```

```
sheetData[i]['gsx$craft']['$t'],
```

```
sheetData[i]['gsx$launchdate']['$t'],
```

```
sheetData[i]['gsx$failedate']['$t'],
```

```
sheetData[i]['gsx$lifetime']['$t'],
```

```
sheetData[i]['gsx$exponentialfittingonlaunchdate']['$t'],
```

```
sheetData[i]['gsx$movingwindowaverageoflifetimeinlaunchdateorder']['$t'],
```

```
sheetData[i]['gsx$movingwindowstdevoflifetimeinlaunchdateorder']['$t'],
```

```
sheetData[i]['gsx$log2lifetimeinlaunchdateorder']['$t'],
```

```
sheetData[i]['gsx$movingwindowaverageoflog2lifetimeinlaunchdateorder']['$t'],
```

```
sheetData[i]['gsx$newfit1']['$t'],
```

```
sheetData[i]['gsx$newfit2']['$t'],
```

```
sheetData[i]['gsx$newfit3']['$t'],
```

```
];
```

```
array2.push(objson);
```

```
}
```

```
var dataSM1 = array1,
```

```
dataSM2 = array2,
```

```
chart10,
```

```
chart11,
```

```
chart12,
```

```
chart13,
```

```
tableElement1 =
```



```

document.querySelector('#hTable1'),

 tableElement2 =

document.querySelector('#hTable2');

var tableSetting1 = {
 data: dataSM1,
 stretchH: 'all',
 width: 1510,
 autoWrapRow: true,
 height: 50,
 maxRows: 1,
 rowHeaders: true,
 colHeaders: [
 '(A) RMSE (Exponential
fitting)',
 '(B) RMSE (Moving window
average of Lifetime)',
 '(C) RMSE (Moving window
average of log2(Lifetime))',
 '(D) RMSE (New fit 1)',
 '(E) RMSE (New fit 2)',
 '(F) RMSE (New fit 3)',
],
 columnSorting: {

```

```

 indicator: true
 },
 multiColumnSorting: {
 indicator: true
 },
 autoColumnSize: {
 samplingRatio: 23
 },
 mergeCells: true,
 contextMenu: true,
 manualRowResize: true,
 manualColumnResize: true,
 manualRowMove: true,
 manualColumnMove: true,
 formulas: true,
 licenseKey:
'non-commercial-and-evaluation',
 };

```

```

var tableSetting2 = {
 data: dataSM2,
 stretchH: 'all',
 width: 1510,
 autoWrapRow: true,

```

```

height: 200,
maxRows: 600,
rowHeaders: true,
colHeaders: [
 '(A) Ordinality',
 '(B) Craft',
 '(C) Launch date',
 '(D) Failure date',
 '(E) Lifetime',
 '(F) Exponential fitting on
launch date',
 '(G) Moving window average
of lifetimes in launch date order',
 '(H) Moving window stdev of
lifetimes in launch date order',
 '(I) log2(Lifetime) in launch
date order',
 '(J) Moving window average
of log2(Lifetime)in launch date order',
 '(K) New fit 1',
 '(L) New fit 2',
 '(M) New fit 3',
],
columnSorting: {

```

```

 indicator: true
 },
 multiColumnSorting: {
 indicator: true
 },
 autoColumnSize: {
 samplingRatio: 23
 },
 mergeCells: true,
 contextMenu: true,
 manualRowResize: true,
 manualColumnResize: true,
 manualRowMove: true,
 manualColumnMove: true,
 dropdownMenu: true,
 filters: true,
 formulas: true,
 licenseKey:
'non-commercial-and-evaluation',

 exportFile: true,
 observeChanges: true,
 "afterInit": function(firstTime) {
 chart11 = createChart11(

```

```
formatChartData(this.getData())
```

```
);
```

```
chart12 = createChart12(
```

```
formatChartData(this.getData())
```

```
);
```

```
chart13 = createChart13(
```

```
formatChartData(this.getData())
```

```
);
```

```
chart10 = createChart10(
```

```
formatChartData(this.getData())
```

```
);
```

```
},
```

```
"afterChange": function(changes,
```

```
source) {
```

```
 if (changes === null)
```

```
 return;
```

```
 chart11.dataProvider =
```

```
formatChartData(this.getData());
```

```
 chart11.validateData();
```

```
 chart12.dataProvider =
```

```
formatChartData(this.getData());
```

```

 chart12.validateData();

 chart13.dataProvider =
formatChartData(this.getData());

 chart13.validateData();

 chart10.dataProvider =
formatChartData(this.getData());

 chart10.validateData();

 },

};

var myTable1 = new
Handsontable(tableElement1, tableSetting1);

var myTable2 = new
Handsontable(tableElement2, tableSetting2);

var button =
document.getElementById("export-csv1");

var exportPlugin =
myTable2.getPlugin("exportFile");

button.addEventListener("click", function() {
 exportPlugin.downloadFile("csv", {
 bom: false,

```

```

 columnDelimiter: ',',
 columnHeaders: true,
 rowDelimiter: '\r\n',
 rowHeaders: false,
 exportHiddenColumns: true,
 exportHiddenRows: true,
 fileExtension: 'csv',
 filename: 'Table_[YYYY]-[MM]-[DD]',
 mimeType: 'text/csv'
 });
});

```

```

function formatChartData(tableData) {
 var chartData = [];
 for(var i = 0; i < tableData.length; i++) {
 chartData.push({
 "ordinality":
 tableData[i][0],
 "craft":
 tableData[i][1],
 "launchdate":
 tableData[i][2],
 "failuredate":
 tableData[i][3],

```

```

 "lifetime":
 tableData[i][4],

 "exponentialfittingonlaunchdate":
 tableData[i][5],

 "movingwindowaverageoflifetimeinlaunchdateorder":
 tableData[i][6],

 "movingwindowstdevoflifetimeinlaunchdateorder":
 tableData[i][7],

 "log2lifetimeinlaunchdateorder":
 tableData[i][8],

 "movingwindowaverageoflog2lifetimeinlaunchdateorder": tableData[i][9],

 "newfit1":
 tableData[i][10],

 "newfit2":
 tableData[i][11],

 "newfit3":
 tableData[i][12],

 });

}

return chartData;

}

```



```

 })

 AmCharts.addInitHandler(function(chart) {

 // set resizeCategoryWidth

 if (chart.resizeCategoryWidth === undefined)

 return;

 // set base values

 var categoryWidth = chart.resizeCategoryWidth;

 // calculate and set container dimensions

 var containerWidth = chart.resizeCategoryWidth *
chart.dataProvider.length;

 if(chart.rotate) {

 chart.div.style.height = containerWidth +

"px";

 }

 else {

 chart.div.style.width = containerWidth +

"px";

 }
 });

```

```
}, ['serial']);
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```