Methods for extracting data from the Internet

by

Joel Willers

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Sociology

Program of Study Committee:
Shawn Dorius, Major Professor
David Peters
Olga Chyzh

The student author and the program of study committee are solely responsible for the content of this thesis. The Graduate College will ensure this thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2017

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to thank my committee members, Olga Chyzh, David Peters, but especially my major professor, Shawn Dorius, who without his guidance and patience, this would never have been possible.

In addition, I'd also like to thank the department faculty and staff, especially Rachel Burlingame, who prevented me from missing all of my deadlines. I want to also thank my friends and family for their support, especially Leigh Ann Long, and, of course, my wife, Bethany, for her patience and support.

ABSTRACT

The advent of the Internet has yielded exciting new opportunities for the collection of large amounts of structured and unstructured social scientific data. This thesis describes two such methods for harvesting data from websites and web services: web-scraping and connecting to an application programming interface (API). I describe the development and implementation of tools for each of these methods. In my review of the two related, yet distinct data collection methods, I provide concrete examples of each. To illustrate the first method, 'scraping' data from publicly available data repositories (specifically the Google Books Ngram Corpus), I developed a tool and made it available to the public on a web site. The Google Books Ngram Corpus contains groups of words used in millions of books that were digitized and catalogued. The corpus has been made available for public use, but in current form, accessing the data is tedious, time consuming and error prone. For the second method, utilizing an API from a web service (specifically the Twitter Streaming API), I used a code library and the R programming language to develop a program that connects to the Twitter API to collect public posts known as tweets. I review prior studies that have used these data, after which, I report results from a case study involving references to countries. The relative prestige of nations are compared based on the frequency of mentions in English literature and mentions in tweets.

CHAPTER 1

INTRODUCTION

The Internet is a valuable data repository for the modern world. In addition to being able to share data directly, a researcher can also gather data from the World Wide Web. The advent of the Internet and large-scale storage have made possible exciting new opportunities for social scientists to collect data and develop new measures of public opinion and this includes gathering data with online surveys and sharing data in online repositories. The Internet combines a number of benefits of previous communication technology and adds several more (Bargh and McKenna 2004). This project explores the utility of two methods of gathering data from the Internet: web scraping and application programming interface (API) technologies.

Since there are many Internet tools that a researcher might employ, I will briefly discuss other types of web-based tools available to researchers. I will also review other studies that use either web scraping or APIs. Finally, I will present a case study utilizing both methods. I will collect data from the Google Books Ngram Corpus with a custom web scraping tool and from Twitter using a custom program to access the Twitter Streaming API. I will present previous significant research in order to validate the use of these two sources of data. The case study will use mentions of a country as an indicator of national prestige and compare frequencies as a relative measure of total prestige for a nation. This relative measure of national prestige will then be compared to conventional measures of national prestige, such as the Nation Brand Index (NBI).

In recent years, the library of tools a researcher can harness has rapidly expanded. Some tools allow for better collaboration, like data repositories that can be shared with colleagues around the world (Boulos et al. 2006). Others allow for better display of information, such as creating word clouds or word maps (Kavanaugh et al. 2012). Some of the most powerful tools and methods for scientific research were developed to aid in Internet-enabled data collection. These include online surveys, processing large data sets (Big Data), web scraping and APIs (Cook et al. 2000; Schmidt 1997; Glez-Peña et al. 2014; Aitamurto and Lewis 2013). The data collection methods of web scraping and utilizing an API will be the main focus of this paper.

This project uses web scraping to gather data from a site that displays the frequencies of words or groups of words in published books (Michel et al. 2011). Public access to the Google Ngram Viewer can be found at https://books.google.com/ngrams, which displays data from the Google Books Ngram Corpus, a database of words extracted from millions of books that were digitized and catalogued. While this is a specific example, almost any website can be accessed and 'scraped' for the rich data that might be of interest (Yang et al. 2010; Dewan et al. 2007; Alonso-Rorís et al. 2014; Glez-Peña et al. 2014).

While Google Ngram Viewer produces nice graphs for users, having the raw data is often more useful. Google makes available the complete data set for download, but it is rather large and unwieldy (one of the problems of Big Data research). This tool allows researchers to develop custom search criteria, which are then sent as a request to Google servers. The program parses the response to provide researchers with a raw data in a structured, comma separated data file (CSV). Since the data is restricted to only the search criteria, the table size is much more manageable than what Google made available to

researchers. Dozens of data files containing the full corpus had to be downloaded and processed, which is both slow and costly.

The term 'web scraping' refers to the process of requesting a web site from a server and parsing the information returned for the specific data you seek. When a request is sent to a web server, the reply is written in HTML, a coding language used by web browsers. Finding the section containing the desired data is a manual process. Once a program returns the raw HTML, the coded response must be visually searched for the required data. Once the appropriate data have been identified within the page of code, researchers must write the code so that it can reliably access the correct section of the web site.

With Google Ngram Viewer, different search requests returned different layouts, so the code had to be responsive enough to handle these layouts. Sometimes this just comes with use by outside individuals. Everyone uses a product differently, so the code needs to be robust enough to handle users with very different technical abilities and research objectives. While it would be simpler to develop a program and data collection tool that is highly customized to a specific research question, it would also mean that each researcher wishing to access data from the Google Books corpus would need to develop their own custom program. This is both time consuming, inefficient, and error prone, since each custom program is at risk of programming (processing) errors (Groves 2010). For comparative purposes, I follow the strategy of custom program development to collect data from the Twitter Streaming API.

One of the design goals of the custom web scraping tool is to make it available to the larger scientific community and to the public at large. The tool is online for anyone to use at http://shawndorius.com/www1/repository/ngrams2.php. Because of the open audience, I

had to create a 'front end' for users, so they could use the tool without having to download it or directly insert their searches into code. This, of course, creates several obstacles, many of which I address below, and other problems, such as browser compatibility, which I do not explore in depth in this research.

Another method that has become increasingly popular for collecting data through large online repositories leverages the API of a web service. An API is a data interface offered by an online service to more efficiently transfer data. Twitter offers two types of APIs for their publically available user data, known as 'tweets'. In the first, the search API scans tweets that have already been posted. Tweets are ranked based on number of views, how many users 'favorited' the tweet, and how many times it is retweeted by unique users. Approximately 40% of the highest ranked tweets are returned through the search API. This is where the second API, streaming API, comes in. The Twitter streaming API provides users with a constant stream of every public tweet being posted in real time. This API delivers nearly 100% of tweets, which is why I chose to use it.

Twitter is used by nearly 300 million active monthly users (different people using it at least once each month). While the analysis of Twitter data can reveal many scientific insights, the data are sufficiently new to the scientific community that there are a number of uncertainties regarding the reliability, validity, and generalizability of such user-generated social media data (sometimes also referred to Big Data). The uncertainty can pose a challenge to research where the objective is to use the Twitter data to generalize to the general public or other known segments of the population (Kaplan and Haenlein 2010). First, with only 140 characters able to be published in a tweet, many users use abbreviations for words to make their message shorter. This makes it difficult to watch for

keywords or do content analysis. Twitter is free to use, but a person must have an Internet capable device with a live connection to the Internet, limiting it to more wealthy users. Finally, users of Twitter know that their tweets are public, so it is possible that some (many) users post tweets that purposely draw attention to themselves and that may not reflect their personal attitudes, beliefs, or values. This makes the population of tweets potentially not representative of the population at large. In the absence of more detailed information on the Twitter user base in any given data collection, researchers are advised to generalize their results only to the population of Twitter users.

The final goal of this paper is to show the usefulness for sociological research of collecting data using these two methods with a case study. The case study involves measuring the relative prestige of a nation-state by comparing the frequency of mentions of the country's name in English-language books using the Google Books Ngrams Corpus and mentions in publicly available tweets.

While it is time consuming to develop original programming to collect these sorts of data, so long as researchers adhere to open-access and open source standards, the effort can be applied to many different kinds of social scientific problems. With the wealth of information that is online, I urge social scientists to become aware of such methods, even if they and their research teams currently lack the technical skills to develop their own data collection protocols. I will show the validity of the data being harvested with this case study, and demonstrate how these tools made the data collection more efficient.

CHAPTER 2

BACKGROUND

Innovations in technology make scientific research more efficient and the Internet is no exception. The Internet is a great place to find entertainment or news, similar to what television or radio have previously delivered, but it also revolutionizes communication, as the telephone did (Bargh and McKenna 2003). Of course, the Internet didn't replace these technologies, but it did allow for new efficiencies in many fields including, collaboration and research. Collaboration was made more efficient with the increased ability to communicate and share documents. Research could be shared over the Internet, but the content of the web itself could now be the topic of research as well. The advent and expansion of the Internet has been attended by specialized tools that allow for increased efficiency of both collection and processing of data within specific fields of research. The goal of this paper is to demonstrate how the Internet and specialized tools designed to harness the power of the Internet will increase efficiency for researchers in the social sciences. Often, social scientists need to approach large data sets and 'big data' as a computer scientist would (DiMaggio 2015).

One of the most powerful impacts of the Internet is how it has affected collaboration. Wikis are a one such example of a web-based tool that allows experts to relay information to others in an open-access forum. Wikis are typically editable by anyone with access, so users can build their own guidebooks or resources to be shared with others. Medical researchers used this technology to not only track the impact of the avian flu, but

also to spread information about how to prepare for this pandemic (Boulos et al. 2006). The medical community also used similar tools to update the guidelines for the sustained use of sedatives and painkillers for critically ill adults. Researchers used specialized software to collect over 19,000 references from search engines and enabling contributors to review and offer feedback to ultimately create an updated guide that benefitted everyone (Barr et al. 2013). Government agencies have used Internet technology for years to collect information on the public they serve. The concept of e-government has been studied for many years and has been shown to harness the technology to fundamentally change how constituents interact with government agencies (Tat-Kei Ho 2002).

The Internet of Things (IoT) is a term that applies to devices that can be tracked or monitored or controlled via the Internet. These devices might be vending machines that monitor inventory levels or sensors that measure wind speed. Communication protocols are as diverse as functionality with different approaches for security, storage, and privacy considerations (Atzori et al. 2010). Two-way communication to and from these devices that can have minor processing capabilities allows a certain level of smart technology (Whitmore et al. 2015).

When it comes to social work, turning inanimate objects into responsive tools opens up workers to help even more. Health professionals can monitor weight or prescription levels and care workers can monitor nutritional information in order to elevate the relationship of caregiver and client (Goldkind and Wolf 2014). One of the pioneers in the IoT paradigm is Kevin Ashton who wrote, “The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so.” (Ashton 2009).

In the field of sociology, surveys are a powerful data collection tool. Innovation in web surveys and statistical methods to bolster representativeness help reduce the problem of low response rates which have steadily increased over the years, making surveys less and less representative of the populations they purport to study. Dramatically reduced costs for web surveys have contributed to the explosion of the use of this efficient tool (Cook et al. 2000). One of the first papers written about online surveys discussed their benefits and how to overcome the flaws with online surveys (Schmidt 1997). One of the leading authorities in surveys, Don Dillman, wrote in a 2000 book, Mail and Internet Surveys: The Tailored Design Method, which explored methods for pairing conventional survey methods with the Internet to advance new ways of measuring publics. Suffice it to say that the increased efficiency that these Internet tools provide is of interest to many researchers.

While generic methods and resources are useful, few services can outperform tools designed for a specific use. Even though they may be narrow in scope, the narrowness is vastly overshadowed by their usefulness in a given field. Searching "online tool" on Google Scholar produces over 32,000 results, most of which describe custom tools designed for specific research purposes. Areas of study for these tools include bioinformatics, cartography, viral medicine, and genetics, to name a few. One such tool is a 'web crawler', which follows links on web pages. Web crawlers are an early example of web scraping that has been used, for example, to measure connectivity of students by adding up links to other students' web pages (Adamic et al. 2003).

The data collection capabilities of web scraping can be applied to many different types of research. Pricing and availability data, supplied by retail companies to their customers, such as Amazon.com, can be collected by competitors for price adjustment and marketing purposes. Similar techniques are used by financial markets in order to offer more competitive prices for stocks throughout the market (Dewan et al. 2007). Climatic data is incredibly valuable in the agriculture industry. Several agencies collect and distribute climatic data and web scraping is a useful tool to collect data from these sources (Yang et al. 2009).

Several web scraping tools have been developed by researchers and described in scientific journals. Two such examples involve web tools that were developed to access data from published data sets (Alonso-Rorís et al. 2014; Glez-Peña et al. 2014). The first describes methods for collecting data from many sources, including digital TV streams (Alonso-Rorís et al. 2014). The second uses web scraping to collect data on bioinformatics; specifically scraping data about genetics (Glez-Peña et al. 2014). In both cases APIs were described as the modern method of delivering data. Unfortunately, not every site offers an API, and those that do sometimes do not offer all the data needed by the researcher. Both of these methods, though, allow for large data sets to be made more manageable. Only a small slice of the data set is used, reducing time and money expenditures.

Of course, an API is still a powerful tool for disseminating data. Similar to creating a web scraping tool, programming knowledge is needed, however the tools themselves are often designed for researchers who often have limited programming knowledge. Web scraping tools can also be made into functional APIs for use by researchers. Since APIs

create interfaces for programmers to more easily access published data, they are important tools in collecting and processing data.

Any sort of data repository can have an API created for it. In the medical field, surgical gestures were parsed using raw motion data and an API was created to serve the data for use by researchers. This data can then be used to assess the surgical skill of doctors or even create robots to do surgical procedures (Lin et al. 2006). Another example is the creation of an API to deliver soil sample data to researchers. Using the Google Earth API and collected soil data, an API was created to help researchers view and collect soil survey data in a quick and easy way (Beaudette and O'Geen 2009). News organizations also often offer researchers an API. Data sharing for journalism has been an innovation that works as a research and development lab along with a tool that can identify new markets and business models (Malhotra et al. 2012).

For social researchers, collecting qualitative data from social media networks like Facebook or Twitter has powerful research potential. Trying to approach this using a manual method of copying data and pasting it into a database is tedious and can lead to data processing error (Groves 2010). Utilizing an API, if well made, can offer reliability and efficiency. Using APIs, researchers studied the digital footprints of users of Facebook, LinkedIn, Twitter and YouTube. Content analysis of public data have been shown to be able predict accounts by the same user with over 90% accuracy, which can cause privacy concerns (Malhotra et al. 2012).

Of particular interest is the Twitter API, one of the most popular microblogging sites on the Internet with large, active user bases in virtually every corner of the world. Twitter gives researchers the ability to do content analysis and to pair this with spatial analysis due

to the availability of locative data. This allows researchers to study not just *what* users are talking about but *where* these users are when they are sharing their opinions. One of the most well-known examples of the use of Twitter in scientific research involves its deployment in the tracking of emerging diseases. Locative information is voluntary on Twitter, so the percentage of tweets containing this information is small, but because the user base of Twitter is so large, researchers can still study patterns of reports of disease through the Twitter API (Burton et al. 2012). Twitter location data has also inspired innovation in cartography and the display of locative information (Field and O'Brien 2010).

The limitations of Twitter and the Twitter API are discussed in detail by Oussalah et al. (2013). This research reviews the ins and outs of creating a database for tweets based on the Twitter Streaming API. Their discussion describes challenges that have not often confronted social researchers, including the sheer size of the database (Big Data), the challenges of language algorithms, and sampling rates. I address some of these difficulties in the methods section of this paper.

Utilizing APIs is an efficient method for collecting data from the online data repositories. This paper gives primary attention to the procedures involved in online data collection, but the validity of the data itself is important to address. As an example of the impact Twitter has had on the scientific community, a search of Google Scholar for the term "Twitter" returned nearly seven million hits (as of April 2017). It is difficult to trace how many times a paper has actually been cited, but Google Scholar lists one of the earliest introductions to Twitter as being cited over 2,000 times (Java et al. 2007).

There is great potential to harness social media in profit-making activities, so marketing divisions attempt to identify methods for harnessing advertising potential

(Kaplan and Haenlein 2010). Researchers hypothesize that consumers use social media as a tool to identify, which leads to brand identification and targeted marketing (Schau and Gilly 2003). Word of mouth advertising is a powerful marketing tool and Twitter represents a modern method to quickly spread personal recommendations or criticisms of products (Jansen et al. 2009). Twitter has also been used to predict the volatility of the stock market by doing content analysis of tweets to determine the average mood of the populace. Changes in mood were found to be reflected in the Dow Jones Industrial Average a few days later (Bollen et al. 2011).

Twitter can be useful for other fields of research beyond business and marketing applications. Researchers have found that Twitter can be used to track information about diseases. In 2011, researchers used tweets to track public concern about the spread of H1N1, better known as swine flu. In addition, Twitter can be used to actually track the spread of diseases such as the swine flu (Signorini et al. 2011). Twitter has been used to identify misinformation or misuse of antibiotics by the public and to help spread valuable corrective information (Scanfeld et al. 2010).

Social scientists have also utilized Twitter as a method to measure various social phenomena. One such example is Golder and Macy (2011) studying the diurnal and seasonal moods of Twitter users. They found that patterns were similar among different nations and cultures and that it confirmed expected circadian rhythms (Golder and Macy 2011). Another example applies Goffman's theories of identity presentation and audience (among others) to social media (Marwick 2011).

The other data resource besides Twitter used in this paper is Google's ngram project found at https://books.google.com/ngrams. Uses of the data from this resource became

widespread with the introduction of the Google Books Corpus in a 2011 paper published in *Science* (Michel et al. 2011). As of April 2017 the paper had been cited over 1000 times. In 2012, an extension to the corpus was announced with syntactic annotations added (Lin et al. 2012).

Evans and Foster (2011) discuss the idea of data about data (which they call 'metaknowledge') using the ngram corpus as an example. The researchers discuss how a paper offers results about a topic, but how many papers cover a topic is also of interest. This can also be applied to other parts of society, like measuring the importance of an idea by tracking the number of times it is mentioned in literature, as the ngrams corpus does (Evans and Foster 2011). Some researchers have compared the validity of Google's corpus for all uses, such as word recognition by people today (Brysbaert et al. 2011).

Researchers who used the database mostly discuss language itself, such as the 2012 paper about new words in a language over time (Petersen et al. 2012). Others use mentions of a word or phrase in literature as a unit of measure to calculate the importance of a concept to society. One example is to measure how often literature printed during a year mentions the past, present or future by charting mentions of a year throughout the entire corpus (Preis et al. 2012). Happiness can also be tracked using a corpus of words by comparing how often positive words are used compared to negative words throughout the years of the corpus (Dodds et al. 2011).

The main purpose of this paper is to demonstrate the use of two methods for collecting information. These two methods are used on two data sources as examples which, in turn, are used in a case study to investigate how these data sources can be used to measure relative prestige among countries.

CHAPTER 3

TOOL DEVELOPMENT

The development of the tool for scraping the Google Books Ngram Corpus was quite a bit different from creating the code for accessing Twitter's Streaming API. The major difference in the design restrictions of the two tools was the audience. In order to access Twitter's Streaming API, users must create a personal Twitter account, after which they can write code to access the data for their own research purposes. The goal of the web scraping (CorpusExtract) tool was to create a public interface, so the approach was completely different.

Web Scraping and Google Books Ngram Corpus

When approaching the creation of this custom tool, there were several design goals to keep in mind. First, the CorpusExtract tool would be used by non-programmers. Second, it needed to be open to the public. Third, the tool would need to be easy to use and intuitive. Finally, it had to return data in a usable way.

Because of these constraints, I wanted to build a tool that could be hosted online for people to use, rather than developing a downloadable executable file, which would require users to install and run the program on a local computer. I used the PHP programming language due to its popularity among programmers and ease of publishing and deployment. PHP works alongside HTML, the programming language of the Web. The display is simple: a form that collects the same options as the Google Ngrams Viewer. A

user enters the options and the tool displays the results as a data table (instead of a graph) which the user can then easily download or import into any standard data analytics package (e.g. Stata, SPSS, SAS, R).

In order to get a table of data from the submitted form, the CorpusExtract tool performs several functions. First, it turns the search parameters into code to be sent to the Google Ngrams Viewer page. Next, the tool requests the data from the page which is given in HTML. The tool breaks down the HTML to find the needed data. Finally, it parses the data and creates a table to display the results.

Writing the Code

Behind the scenes, the program is much more complicated. The Google Ngram Viewer sends data through the web address or URL, which means if a user wants to copy a graph, he or she can simply copy the URL of the search. This allows a programmer to test the form and see how the URL changes to reflect the different search parameters. Every selection will change the URL in a particular way, so for making the CorpusExtract tool, the changes must simply be emulated.

In order to pass search parameters to the Google Ngram Viewer, search terms entered in the form needed to be translated into code that could be passed into the URL of the Google Ngram Viewer page. To perform this operation, I wrote a function that did all of the translation from input to URL encoding. It removes all extra white space, turns “end of line” into a comma, and translates the request into URL-ready encoding. It takes the text search as an input and returns an encoded search string that the Google Ngram Viewer can understand.

Every field is passed through this translation function, not just the search terms. The range limit of publication year is two fields, the start year and the end year. The start year defaults to the earliest publication date of 1500 CE if left empty or if an earlier year is chosen. The end year defaults to the latest publication date in the corpus, which is 2008 CE if the field is left empty or if a later year is chosen. Users can also choose a corpus language (see Appendix for list). Smoothing can also be chosen, which averages responses from surrounding years so that extreme years are 'smoothed' out. Finally, a user can select whether or not Google should consider case sensitivity in the results ("test", "Test", and "TEST"). All of these fields are translated and combined to create a URL to be sent to the Google Ngram Viewer.

Another function processes the request; it sends this URL-ready call to the Google Ngram Viewer website and parses the response, a process known as 'web scraping'. Within the HTML reply from Google, there is a table of numbers showing the frequencies of the search terms within the search range. The layout of this data table changes with different types of searches, so the contents of the search itself must be checked so that different parsing procedures can be performed on the HTML response. The frequencies returned in the data table give the actual number of mentions for each search term, so the total number of books for a year must also be displayed in order to give relative frequencies, similar to what is given on the Google Ngram Viewer graphs.

The entire processing function uses three variables. The first is the URL built from the translation function that will be sent to the Google Ngram Viewer site. The other two are the starting and ending range of the years, used to create the display table. The URL is executed and the HTML response is searched for the data table. Once the data table is

found, it is broken down into an array to be returned by the processing function. White space and noise are removed from the response and the first row is used as the header row.

When any HTML-based web form processes, there are two ways the information can be sent to the program; either passed directly from the server or passed through the URL. In the first way, known as POST, the server sends data directly to the page for processing. This adds a layer of security for forms that transfer sensitive information, because the information is not easily viewable. There are also no limits to the length of the data. The other method of passing information from a form, is called GET. This method passes the form data through the URL. For instance, if a form at www.example.com wants to pass a first name and last name, the URL might be www.example.com?first=John&last=Doe. This passes the string "John" to the associated variable "first", and similarly the string "Doe" to the associated variable "last". This URL can then be bookmarked or copied to a colleague for reference. This can be a security issue though, as the data would be cached and stored in places like the browser history.

Since search parameters on the CorpusExtract tool are not sensitive security concerns, I opted to use the GET method. This allows users to easily share search parameters with each other. The Google Ngram Viewer page also uses the GET format, which made the CorpusExtract tool much easier to develop as the form elements were all visible in the URL.

In order to make repeated searches easier for the user, the CorpusExtract tool enters the search parameters from the URL into the form. After the parameters are processed and the array is returned, the parameters are taken from the URL format and returned to the form ready to be run again. Not only does this make repeat searches easier,

it also allows a user to send a link to the URL of the tool which includes search parameters, in order to prepopulate the form.

There were many changes required to convert a search request into a coded URL request and vice versa. One example of the difference between entry in the form and the URL-ready string is that a plus (+) within a search means to tie two items together. For instance, “Albert Einstein” searches for the 2-gram “Albert Einstein”, while “Albert+Einstein” searches for “Albert” and “Einstein” separately, and then combines the frequencies of the two. However, in the URL, using a plus is the same as a space. So “Albert Einstein” turns into “Albert+Einstein” after translated into a URL-ready format. Commas and pluses are special characters, so they become HTML encoded. Comma becomes %2C and plus becomes %2B. So the translated version of the search “Albert+Einstein,Sherlock Holmes” would be “Albert%2BEinstein%2CSherlock+Holmes” when made URL-ready.

Parsing the response from the Google servers was a bit of a challenge, one that is often encountered when web scraping. As noted earlier, different types of searches return different types of responses which have to be parsed slightly differently. The response is built in a type of array format, and retrieving the desired information took a bit of trial and error. When creating a parser, it seems that there are always outside cases that need to be handled. After developing the initial program, I tested it using different searches, which were then compared to results on the official site. When differences were found, I would look at the source page and augment the parser code.

Another obstacle to overcome was that Google only allows twelve terms to be searched at a time. I designed the CorpusExtract tool to be able to accept any number of search terms. The tool breaks the list into chunks of twelve in order to process them with

the Google server, but displays them all together in one data table. This is another advantage of using the tool over the Google Ngram Viewer, as the data table is much more usable.

There were other minor obstacles that also had to be overcome. One was to ensure that form settings were within the limits of Google's range. For instance, "smoothing" has to be between 0 and 50. If you enter something outside of the range, the tool sets it at 0 or 50, whichever is closer. Smoothing was a small challenge in itself. Since the totals for the year are set, the smoothing can cause some weird results, so it is done within a function of the CorpusExtract tool, instead of being done by Google. Another example of a minor obstacle was discovered with the 2009 corpora. The files with the total number of word counts for each year were in a vastly different format than the 2012 corpora, so the 2009 corpora were removed from the selection. This should not impact the usability of the tool because the 2012 corpora contains a great deal more books, and thus should be more accurate for word frequencies than the 2009 corpora.

The Twitter Streaming API

The design goals of the Twitter collection tool were must less stringent than for the CorpusExtract tool. Twitter offers two main choices to access their data. You can attach to the Streaming API to get tweets in real time, or you can use the Search API, but the results would be filtered. Since I wanted more accuracy, I chose the Streaming API. Either way, an application needed to be registered with Twitter in order to obtain credentials for accessing the Twitter data. I made the tool private, rather than public to avoid having to create an interface for users to sign in with Twitter.

According to quick comparisons, it was shown that the Search API filtered some of the results from the stream, while the Streaming API delivered nearly all public tweets. According to the Twitter API documentation, “Before getting involved, it’s important to know that the Search API is focused on relevance and not completeness. This means that some tweets and users may be missing from search results. If you want to match for completeness you should consider using a Streaming API instead.” For the sake of thoroughness, I elected to focus on the Streaming API.

Several code libraries are available to access the Streaming API, so a programmer does not have to create the entire program. I tested two such libraries, one in PHP and another in R. Both had benefits and restrictions, but the ‘streamR’ package proved to be easier to use and collected the information in a useful way. Also, R is a statistical package, which allowed me to both collect, tidy, and analyze the data within a single programming language and software environment.

Other design decisions were made before the final code was written. A text file will be used for search terms so that changes can be documented, if needed. I also wanted to consider some file size limitations so that processing and copying would not be too problematic. Each tweet collected is around 6.5kb of data, so in order to have a manageable file size of less than 20Mb for each file, I will have to limit the number to around 3000 tweets per file. Because this is custom code and not a tool designed to be used by the public, any other design decisions would be based on a specific case.

The basic overview of functionality is pretty typical for a program of this type. First it loads up the libraries it will use for connecting to the Twitter Streaming API. Next, it must send credentials to open the Twitter Stream (called the Firehose). These credentials are

given when you register an application with Twitter. You can save this authorization to a file so they will not need to be checked every time the program is run. Next, the storage locations and base filenames are defined. The next step is to define how many files will be created and how long the Stream will feed into each file. Finally, a loop is created that iterates once per file that was set in the previous step with each iteration lasting for the defined time.

Writing the Code

The two libraries used are 'ROAuth' and 'streamR'. The ROAuth library allows the user to authenticate the transaction with the Twitter API. Before a user can access the Streaming API, she must first register an application with the Twitter developer site. This is a free registration that supplies credentials to be used within the code. When the code is run for the first time, the application requests a code supplied by Twitter. Twitter will prompt the user to accept the terms of conditions and allow the application to have access to the user's account. If accepted, a code is supplied that must be entered into the application. Once this is done, the authentication can be saved for future use so the user can avoid repeating the acceptance process each time the application is used. After all of the authentication is done, the streamR library can be used to access the Twitter Streaming API.

The streamR library offers two main functions: filterStream and parseTweets. The filterStream function opens the Twitter Streaming API 'Firehose', and collects all tweets meeting certain criteria and saves them to wherever the user defines. The user can tell the function where to save the information, either to a file or to the console. The function also

needs to know what string (or a list of terms) to watch for within the Firehose. The user can also give a user ID (or a list of IDs) to monitor. Latitude and longitude can also be filtered, although a small percentage of users were found to supply locative information. Language can also be filtered, so if you want to watch for words that are ambiguous (like 'favor' which is both an English and Spanish word), you can specify which language you prefer.

Beyond filters, the filterStream function can collect a certain number of tweets or it can collect all tweets until a certain amount of time has passed. Only one or the other limits can be set, and once the limit is reached, the stream closes. While the function has the stream open and monitoring for tweets, the program does nothing else. Once the stream is closed, the program continues on to the next line of code. In order to have smaller file sizes, the function must end after a certain period of time or a certain number of tweets and then open again in order to save tweets into the next file.

The other function from the streamR library that we use is 'parseTweets'. The filterStream function saves tweets as a single string of data, while the parseTweets function takes the string and turns it into a data frame, which R uses as a data type that can easily be saved to a CSV file (or similar). I use this data frame to more easily manipulate the data and search for interesting trends using statistical models. This will be explored more in the case study.

Before collecting tweets, the user must decide where to store data and what the file structure will look like. The user first sets the folder that will store all of the data. Of course, the program must have permissions to write to that location. Next, the base filename needs to be set. Since we have a concern for file size, we set a loop to collect a certain amount of

data and then close the file and open the next. The base filename will be amended with the current loop count. So if the user sets the base filename to 'Twitter data ', then this will save the first set of data to 'Twitter data 1.txt', then 'Twitter data 2.txt' and so on. This data will just be strings of data since they haven't been parsed yet, which conserves storage space.

A user can decide to limit the amount of tweets to process at a time by a time limit or by setting a specific number of tweets. I chose to limit based on a time of six minutes, even though that can cause a variation in file sizes, because the stream is opened and closed every time the function is called. Twitter limits the number of times an application can terminate and reconnect to its API, so in order to ensure a longer connection, I set a time limit. If too many tweets came in a short amount of time, like for a breaking news story, the connection could reset too often, resulting in lost data.

The data collected from the Twitter Streaming API has 42 fields. These include the user ID, the content of the tweet, location information, time information, language settings, and several more attributes of the tweet (see Appendix for the full list). The content filter watches for the content in any field, not just the tweet itself. For instance, if a user was monitoring the Firehose for "zebra" and a user's description was "A true zebra lover," every tweet from that user would be flagged, even if the tweet did not include the word "zebra". Of course, this can be processed once the tweets are parsed; however, it's important to note that if a user wanted 1000 tweets that include the word "zebra", he or she should collect a few extra for this reason.

One of the biggest obstacles for this tool is the inherent difficulty of searching for phrases in a multi-lingual application. This program monitors the Firehose for the exact search phrase. Searching for "hello" and setting the language to "esp" (Spanish) will not

capture “hola”. The program simply monitors strings of letters with the specific settings set by the user. More obstacles for search metrics will be discussed in the case study.

CHAPTER 4

CASE STUDY

The prestige of a country is a difficult concept to quantify. It is similar to measuring the development of a country, however, there are some major differences. The concept of prestige combines many aspects, such as power, notoriety, resilience and visibility. It is also an aggregate measure of smaller groups, like cities, companies, or citizens that reside within the nation. With this in mind, we can measure each of the pieces that contribute to national prestige (causes), or we can measure the indicators of prestige, such as being the topic of discussion. The goal of this case study is to conduct an exploratory and descriptive analysis of national prestige using big data. In order to gain insights into the historical evolution of national prestige, I will use mentions in literature by accessing the Google Books Ngrams Corpus with the CorpusExtract tool. I will use the Twitter API to monitor the Twitter feed for mentions of countries to measure current prestige.

Case Study Background

Before discussing the measurement of prestige for a country, it becomes important to first delve into the generic concept of prestige. An entity that possesses more prestige, or an individual that is a member of a prestigious group, is regarded to have earned more respect in society (Henrich and Gil-White 2001). Measuring prestige is most often done by asking participants of a survey, but Robert Merton used citations as an indicator of prestige

(Merton 1968). Though citations are not a one to one direct measure of prestige, they show increased visibility, which can indicate a level of respect or notoriety.

Since occupations are highly regarded in industrialized societies, occupational prestige has been studied for decades. The North-Hatt prestige survey of 1947 was one of the earliest attempts to measure occupational prestige and the concept was added to the General Social Survey, and later updated as the early scales began to age (Nakao and Treas 1992).

In 1977, Donald Treiman wrote a thorough analysis of occupational prestige in his book, *Occupational Prestige in Comparative Perspective*. Treiman supports the study of occupational prestige arguing that the social standing of a person in any society that values occupational stratification will be heavily influenced by the prestige of his or her occupation. Treiman also argues that occupational prestige is comparable across cultures in different areas or even times and developed a Standard International Occupational Prestige Scale. Another benefit of using occupational prestige is the relative ease of measurement. Treiman states in his introduction, “Every adult member of society ordinarily is able to locate occupations on a hierarchy of prestige.” (p. 1)

Although it seems easy to just ask respondents to rank occupations, that can still be rather complicated due to the large number of occupations to rank. The methodology used in the aforementioned 1989 General Social Survey was to take the GSS sample of 1500 respondents and break them into 12 subsamples. Of the 12 subsamples, 10 were chosen to rate 110 occupational titles (40 of the titles were common among all the subgroups while the other 70 were unique). This measured a total of 740 different job titles. The titles were then given a score of 0 to 100 based on the responses (Nakao and Treas 1990).

Occupational prestige is a useful concept when measuring the social position of a person. Respondents might be unwilling to reveal wealth or earnings information, which is often less stable than job title (Connelly et al. 2016). Occupational prestige is also a useful predictor of social mobility and indicator of social inequality (Hauser and Warren 1997).

Measuring the prestige of a social concept that is less common can be more difficult than measuring job titles. Different concepts often need creative methods to be measured in terms of their prestige. A movie, for instance, might use box office earnings or the cumulative rating from critics. Online videos might be rated based on number of views. A phenomenon that Robert Merton called the "Matthew Effect" might influence the social significance of cultural artifacts (Merton 1968). Merton theorized that the prestige of a researcher or paper was reflected in how many citations it accrued. As a researcher was cited more often, he or she would gain prestige, which, in turn, would make researchers want to cite him or her more. The Matthew Effect can be seen with movies and online views as well. When a movie makes more money, more people want to see it, increasing its earnings. When a video gains more and more views, more people want to see it. Since these reflections of prestige never expire, the effects are cumulative.

Disadvantage can also have a cumulative effect. Cumulative disadvantage has been researched with regards to health impact (Angela 1996; Seabrook and Avison 2012; Crystal et al. 2016) and crime (Bernburg and Krohn 2003; Lopez et al. 2012; Kutateladze et al. 2014) among other topics. Both positive and negative cumulative effects are exacerbated by social media. Because of higher social connectivity, positive or negative actions and reactions are faster and often more impactful (Goldman 2015).

This paper approaches the topic of the prestige of a country, which is similar to nation branding. Simon Anholt has cautioned against the use of the term 'nation branding' since it often gives an incorrect connotation. He begins the introduction of his 2010 book *Places: Identity, Image and Reputation*, "Let me be clear: there is no such thing as 'nation branding'." (Anholt 2010) In this paper, I refer to prestige rather than branding, as I feel it is a more precise, and sociological, description of what I measure. Anholt goes on to say that because of the term 'branding', some governments believe that a simple marketing campaign can improve their brand, but it is not as simple as that, which is why he offers the alternate term 'competitive identity'. One of the most cited sources for nation branding is Keith Dinnie's *Nation Branding: Concepts, Issues, Practice* (Dinnie 2016). He writes, "In this book the nation brand is defined as *the unique, multidimensional blend of elements that provide the nation with culturally grounded differentiation and relevance for all of its target audiences*." This definition includes all aspects of what might influence the prestige of a country, including marketing attempts.

Where prestige and branding might differ is in negative press. Prestige encompasses power and respect, so a country like North Korea might score high for prestige due to military strength, but low on branding, due to human rights infractions. Differentiating between North and South Korea is very important to South Korea, considering such negative publicity with North Korea. Since the brand of 'Korea' was so damaged, Korean products were undervalued for years, but now "Korea Discount' is becoming "Korea Premium' through years of image control by South Korea (Dinnie 2016). For this analysis, I use citations to measure the visibility of a country, which is an indicator of prestige.

The economy of a country can be greatly influenced by its brand because corporations can use nation branding to help market products. For instance, a perfume from Mexico might use a French-sounding name. On the other hand, nation branding might not have as large of an impact as some might think, since a majority of Chinese citizens think negatively of Japan, but they still purchase Japanese cameras more than any other country of origin (Fan 2006).

The Nation Brand Index (NBI) ranks countries based on what Simon Anholt calls the Nation Brand Hexagon, which is six dimensions that comprise the brand of a nation (Anholt 2005). These dimensions are: tourism, exports, governance, investment and immigration, culture and heritage, and people. Researchers can rank countries based on surveys of consumer perception of competence in each of these six fields.

Case Study Methods

In both historical and current prestige measurements, mentions of countries will be compared to national statistics, such as GDP, population, and per capita GDP. Mentions will be limited to the English language, which suggests that what is really being measured is the prestige of countries within the Anglosphere. The methods I outline below can be used to measure prestige in other languages, but that is beyond the scope of this research.

Even within a single language, a country could have many names. A great example is the United Kingdom. The country might be referred to as the UK, England, Great Britain, or rarely by the full name: "The United Kingdom of Great Britain and Northern Ireland". For comparisons, I tried to find the most often used and simple variations of a country name to ensure something close to an 'apples-to-apples' comparison. The majority of country names

used in the research reported below were extracted from the Food and Agriculture Organization. I used country short names, rather than official country names. For example, the short name for Iran is Iran, while its official name is the Islamic Republic of Iran.

Several countries were excluded from analysis due to unusually high measurement error. Examples include Chad (a popular first name in English), Georgia (the name of a US state), Jordan (a popular first name in English), and Turkey (a bird). A number of small, mostly island countries that had the word "and" in the name, like "Antigua and Barbuda" or "Trinidad and Tobago" were also excluded. Counting names with 'and' is difficult as there are mentions of one without the other and they are all small nations with few citations in the databases I search, which further magnifies the risks of measurement error. Dominica and Niger were removed because they are the roots of Dominican Republic and Nigeria, respectively, making it hard to distinguish them from one another. Oman and Togo had similar problems, so they were removed. Timor-Leste was removed because the hyphen caused problems with the code. Hong Kong and Yugoslavia were removed because of problems with naming and sovereignty. Even with the removal of these countries, I was able to obtain data on the vast majority of world countries and these countries collectively account for more than 90 percent of the world's people. If a researcher was looking for specific information on some of these nations, validity problems with the mentions might occur, especially with ambiguities. The complete list of country names is in the appendix.

Historical Measures of National Prestige

I used the CorpusExtract tool to retrieve the number of mentions of nations within literature from 1870 to 2000. 1870 was chosen as the start period for this study because it

broadly overlaps with what has been identified as the beginning of the modern world political economy (Wallerstein 1979). These mentions in literature are operationalized as an indicator of historical prestige for each nation-state. I measure the annual total prestige for each county by taking the sum of the individual country-mentions and dividing it by the total mentions for all of the countries. In this way, results for any one country are presented as a percentage of all country mentions in any one year.

These mentions in literature will be compared to other historical national statistics that are used to measure prestige. These statistics come from various data repositories such as the World Development Indicators, the United Nations, and Organization for Economic Co-operation and Development. Since these are secondary data, some comparisons are difficult to make. If data are unavailable for certain countries during certain years, we cannot simply re-run the query.

Univariate Analysis

When comparing mentions in literature, a researcher can compare the raw number of mentions of a nation per year. The problem with such an over-time comparison is that as time passes and new books are published, the mentions will naturally rise with time. A frequency comparison is more appropriate, as it standardizes country-mentions within each year, allowing for time-series comparisons. I could have compared the frequency of mentions of a country to all words, but this could cause problems as some years people are more interested in countries in general. My choice for comparing country mentions is to sum all country mentions by year, and then convert country-specific mentions into shares of total country mentions. In this way, I can compare the 'market share' for countries.

I demonstrate this in Figure 1, where I report a scatterplot of the raw mentions per year (every decade) for all countries and in Figure 2, where I graph the frequencies. In the first plot, there is an obvious trend of more mentions for all countries as time goes on. In 1870, the first year on the graph, France received the most mentions (n = 100,569). In 2000, the last year graphed, France had 969,399 mentions, but the United States had surpassed France with 2,209,634. The second plot is more uniform, as expected. Because of this standardization, I will use the relative frequency as my basis of comparison.

With so many countries in the list, it helps to break them into smaller groups for comparison. I group countries into five ethno-cultural/geographic regions as follows: Northwestern Europe (including European offshoots), Southeastern Europe, Latin America, Asia, and Africa (country lists are included in the appendix). Table 1 includes the descriptive statistics for the mentions in literature for these groups for individual years.

The first region contains 25 countries of Northwestern European ancestry, which includes countries from continental Europe and neo-Europe's such as the United States and Canada. At the end of the 19th century, six countries were separated from the pack, including France, United States, Ireland, Germany, Canada, and Austria (Figure 3).

One obvious omission from the list of leaders is the United Kingdom. In 1870, it was very powerful and influential, yet it was mentioned less than a tenth as much as Ireland. This is likely due to the various ways to reference the country. As mentioned before, the United Kingdom can be referred to by many names (e.g. England, Britain, Great Britain) and including more of those names will increase mentions. Figure 9 is a graph of the usage of some of the names of the area/nation that are often used interchangeably, even though most of them actually refer to different land masses. It turns out that literature in English

Table 1. Descriptive statistics for mentions in literature by region.

Group/Year	Median	Mean	Max	SD
All countries, All years	8,770	41,310	2,210,000	117,500
NW Europe, All years	37,140	116,500	2,210,000	241,100
NW Europe, 1870-1900	6,307	22,390	262,500	44,930
NW Europe, 1910-1950	14,490	43,700	464,900	86,030
NW Europe, 1951-1980	36,110	91,320	1,052,000	177,700
NW Europe, 1981-2000	77,020	189,600	2,210,000	333,700
SE Europe, All years	9,936	39,610	524,800	68,940
SE Europe, 1870-1900	773	9,091	119,700	19,290
SE Europe, 1910-1950	4,061	17,340	140,900	31,130
SE Europe, 1951-1980	8,320	33,950	327,800	57,660
SE Europe, 1981-2000	19,220	59,580	524,800	89,030
Latin America, All years	11,540	25,650	638,200	50,800
Latin America, 1870-1900	724.5	2,220	37,830	5,021
Latin America, 1910-1950	3,118	7,075	86,380	11,970
Latin America, 1951-1980	9,721	19,440	257,700	33,410
Latin America, 1981-2000	25,130	44,060	638,200	72,470
Asia, All years	5,136	40,060	1,079,000	112,700
Asia, 1870-1900	39	3,280	98,960	11,950
Asia, 1910-1950	319	7,381	181,700	24,080
Asia, 1951-1980	4,050	29,950	644,100	83,720
Asia, 1981-2000	16,990	70,680	1,079,000	158,500
Africa, All years	5,812	15,270	299,600	28,570
Africa, 1870-1900	50	1,412	48,110	5,942
Africa, 1910-1950	254	2,358	52,340	7,153
Africa, 1951-1980	4,572	11,620	156,200	21,040
Africa, 1981-2000	15,130	26,740	299,600	38,660

refers to the country as “England” nearly 50 times as often as the more official name of “United Kingdom” in 1870. By 2000, though, it’s only about 7 times as often. For a more

detailed analysis, a researcher might use a comparison of common names to determine the most efficient to use for each country, but that is beyond the scope of this paper.

The next region contains the other 25 European countries (Figure 4). In 1870, two rise far above the rest, and those are Spain and Italy. The next two are Russia and Greece. By 2000, Russia moves into the top spot, with Italy passing Spain, which comes in third. These three far outpace the rest of the pack in Southeastern Europe and Greece falls to a distant fourth, nearly tied with Poland.

Latin America is the next region (Figure 5). This region of 34 countries is dominated by Mexico. In 1870 Brazil is a distant second place. Through the years until 2000, Brazil trades second, third and fourth with Peru and Cuba, recovering a solid second by 2000. Panama does pop up in the 1940s, but by 2000, drops back down with the rest of the pack.

Asia, the region with the most countries at 58, had six highly cited countries in 1870 (Figure 6). The top country by far was India in 1870. A distant second, yet still far ahead of the rest was Israel. Separated by a similar margin was China, which had substantially more citations than Syria, Palestine and Japan, which round out the top six Asian countries. By 2000, though, Israel has dropped to fourth with India and China practically tied. The prestige of Japan substantially increased after passing China for second place in the 1990s, though it dropped to third by 2000.

The final region, Africa, has the second most countries at 46. In 1870, one country stands far above the rest (Figure 7). Egypt dominates the rest of Africa at the end of the 19th century, but with the passage of time, Egyptian prestige steadily declined such that it was eclipsed by a rising South Africa in the late 1980s. By 2000, four countries rose above the pack: Egypt was first and South Africa second, with Nigeria third and Kenya fourth.

While separating countries into regional groups can help with comparisons, it is important to remember that the regions differ greatly. Figure 8 shows the top countries in each region and shows the huge gap in the mentions for those regions. The United States dominates as the representative from Northwestern Europe. India was a distant second in both 1870 and 2000, but in 1920, Russia temporarily jumped into the second place spot, before falling to fourth behind Mexico in 2000. Egypt started with a strong third place, but by 1910, had dropped to the fifth spot, where it was in 2000.

Bivariate Analysis

For the next stage of analysis, I shift from univariate analysis to bivariate analysis so as to compare country-mentions to determinants of prestige. In Figure 10 I report the relationship between GDP and country shares of mentions in the ngrams database. Similar to the relative mention frequencies discussed above, I standardized GDP by calculating the share of the total GDP of country, by year.

Another interesting visual comparison is the graph of World War I countries' GDP and the graph of their mentions. Figure 11 shows the percent share of the GDP for each of the countries involved in World War I from 1870 to 2000, while Figure 12 shows the share of mentions for those countries. The graphs demonstrate that GDP and mentions in literature are not perfectly correlated. France's slow decline throughout the years shows that a nation's prestige is derived from more than its national wealth. In 1870, the top five countries by GDP are: China, India, United Kingdom, United States and France. The top five by mentions are: France, United States, Ireland, Spain, and India.

If we compare mentions to per capita GDP frequency, where we compare relative wealth per person, there is even less correlation. The top five countries in 1870 for per capita GDP were: Australia, United Kingdom, New Zealand, the Netherlands, and Belgium. A small wealthy population does not seem to translate into prestige. The United Kingdom has a large GDP and a large per capita GDP, but a relatively low number of mentions, though this can be attributed to the naming problems mentioned previously.

Multivariate Analysis and Regression

When approaching linear models with citations in literature, the following concepts were considered for variables: wealth, size, history, and religion. Nested models were created with these concepts in mind, and independent variables were chosen to represent each concept. Table 2 shows these models. Model 1 represents the individual wealth of a citizen with GDP per capita (log transformed). Model 2 represents the size of a country, using a factor score that measures a country's size, broadly defined by GDP, population, and total land area. Model 3 represents the history of the country, including years since independence and a factor score that measures a country's level of democracy and political and civic freedoms. Model 4 measures the effect of a country's religious culture on citations, here measured with dummy variables representing the majority religion in each country. Finally, Model 5 contains all of the variables, allowing for the estimation of the joint effect of all variables. I also consider model fit using the adjusted R-square, where a higher value indicates better fit. The dependent variable in all models is cumulative citations from 1870-2000, log transformed to reduce the skewness.

Table 2. Correlates of a country's cumulative citations in books

	Model 1	Model 2	Model 3	Model 4	Model 5
GDP per capita (logged)	-0.019				0.153
(stnd err)	(-0.10)				(0.84)
Size †		0.978***			1.060***
(stnd err)		(6.39)			(6.26)
National independence prior to 1800			1.056*		0.244
(stnd err)			(2.39)		(0.67)
Democracy factor score ††			0.120		0.118
(stnd err)			(1.08)		(0.90)
Catholic				-0.540	-0.182
(stnd err)				(-0.96)	(-0.42)
Protestant				-0.243	0.019
(stnd err)				(-0.41)	(0.04)
Orthodox				-0.460	-0.525
(stnd err)				(-0.45)	(-0.75)
Muslim				-1.193	-0.619
(stnd err)				(-1.75)	(-1.20)
Intercept	11.19***	11.01***	10.80***	11.49***	9.70***
adj. R-sq	-0.022	0.459	0.135	-0.007	0.546

* p<0.05, ** p<0.01, *** p<0.001

NOTES: † Size measured with a factor score that included total population (0.86), gross domestic product 0.61), and total land area (0.71). All three indicators of size were log transformed prior to estimating the factor score. †† Democracy factor score was derived from several commonly used measures of political freedoms and democracy, including: Freedom house ratings of democracy and freedom, and the Polity IV measures of institutional democracy and state democracy. Dependent variable is cumulative (total) citations in the ngrams corpus from 1870-2000. All independent variables are averages of each variable over all occasions of measurement from 1980-2009. Standard errors in parenthesis. N=48.

The first model has only one independent variable, GDP per capita (log transformed). Individual wealth of the citizens of a country has no impact on how many citations a country has in literature. The coefficient (beta=-0.019) for GDP per capita was close to zero and not statistically significant.

The second model measures the effect of country size on cumulative citations, reasoning that size of country enhances its visibility, prompting more citations. Population, GDP, and land area were each strongly correlated with the dependent variable, but taken together were highly collinear. In order to minimize collinearity, I combine them into a single factor scale that measures size of country. Regressing cumulative citations on size of country produced a relatively high adjusted R-square,making the size of a country a good indicator of how visible it is, thus giving more citations in literature. The coefficient (beta=0.978) for size was statistically significant with $p<0.001$.

The third model has two independent variables, how long it has been independent (age), and a variable reflecting the freedom of the citizens. The age of the country variable coefficient (beta=1.056) was statistically significant with $p<0.05$ and positive, so the longer a country has been independent, the more citations it typically has. The democracy factor score (beta=0.120), however, does not seem to have statistical significance on citations in literature. This model has a low adjusted R-square, too, so the model does not seem to be a good fit.

The fourth model measures the impact of religion on citations. A country is marked with a religion if a majority of the citizens practice that religion. The religions tracked are Catholic, Protestant, Orthodox (more of a type of religion than a specific one), and Muslim. None were statistically significant and the model had a negative adjusted R-square, showing a very poor fit.

The fifth model includes all variables and is the fully specified model. It yielded the highest adjusted R-square, reflecting the best fit. The only significant variable was size (beta=1.060, $p<0.001$), but a larger adjusted R-square shows that the other variables do

help with the fit. In order to be parsimonious, however, it seems that using just size is the best way to predict citations, with GDP (log transformed) to be the single best indicator of all.

Table 3 replicates model results from Table 2, but instead reports standardized coefficients (betas) to facilitate comparison of effect sizes across variables of different scales. Size has the largest impact (beta=0.69), but the age model does have some significance (beta=0.34). Due to the collinearity of age and size (larger countries with large populations and economies tend to have longevity), the significance of age is lost in Model 5. With standardization, though, the size model (Model 2) now has the highest adjusted R-square.

Table 3. Correlates of a country's cumulative citations in books (standardized)

	Model 1	Model 2	Model 3	Model 4	Model 5
GDP per capita	-0.01				0.12
Size		0.69***			0.74***
National independence prior to 1800			0.34*		0.08
Democracy factor score			0.15		0.15
Catholic				-0.21	-0.07
Protestant				-0.09	0.01
Orthodox				-0.07	-0.08
Muslim				-0.33	-0.17
adj. R-sq	-0.022	0.563	0.135	-0.007	0.546

* p<0.05, ** p<0.01, *** p<0.001

NOTES: Dependent variable is cumulative (total) citations in the ngrams corpus from 1870-2000. All independent variables are averages of each variable over all occasions of measurement from 1980-2009. Standard errors in parenthesis. N=48.

In order to reduce measurement error, a researcher could use a list of names for a country and then add them together, but that might require some content analysis in order

to differentiate the United States (or US) from the pronoun 'us', especially if used emphatically. Also, this is measuring books in the English corpus, so a comparison could be done using other languages. This research focuses solely on the Anglosphere.

Contemporary Measures of National Prestige

I used the Twitter Streaming API to monitor the Twitter stream for ten days in September 2015, recording mentions of a list of countries (see Table 4). These mentions in tweets are operationalized as a contemporary indicator of prestige for each nation-state. Just like with the Ngrams, each mention is considered one unit of prestige taken as an aggregate measure of the individual aspects of a society, such as the people or institutions, which is then applied to the entire country. If a single tweet mentions a country multiple times, it is only counted once, although mentioning two separate countries counts one for each nation. I measure the total prestige for a country by taking the total sum of the mentions over those ten days and compare that to the total mentions for all of the countries to calculate relative prestige.

Table 4. Descriptive statistics for tweets by group.

Group/Year	Min	Median	Mean	Max	SD
All countries	5	1,253	5,560	75,810	11,400
NW Europe	215	2,764	6,766	38,450	10,030
SE Europe	9	2,115	4,015	18,660	4,797
Latin America	28	2,680	7,672	55,210	12,790
Asia	14	909	7,412	75,810	15,630
Africa	5	478	1,806	25,780	4,213

Using these relative measures of contemporary national prestige, I compared them to other national statistics theorized to influence prestige. Some of these determinants are

Gross Domestic Product (GDP), population size, per capita GDP, and land size (see Table 5). Due to missingness on one or more variables, several countries were excluded from the analysis that follows. For instance, GDP was taken from the World Bank, which does not contain data for the following countries in my list: French Guiana, Gibraltar, Guadeloupe, Guam, Guernsey, Holy See, Martinique, Mayotte, Montserrat, Nauru, Niue, North Korea, Palestine, Pitcairn, Taiwan, Tokelau, and Western Sahara.

Table 5. Top 10 mentions from tweets gathered 9/2/2015 to 9/12/2015.

Country	Tweets	% of total country mentions	2014 GDP (Trillion USD)[a]
India	76k	7.2%	2.0
Japan	68k	6.4%	4.6
Argentina	55k	5.3%	0.5
Syria	51k	4.9%	N/A
Venezuela	41k	3.9%	0.4
France	38k	3.7%	2.8
China	37k	3.5%	10.3
Brazil	34k	3.3%	2.3
Indonesia	30k	2.9%	0.9
Canada	27k	2.6%	1.8

[a]2014 GDP from the World Bank

A bivariate analysis of the independent variable of the 2014 GDP of each country as reported by the World Bank compared to the dependent variable of tweets for the ten day period in September gives us some insight into what impacts mentions on Twitter. A scatterplot of the two variables (Figure 13) depicts nations clustered near the origin, meaning that nations with low GDP often have low mentions, although that is not always

the case, and there are several unexpected deviations from the normative association. Because of the wide disparity in both Twitter mentions and GDP, the visualization becomes more useful after a log transformation for both variables (Figure 14).

The results from the United States are somewhat surprising. The United States has the highest GDP and slightly above average mentions, which seems to break the model. This is likely measurement error arising from the many different names by which the United States is known in popular discourse. Twitter has a limit of 140 characters per tweet. “United States” is a lengthy 13 characters while “US” or “USA” has only two or three. The United Kingdom likely has a similar measurement issue. “UK” is much shorter and conveys the same meaning.

Because of this potential measurement error, we can predict that France will have a larger number of mentions than the United Kingdom, since France does not have a common abbreviation, nor would it save many characters. Both France and the United Kingdom had nearly the same GDP in 2014 (around 2.9 trillion USD), but “France” had over 38000 mentions, while “United Kingdom” had only 2600.

The second highest GDP after the United States was China. China has a nice short name and since the United States has an issue with abbreviation, China should then be first, if GDP and mentions are both similar measures of prestige. However, China comes in seventh, and not first.

In order to figure out what might impact the number of mentions besides GDP, I investigated several other metrics theorized to be positively correlated with countries. Figures 15 and 16 show Twitter mentions versus 2014 population size. India has the second largest population after China. This might be a clue as to why both countries are in

the top 10 most mentioned countries. Another reason might be measurement error. “India” is counted whenever another word is mentioned that contains “India”, such as “Indiana” or “Indian”. This gives “India” many extra hits. With Twitter, there is the potential to collect millions of tweets in a very short period of time. Huge data sets can make thorough analysis difficult; however, a sample can be chosen to determine a likely percentage that are false positives, although such weighting is beyond the scope of this analysis.

Japan is next on the list, and has many of the benefits of India as far as measurement error. Words like “Japanese” increase the “Japan” count, for instance. Another metric for explaining high mentions might be GDP per capita (Figures 17 and 18). Wealthier countries have more access to devices that can utilize Twitter as well as better Internet connectivity. Japan has a large population, and a high GDP per capita (nearly 5x that of China). A large per capita GDP by itself is not a great measure, though, as many small countries have a relatively high GDP per capita. For instance, the top 5 highest GDP per capita are Liechtenstein, Luxembourg, Norway, Qatar, and Bermuda, each of which are very small countries (two have populations below 100,000).

Syria is a great example of how a country can gain mentions without having any of the regular prestige metrics. Syria has had a devastating civil war going since 2011, destroying their GDP. Refugees from this war-torn country are flooding into other countries seeking relief from religious persecution from ISIS. These Syrian refugees have been contentious in many countries, increasing discussion. This is an example of how measuring mentions can be superior to other methods of determining prestige. War, famine, natural disasters, and similar events increase discussion of a nation, but it often is not reflected in other statistics, such as GDP or population.

Multivariate Analysis and Regression

Linear models for tweets mirrors what was done with citations in literature. Nested models were created using the concepts: wealth, size, history, and religion. The same independent variables were used and tweets were log transformed (to reduce skewness) and used as the dependent variable. Table 6 shows these models with Table 7 showing the standardized results. As before, Model 1 represents the individual wealth of a citizen with GDP per capita (log transformed). Model 2 represents the size of a country, using a factor score that measures a country's size, broadly defined by GDP, population, and total land area. Model 3 represents the history of the country, including years since independence and a factor score that measures a country's level of democracy and political and civic freedoms. Model 4 measures the effect of a country's religious culture on citations, here measured with dummy variables representing the majority religion in each country. Finally, Model 5 contains all of the variables, allowing for the estimation of the joint effect of all variables. I again consider model fit using the adjusted R-square, where a higher value indicates better fit.

The first model uses only one independent variable, GDP per capita (log transformed). Unlike with the historical prestige, individual wealth of the citizens of a country has a small impact on how many tweets mention that country. The coefficient was small, but significant (beta=-0.196, $p<0.05$), showing that poorer countries seem to be mentioned more often than wealthier ones (per capita). The adjusted R-squared was much larger this time, but not high enough to be considered a good fit.

The second model shows the impact of the size of the country on mentions on Twitter. Similar to historical mentions, the size of a country enhances

Table 6. Correlates of a country's cumulative citations in tweets.

	Model 1	Model 2	Model 3	Model 4	Model 5
GDP per capita (logged)	-0.196*				-0.0434
(stnd err)	(-2.48)				(-0.56)
Size		0.431***			0.386***
(stnd err)		(6.58)			(5.24)
National independence prior to 1800			0.127		-0.105
(stnd err)			(0.61)		(-0.69)
Democracy factor score			-0.0269		0.133*
(stnd err)			(-0.51)		(2.34)
Catholic				-0.308	-0.275
(stnd err)				(-1.27)	(-1.44)
Protestant				-0.701**	-0.658**
(stnd err)				(-2.76)	(-2.93)
Orthodox				-0.498	-0.555
(stnd err)				(-1.18)	(-1.84)
Muslim				-0.245	0.0632
(stnd err)				(-0.85)	(0.29)
Intercept	18.41***	16.55***	16.53***	16.94***	17.29***
adj. R-sq	0.103	0.485	-0.035	0.104	0.564

* p<0.05, ** p<0.01, *** p<0.001

NOTES: Dependent variable is total citations in tweets from INSERT DATE, log transformed to correct for skew. All independent variables are averages of each variable over all occasions of measurement from 1980-2009. Standard errors in parenthesis. N=46.

visibility, increasing discussion. The same factor scale as before was used, combining the collinear variables of population, GDP, and land area. As expected, a relatively high adjusted R-square shows that using size as an indicator for visibility is a pretty good fit. The coefficient for size (beta=0.431, p<0.001) was again statistically significant.

The third model uses variables for age (how long the country has been independent) and liberty (democracy factor score). Neither variable was statistically significant. As with historical prestige, this model is a very poor fit, and had a negative adjusted R-square.

The fourth model, religious culture, was more significant than with historical mentions in literature. The adjusted R-square was similar to the first model, so not a good fit, but the variable for being Protestant had a coefficient (beta=-0.701) that was statistically significant with p<0.01. Just like before, a country was marked as a religion if a majority of the citizens practice that religion. Catholic, Orthodox, and Muslim were also variables, but none were significant.

Table 7. Correlates of a country's cumulative citations in tweets (standardized).

	Model 1	Model 2	Model 3	Model 4	Model 5
GDP per capita	-0.35*				-0.08
Size		0.70***			0.63***
National independence prior to 1800			0.10		-0.08
Democracy factor score †			-0.08		0.41*
Catholic				-0.28	-0.25
Protestant				-0.59**	-0.55**
Orthodox				-0.19	-0.21
Muslim				-0.16	0.04
adj. R-sq	0.103	0.485	-0.035	0.104	0.564

* p<0.05, ** p<0.01, *** p<0.001

NOTES: Dependent variable is cumulative (total) citations in the ngrams corpus from 1870-2000. All independent variables are averages of each variable over all occasions of measurement from 1980-2009. Standard errors in parenthesis. N=46.

The fifth model is the fully specified model and includes all variables. As expected, it yielded the highest adjusted R-square. Two variables were significant in the model, size (beta=0.386, p<0.001) and Protestant (beta=-0.658, p<0.01). Wealth was not significant, which makes sense due to the high collinearity with size. Table 7 reports standardized coefficients, but reflects the findings of Table 6. Model 5 still has the best adjusted R-

square, which makes sense due to the small, yet significant effects of being Protestant and individual wealth (Models 1 and 4). Size is still the largest factor in predicting visibility and mentions in tweets.

So if Twitter mentions are a measure of contemporary, or instantaneous prestige, other variables would need to be added to a model in order to explain the number of mentions. Measurement error might be reduced by monitoring all names of a country. Longer monitoring periods can also reduce artificial elevation for a country if some event, like a sporting event or natural disaster, happens during the time of collection. The extremely large amount of data, though, would restrict a large amount of terms being monitored for a long period of time. Monitoring for just ten days for this list of terms collected enough tweets to fill over 163GB of disk space. Because of the large size, processing the data using conventional computer hardware specifications took a parsing program over 16 hours to audit the data. Of course, since these were English names, this would also only be applicable to the Anglosphere.

Combining Citation Data with the Nation Brand Index

The final set of models (see Table 8) uses the Nation Brand Index (NBI) as the dependent variable and uses all of the variables from the other models, including historical mentions in literature (logged) and contemporary mentions on Twitter (logged). The first model is just reflects citations, both historical and contemporary, which are used in all models. Model 2 represents the wealth of the country along with citations. Model 3 adds size to citations. Model 4 adds history (years since gaining independence and the country's level of democracy and political and civic freedoms). Model 5 adds religious culture with

citations. Model 6, as before, contains all of the variables to consider the joint effect. I still interpret a higher adjusted R-square as being a better fit.

Table 8. The determinants of national prestige: Correlates of NBI scores

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Ngram citations (logged)	0.607***	0.480***	0.693***	0.331***	0.426***	0.297***
(stnd err)	(5.74)	(7.31)	(5.86)	(3.95)	(3.86)	(3.90)
Twitter citations (logged)	-0.647*	-0.067	-0.414	-0.265	-0.118	-0.032
(stnd err)	(-2.62)	(-0.42)	(-1.44)	(-1.49)	(-0.43)	(-0.17)
GDP per capita		0.655***				0.501***
(stnd err)		(8.69)				(5.93)
Size			-0.301			0.124
(stnd err)			(-1.54)			(1.04)
National independence prior to 1800				0.555**		0.311
(stnd err)				(2.71)		(1.87)
Democracy factor score				0.298***		0.149*
(stnd err)				(5.87)		(2.28)
Catholic					0.420	-0.096
(stnd err)					(1.30)	(-0.46)
Protestant					0.971*	0.016
(stnd err)					(2.62)	(0.06)
Orthodox					0.129	-0.049
(stnd err)					(0.23)	(-0.15)
Muslim					-0.310	-0.202
(stnd err)					(-0.79)	(-0.82)
Intercept	4.024	-10.34***	-0.782	0.645	-3.142	-7.45*
adj. R-sq	0.412	0.785	0.430	0.722	0.519	0.842
* p<0.05, ** p<0.01,	*** p<0.001					

NOTES: Dependent variable is a factor score derived from country rankings collected for the Nation Brand Index studies in 2008 and 2009. All independent variables are averages of each variable over all occasions of measurement from 1980-2009. Standard errors in parenthesis. N=46.

In the first model, the two log transformed variables for citations are used. These have high collinearity, so only one is highly significant, the Ngram citations (beta=0.607, $p<0.001$). Twitter citations (beta=-0.647) do still have significance at $p<0.05$, though. This model has a good fit with the adjusted R-square at 0.412.

The second model adds individual wealth to the two citation variables. GDP per capita (log transformed) has a significant coefficient (beta=0.655, $p<0.001$), but Twitter citations lose their significance. Ngram citations are still significant (beta=0.480, $p<0.001$). This model has a very good fit, with an adjusted R-square of 0.785.

The third model adds size to the two citation variables. Size doesn't seem to impact the NBI very much, though, an insignificant size variable and an adjusted R-square nearly the same as the citations alone.

The fourth model combines the citation variables with age and freedom. Age is significant at $p<0.01$ (beta=0.555), but the democracy factor score is significant at $p<0.001$ (beta=0.298). Ngram citations are still significant in this model (beta=0.331, $p<0.001$). This model has nearly the same fit as Model 2 with an adjusted R-square of 0.722.

The fifth model includes religious culture with citations. Ngram citations maintains its significance at $p<0.001$ (beta=0.426), but no other variables are as significant. Protestant does have significance at the $p<0.05$ threshold, however (beta=0.971). This does have an increased fit over Model 1, showing that religion does seem to have an impact on the NBI.

The sixth model combines all variables into one fully specified model. It yields the highest adjusted R-squared as expected. Ngram citations maintains its significance at $p<0.001$ (beta=0.297). GDP per capita also is significant at $p<0.001$ (beta=0.501), but no

other variable reaches this threshold. The democracy factor score (beta=0.149) does reach significance at the $p<0.05$ threshold, however. Even though this has the greatest fit, probably the best model would be GDP per capita (log transformed) and Ngram citations, in order to be the most parsimonious. Citations are still an important factor, though, in predicting a high Nation Brand Index score.

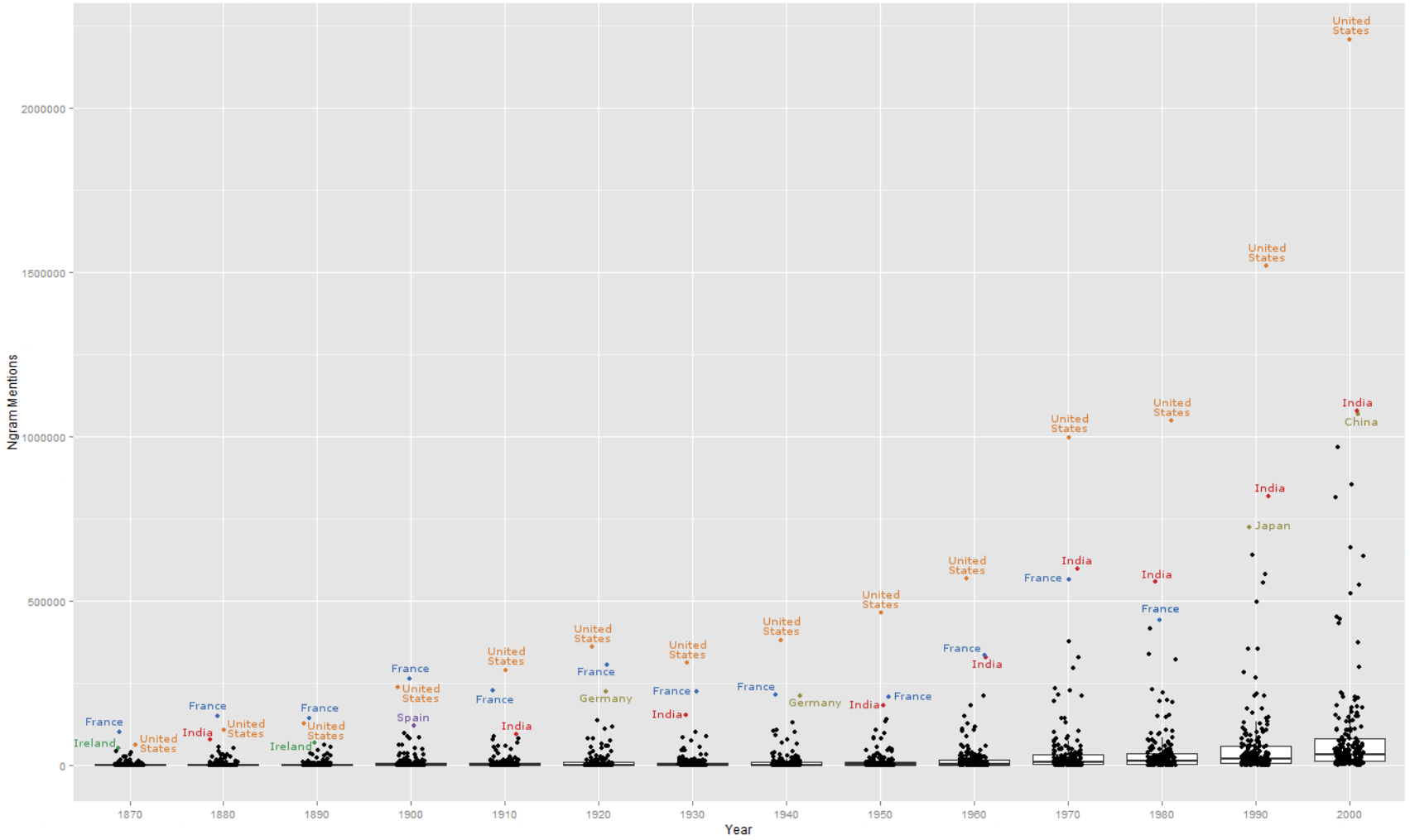

Figure 1: Raw mentions in literature of all countries by decade.

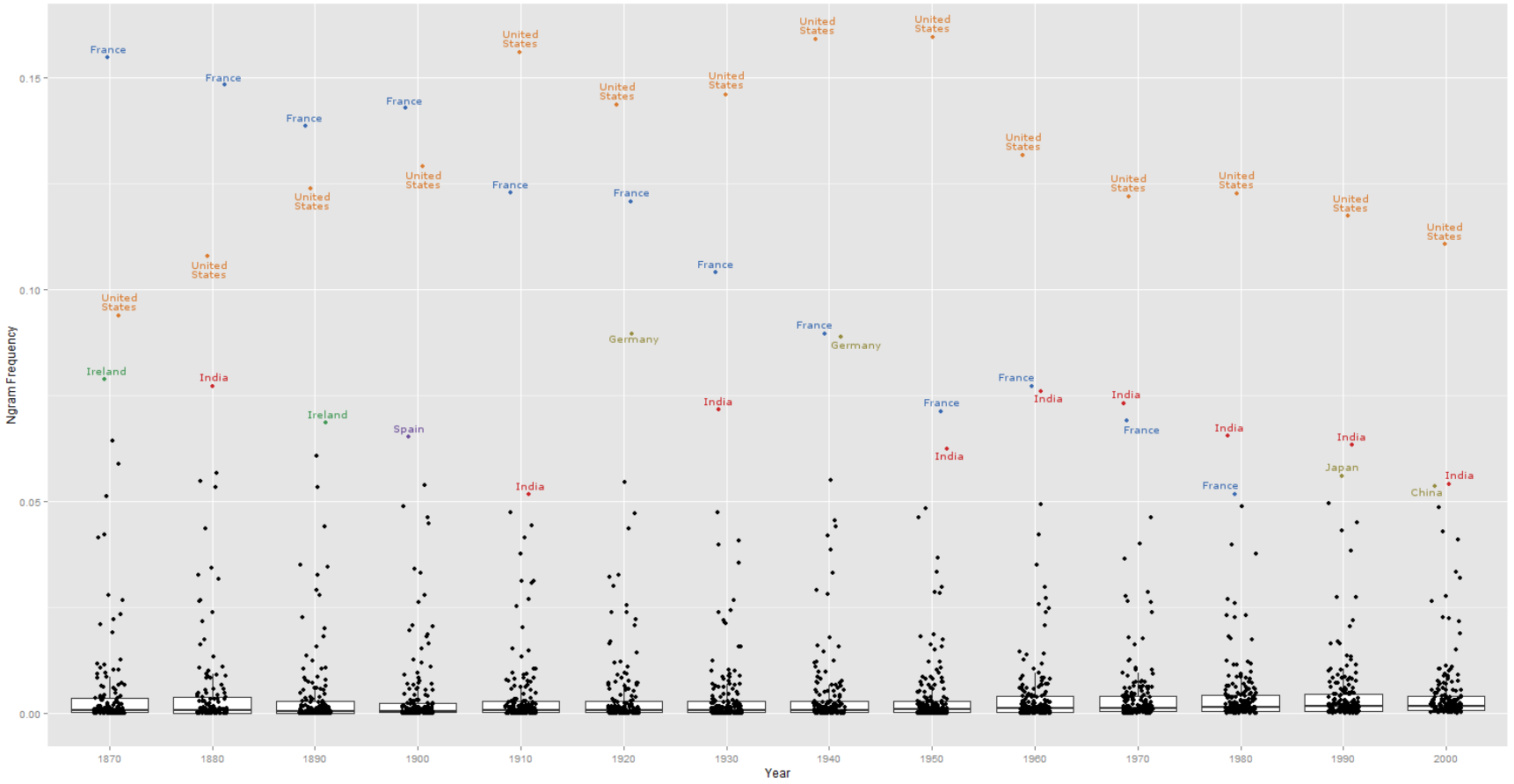

Figure 2: Frequency (%) of mentions for each country of all countries by decade.

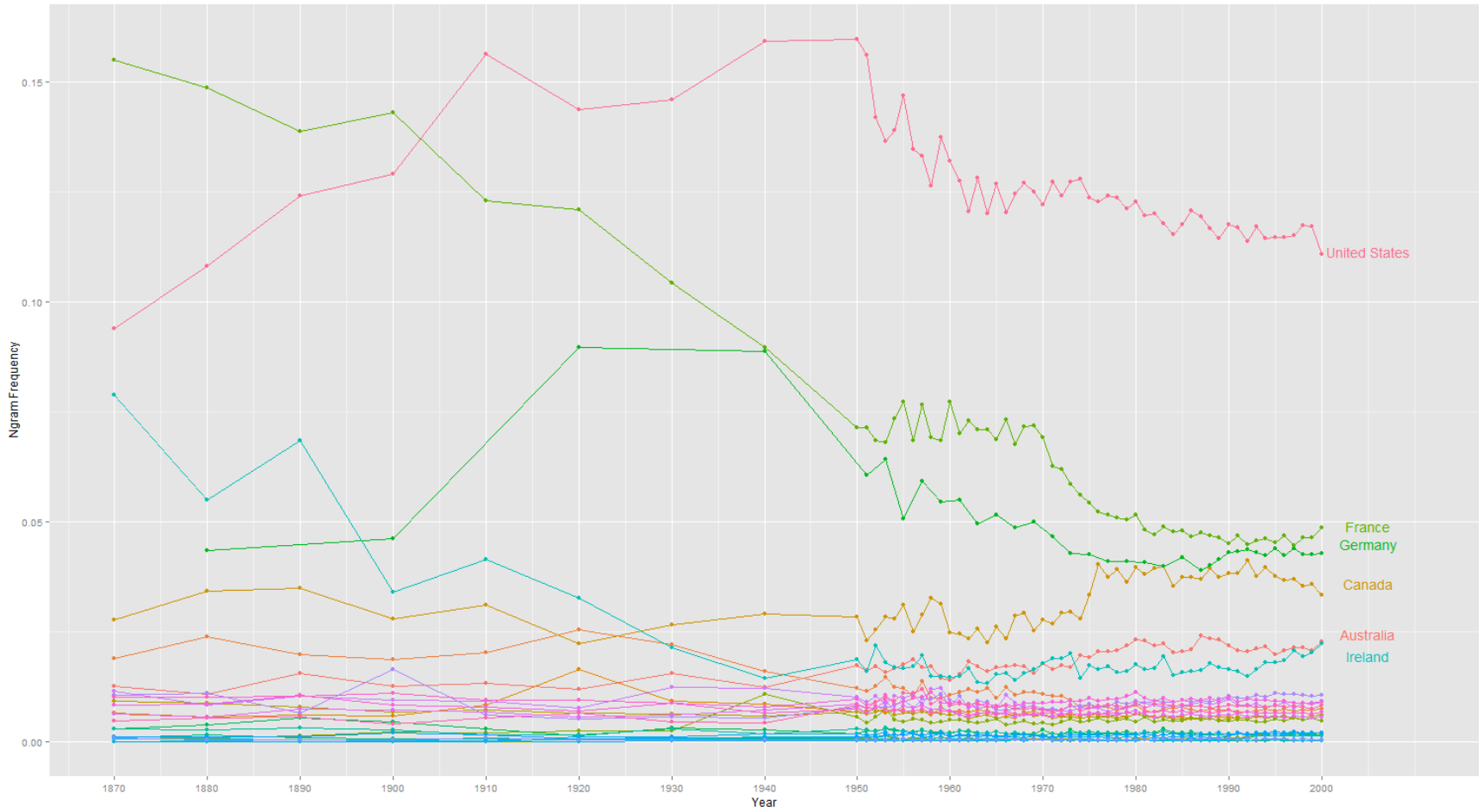

Figure 3: Frequency (%) of mentions for the 25 Northwestern European countries by year.

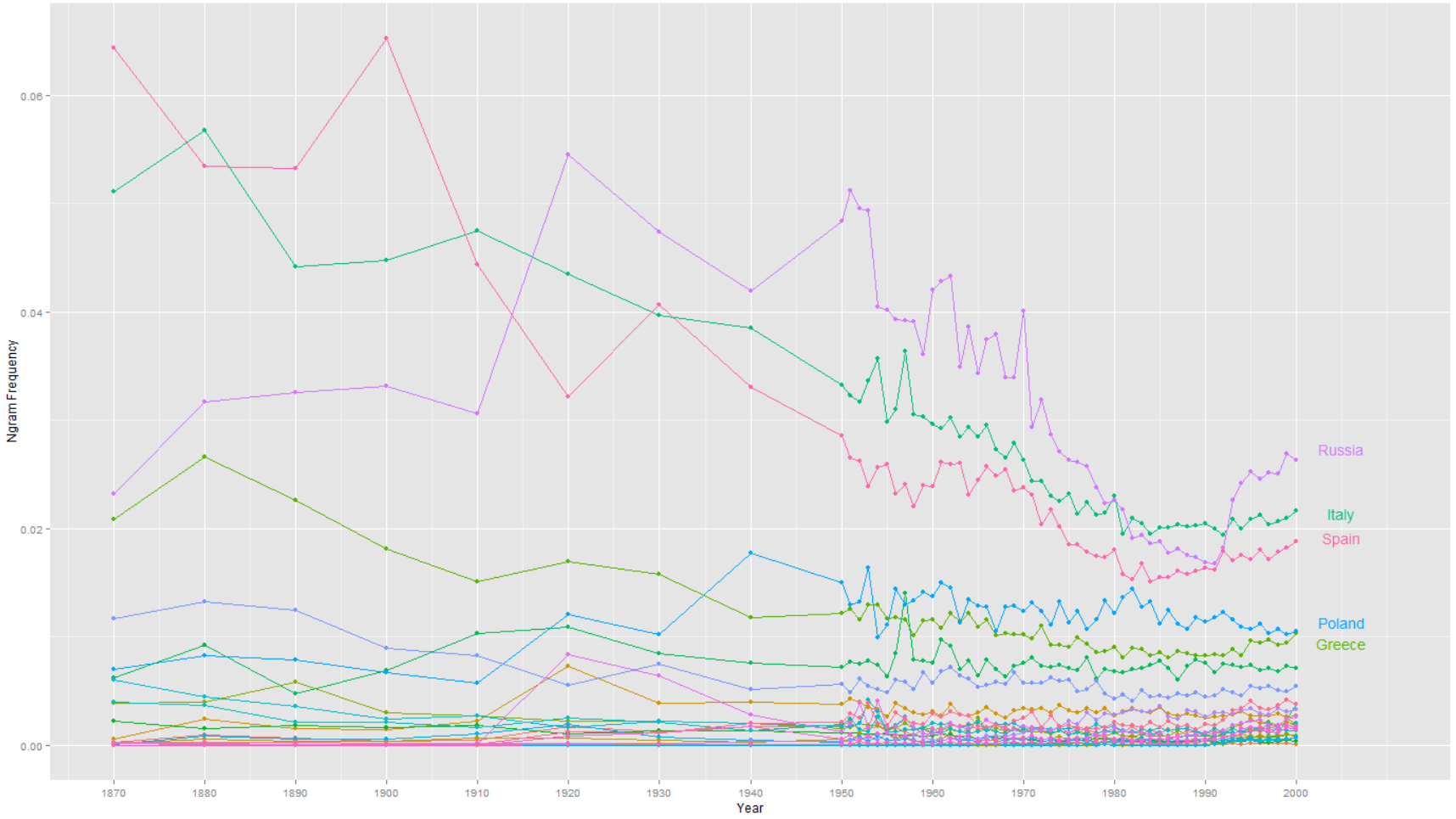

Figure 4: Frequency (%) of mentions for the 25 Eastern European countries by year.

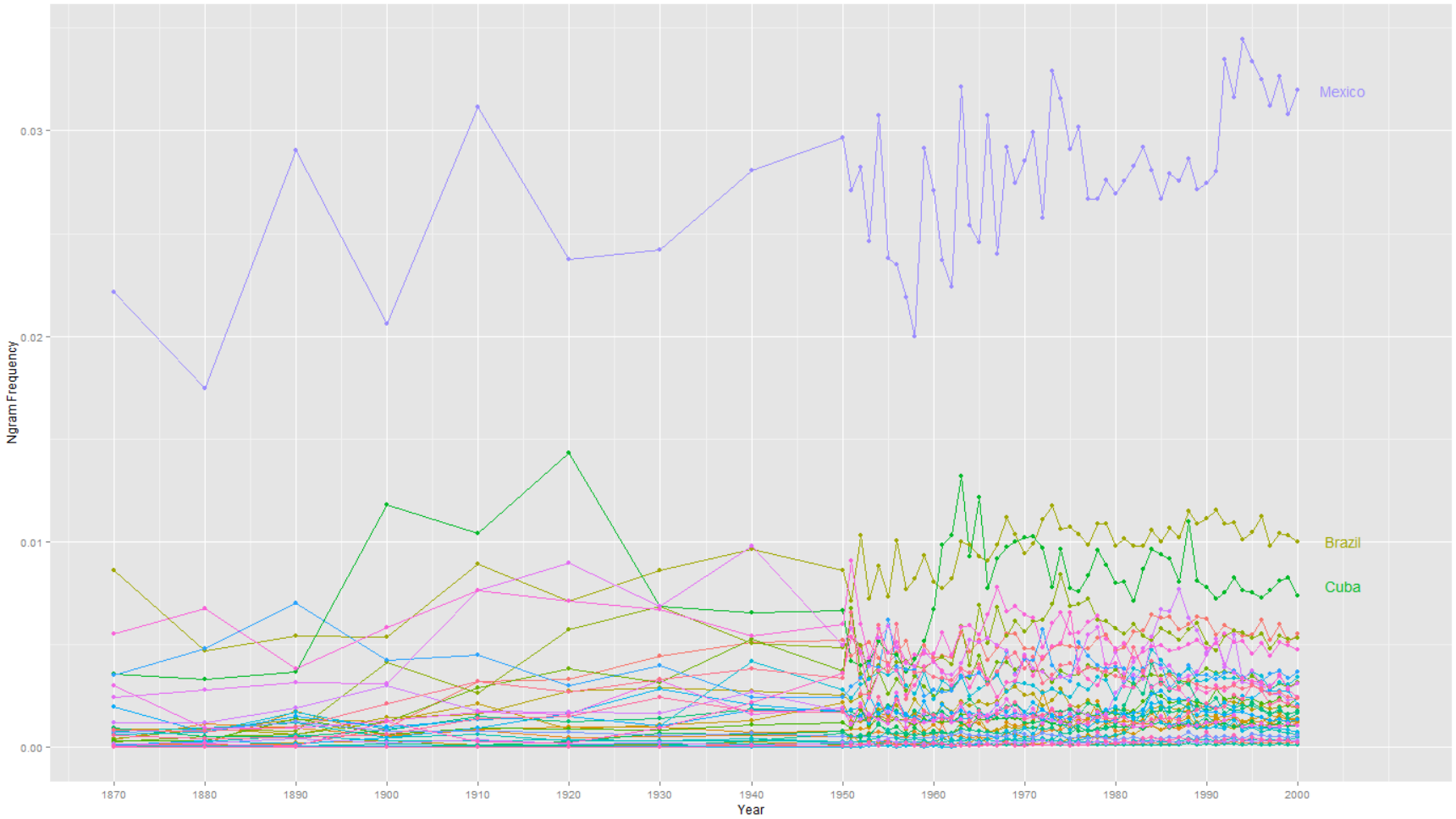

Figure 5: Frequency (%) of mentions for the 34 Latin America countries by year.

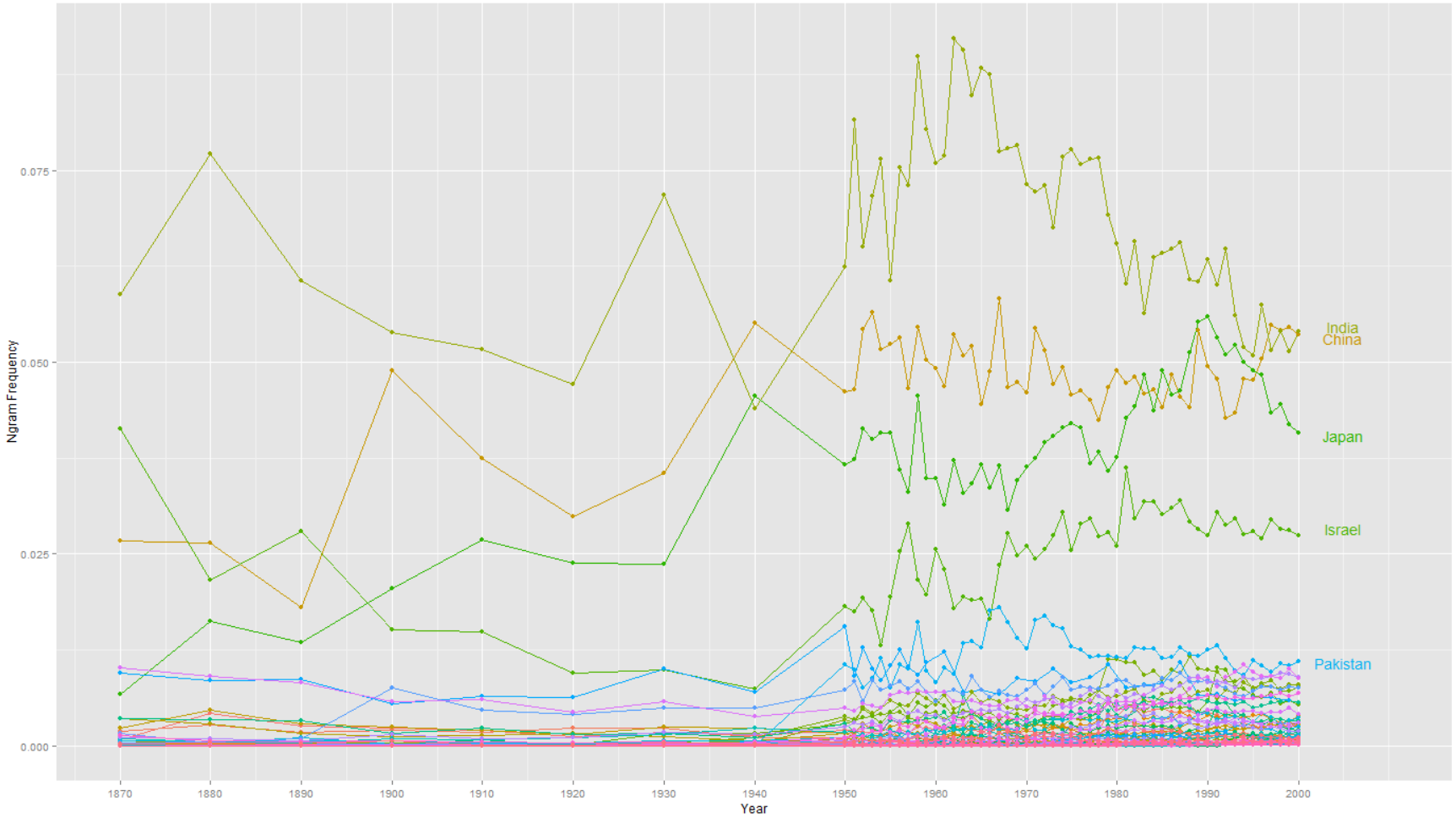

Figure 6: Frequency (%) of mentions for the 58 Asian countries by year.

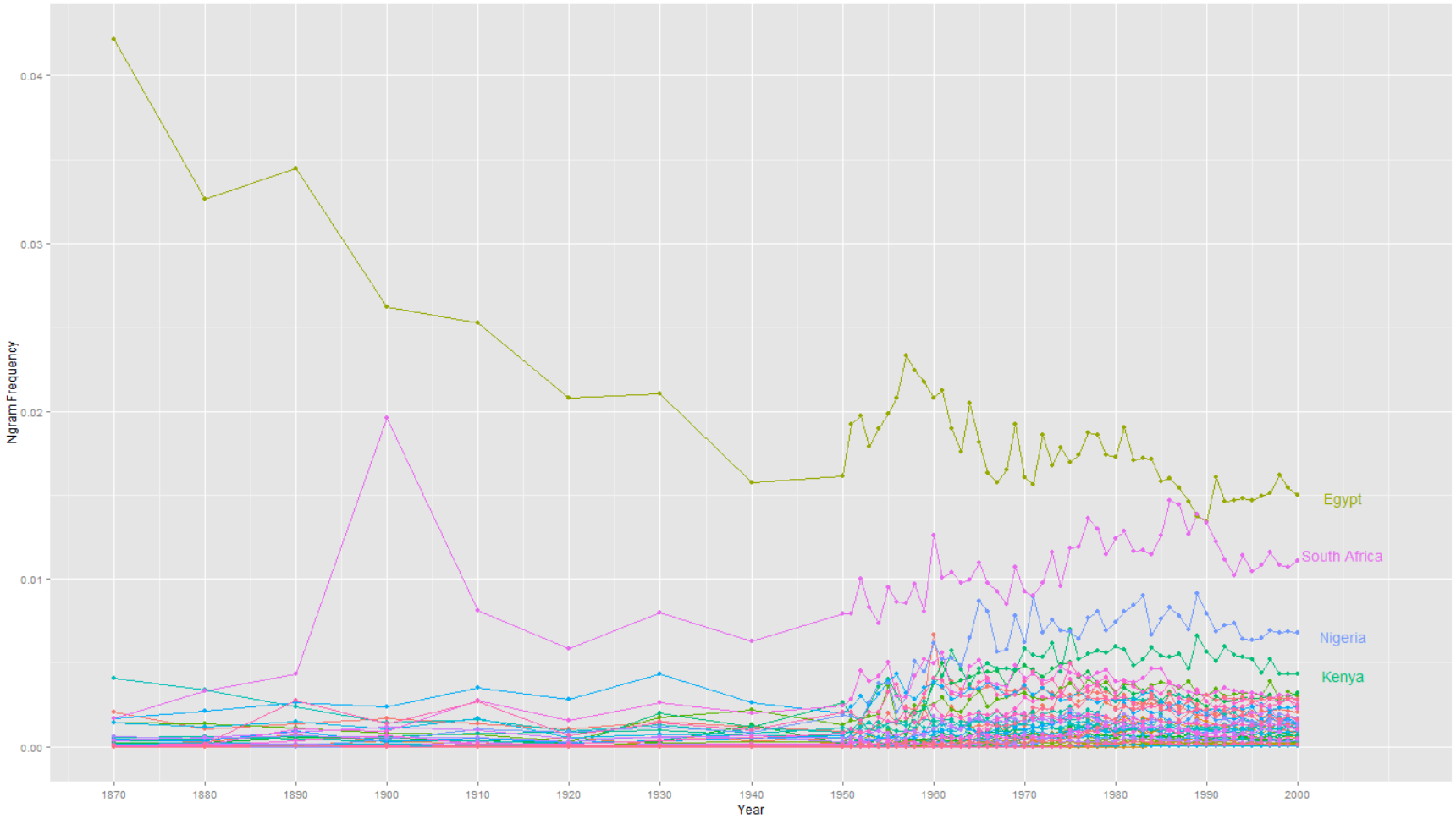

Figure 7: Frequency (%) of mentions for the 46 African countries by year.

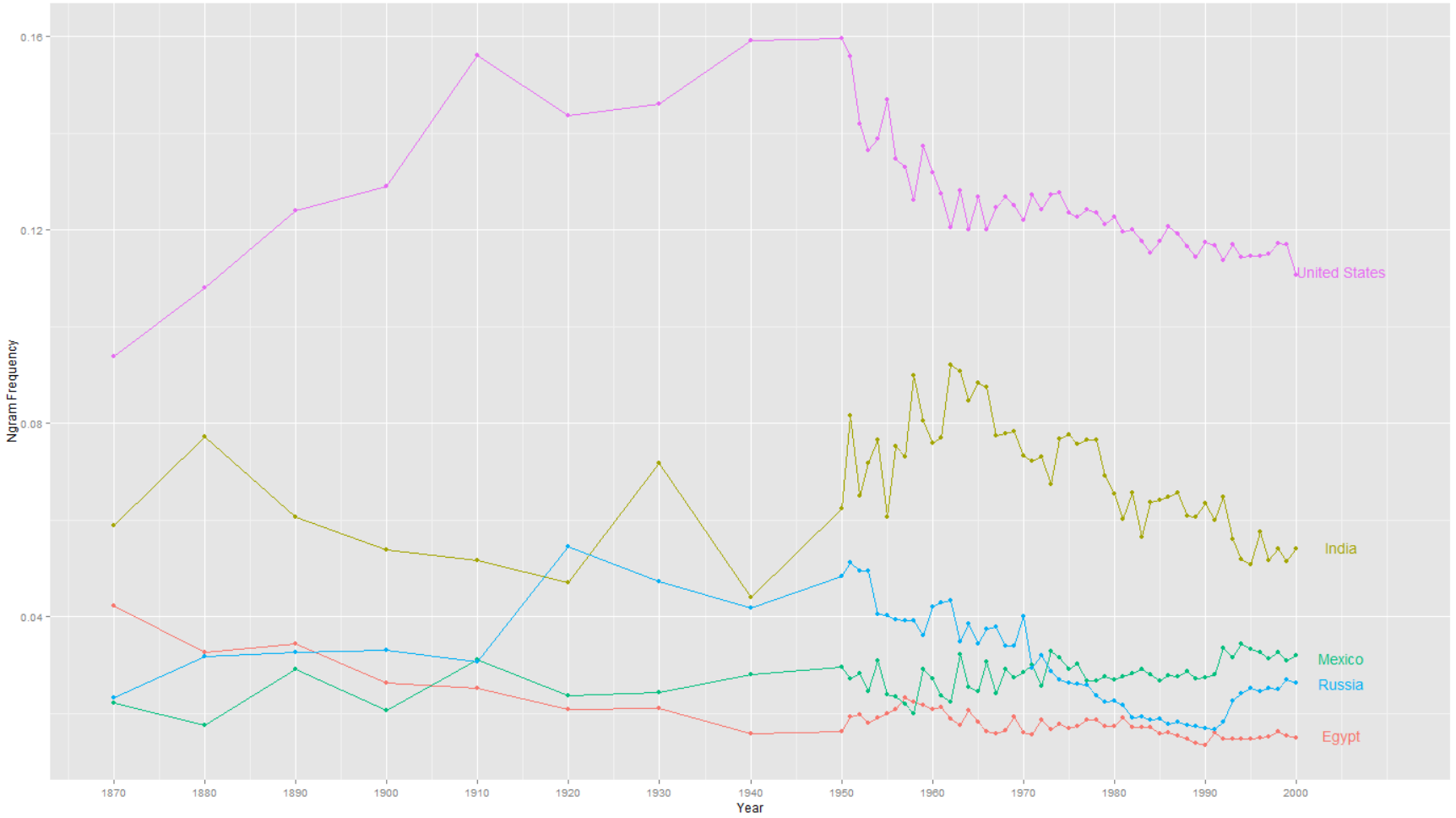

Figure 8: Frequency (%) of mentions for the most mentioned countries in each region by year.

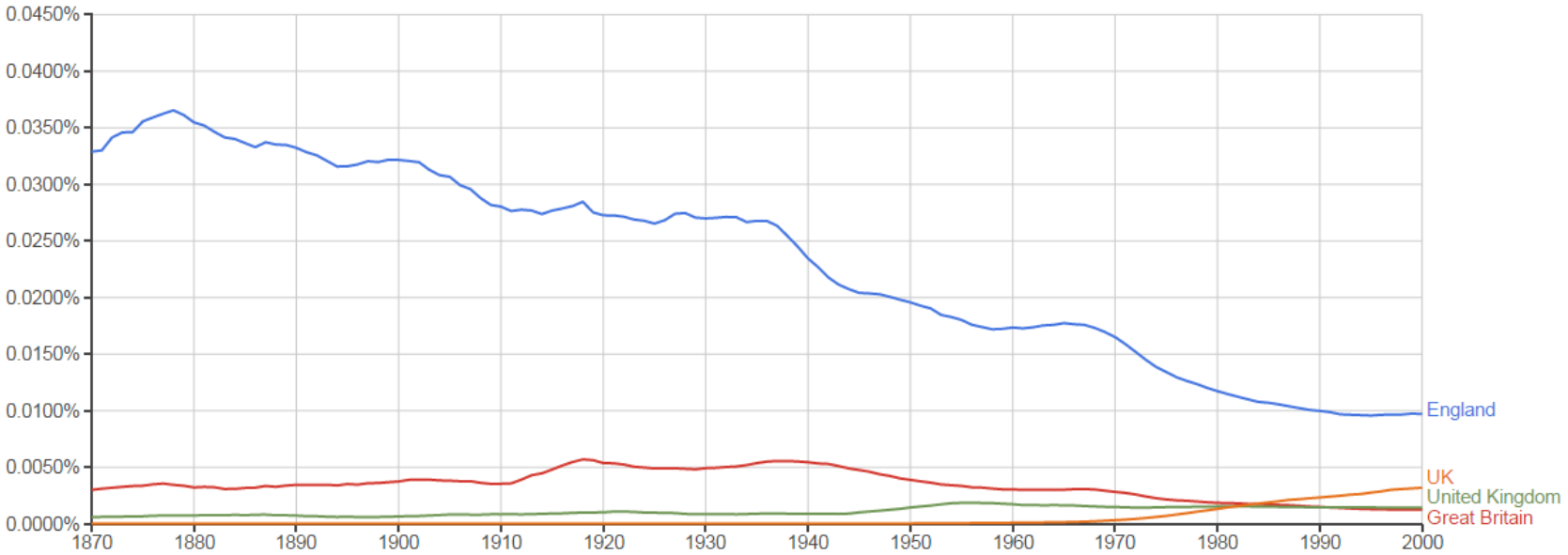

Figure 9: Usage frequencies of different names for the United Kingdom.

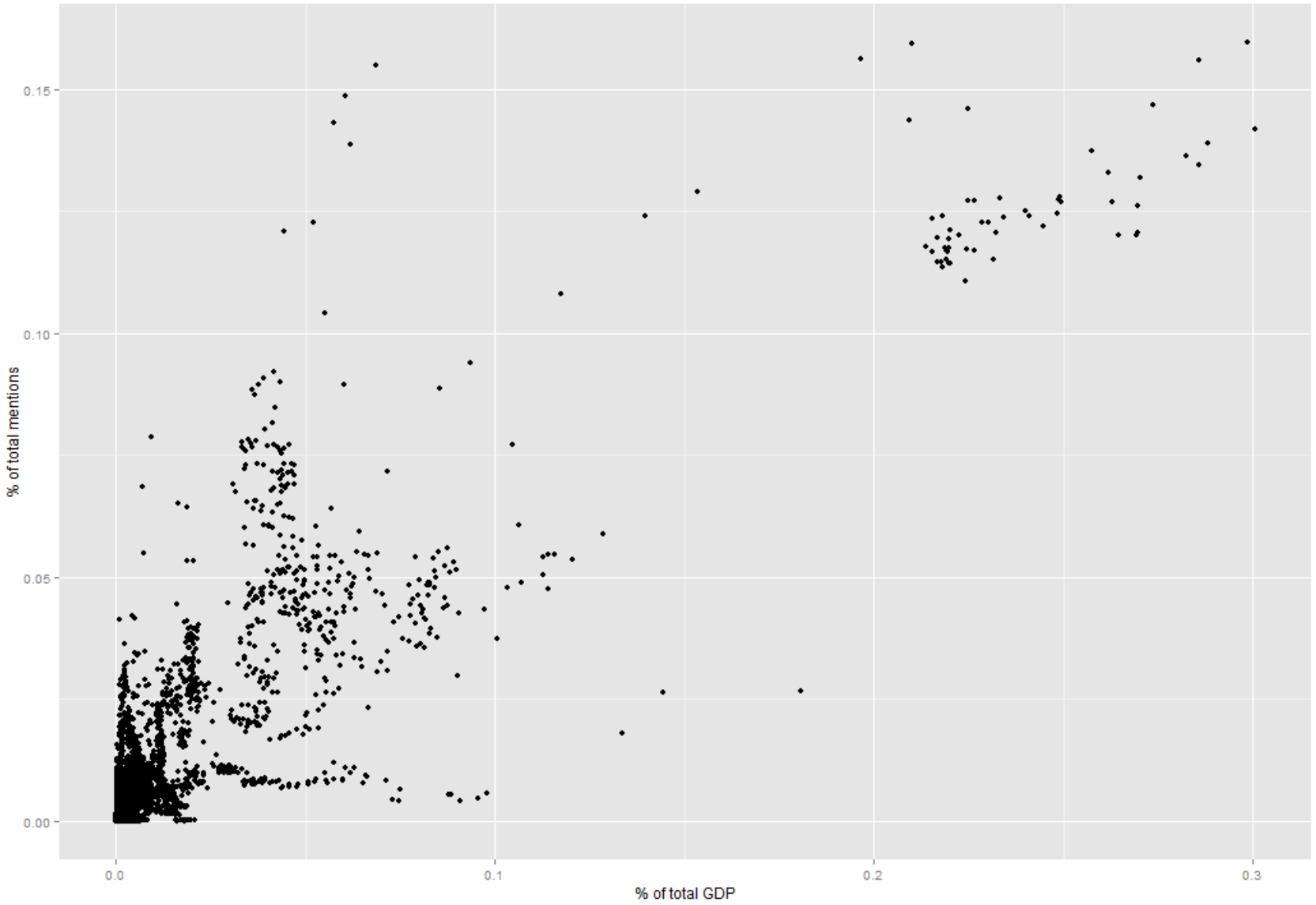

Figure 10: Frequency (%) of mentions of a country versus the share of total GDP for each country each year.

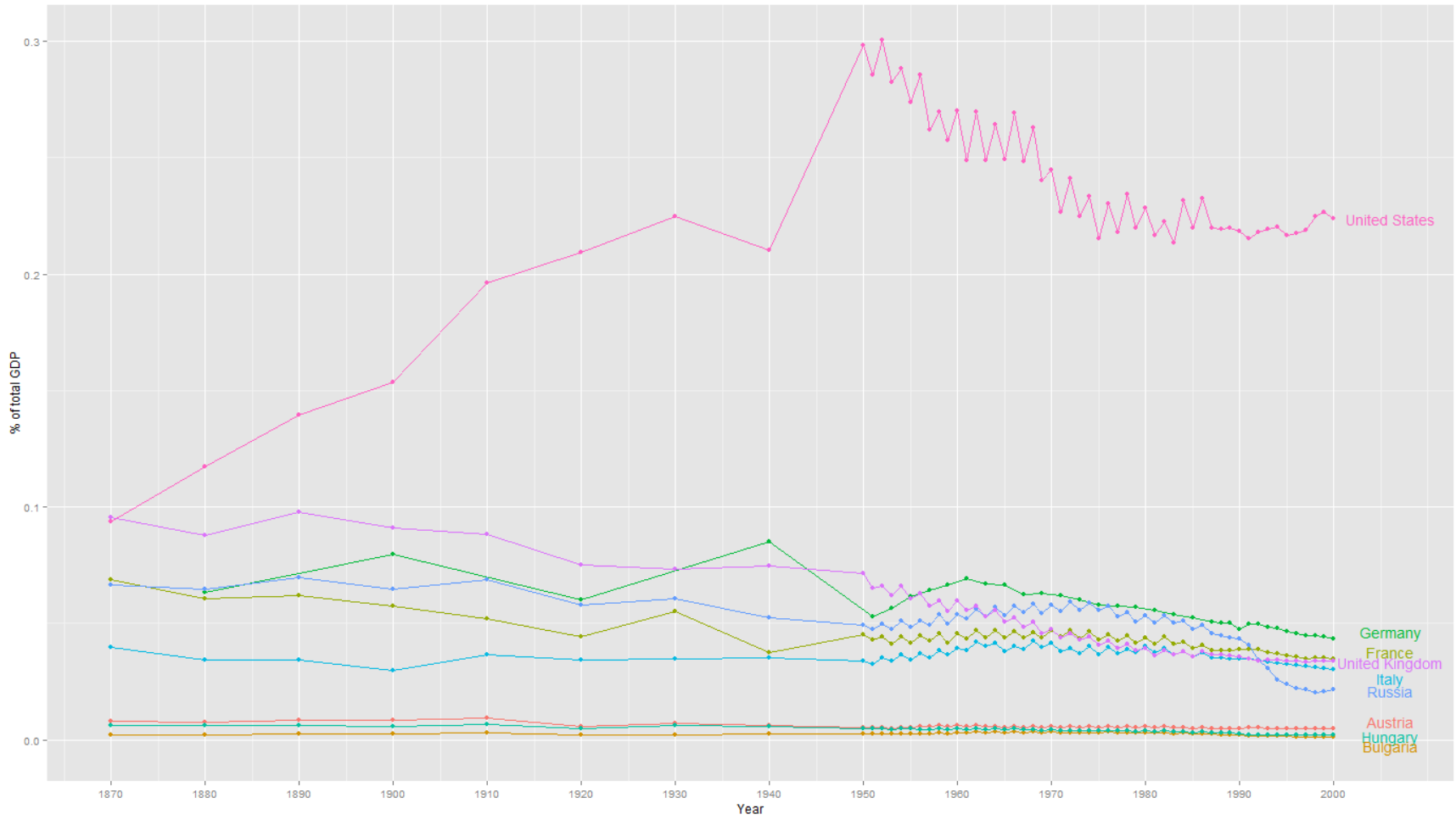

Figure 11: A comparison of WWI countries and their share of total GDP by year.

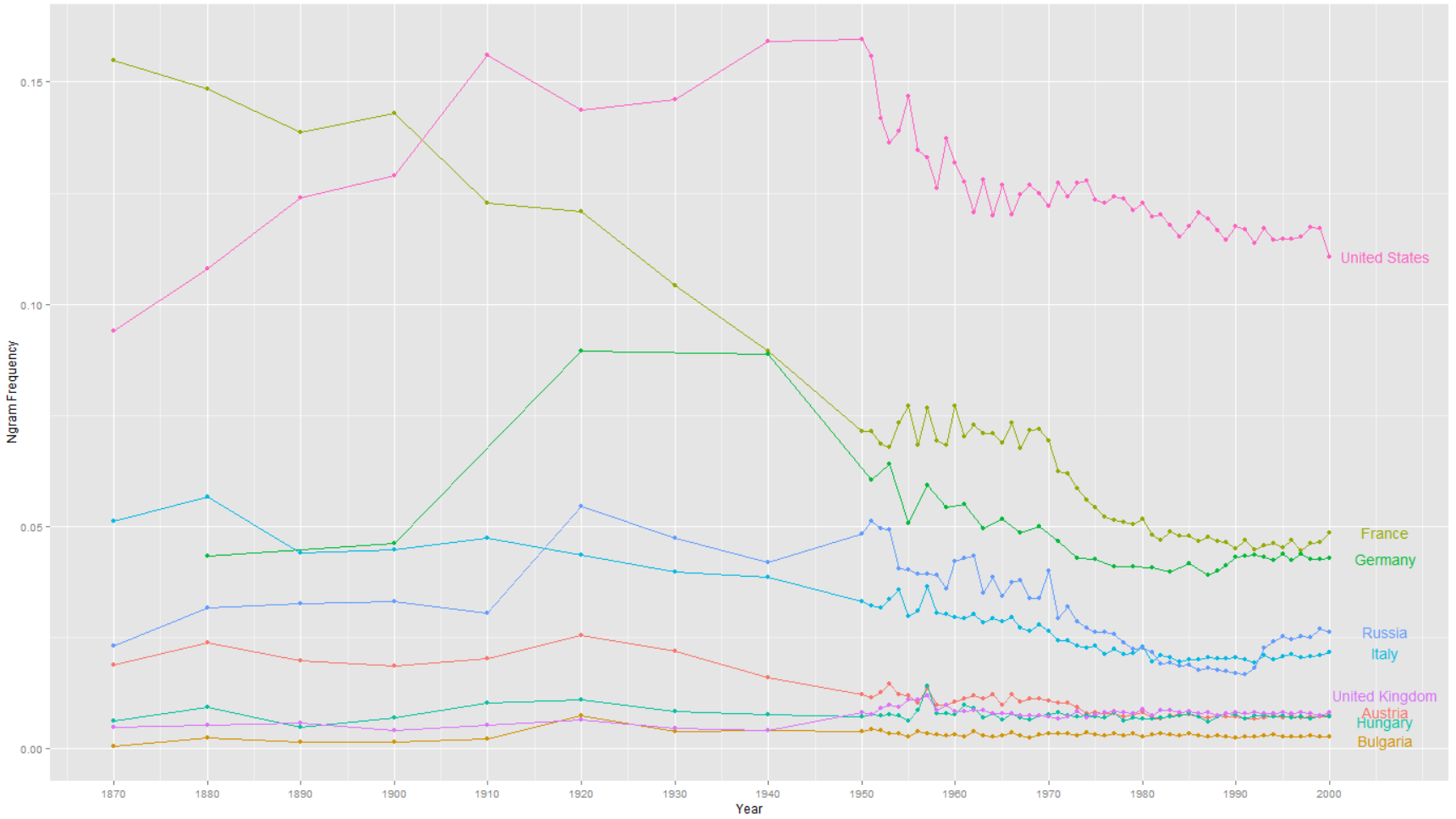

Figure 12: A comparison of WWI countries and their share of mentions in literature by year.

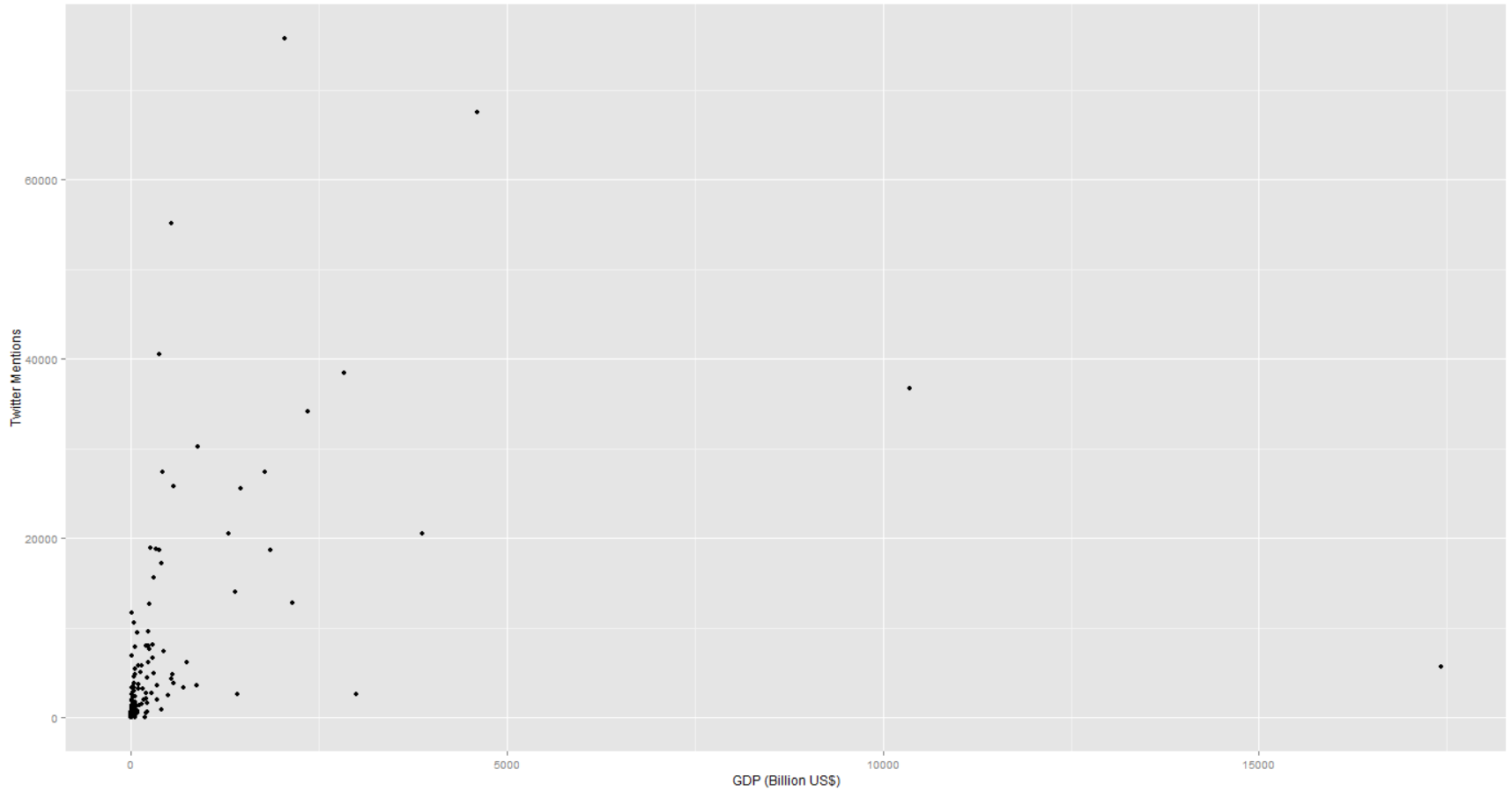

Figure 13: Total Twitter mentions in a 10 day period versus GDP by country in 2014.

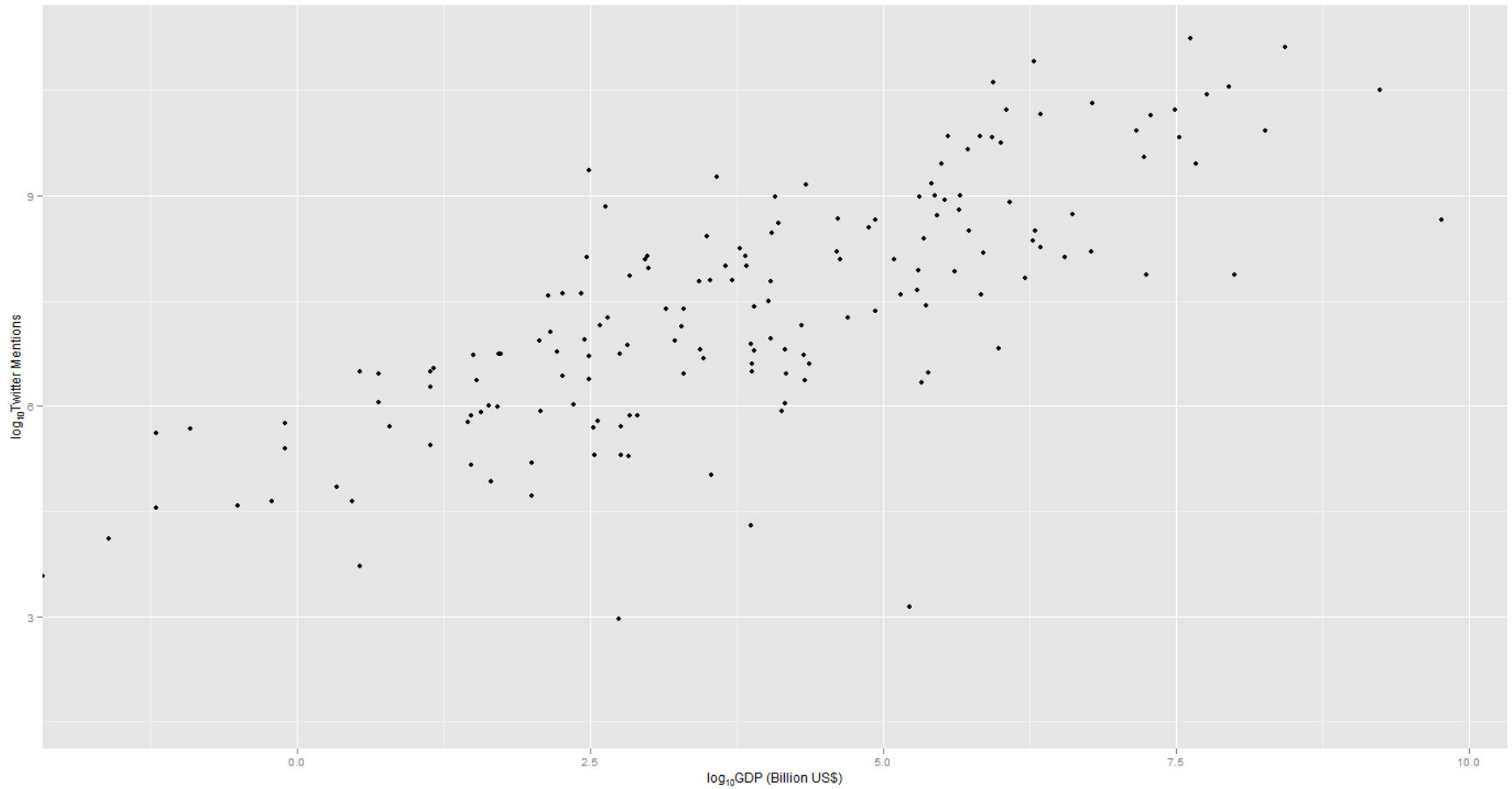

Figure 14: Total Twitter mentions in a 10 day period versus GDP by country in 2014 (log transformed).

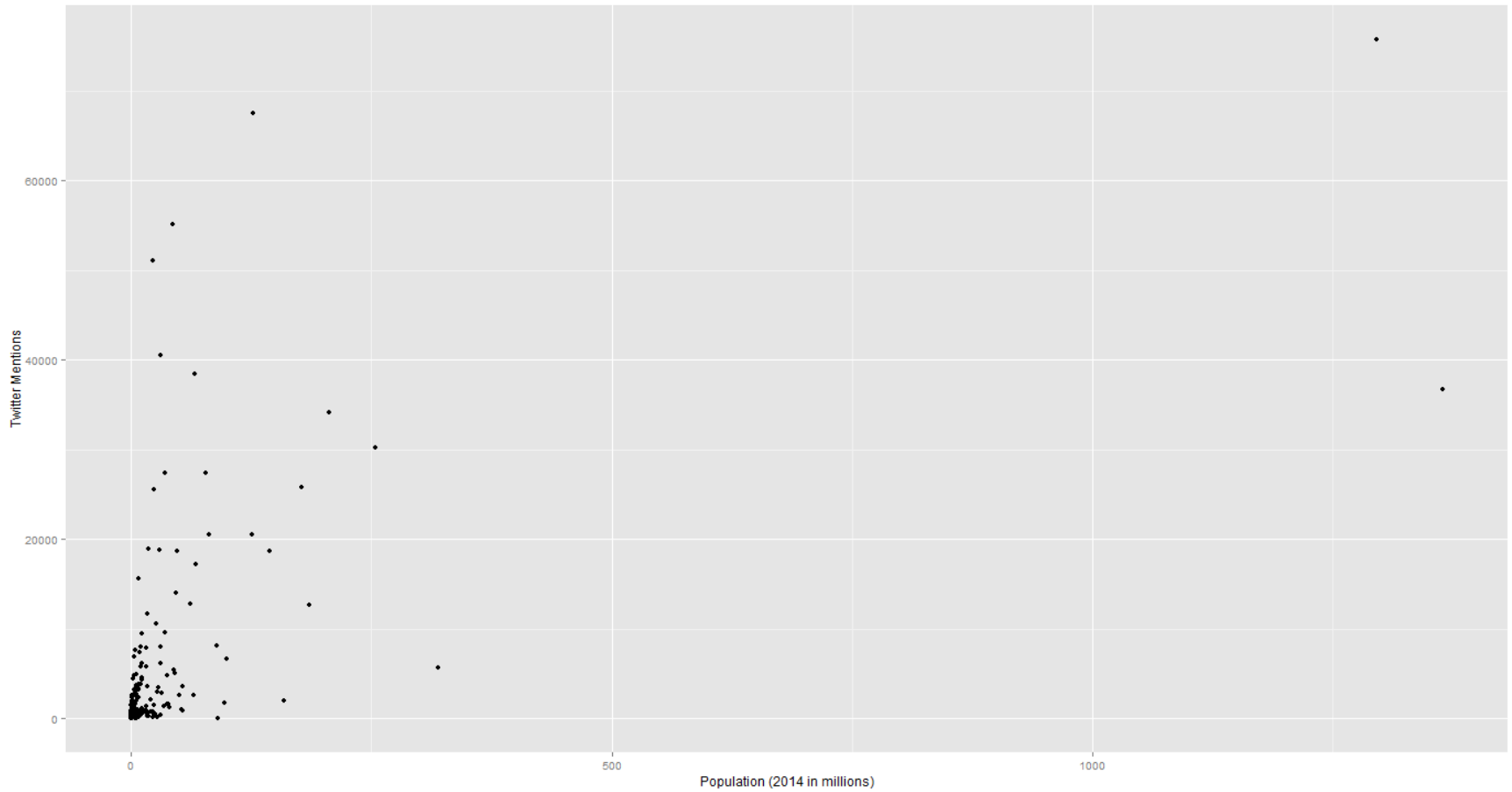

Figure 15: Total Twitter mentions in a 10 day period versus population by country in 2014.

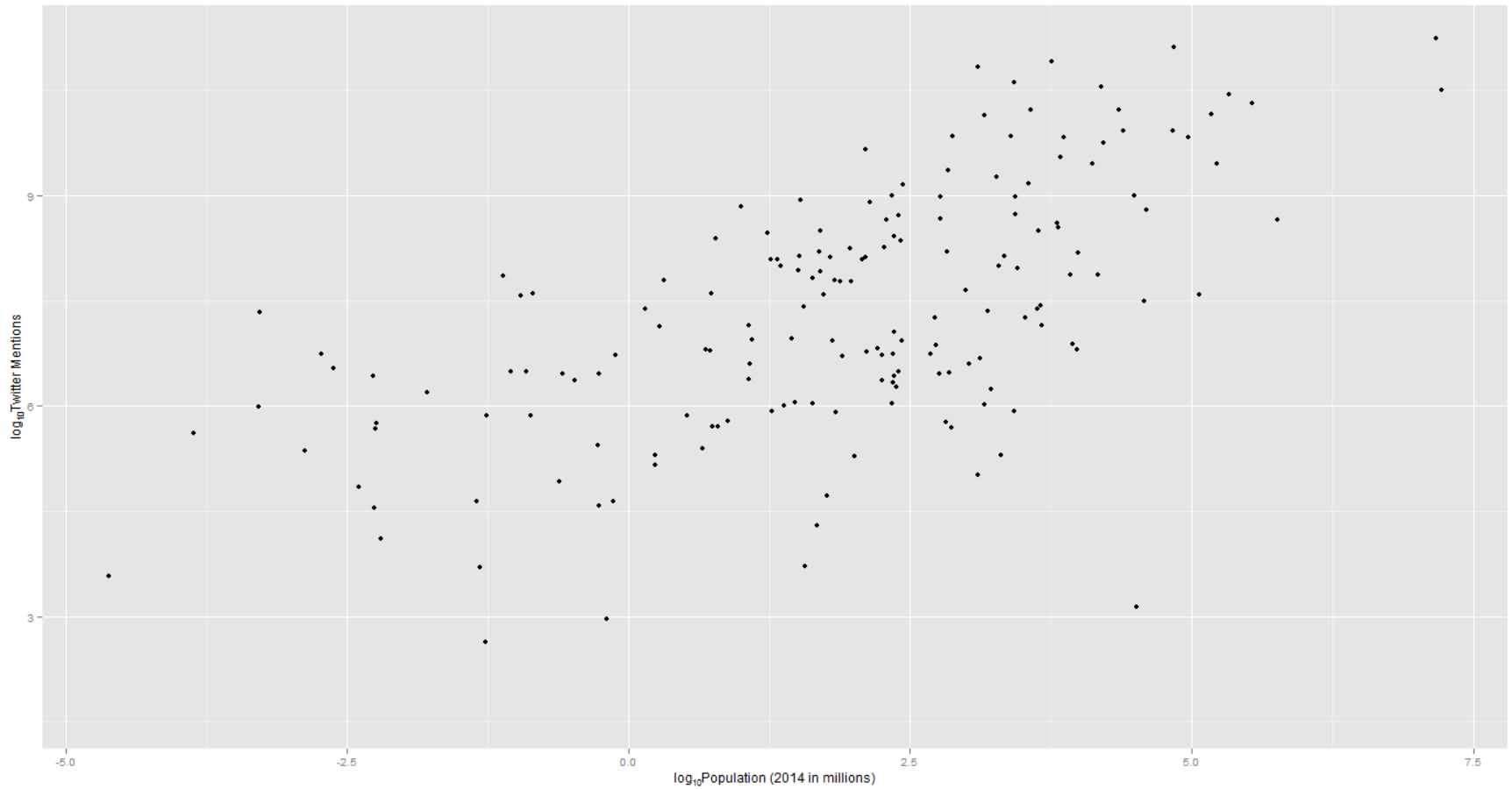

Figure 16: Total Twitter mentions in a 10 day period versus population by country in 2014 (log transformed).

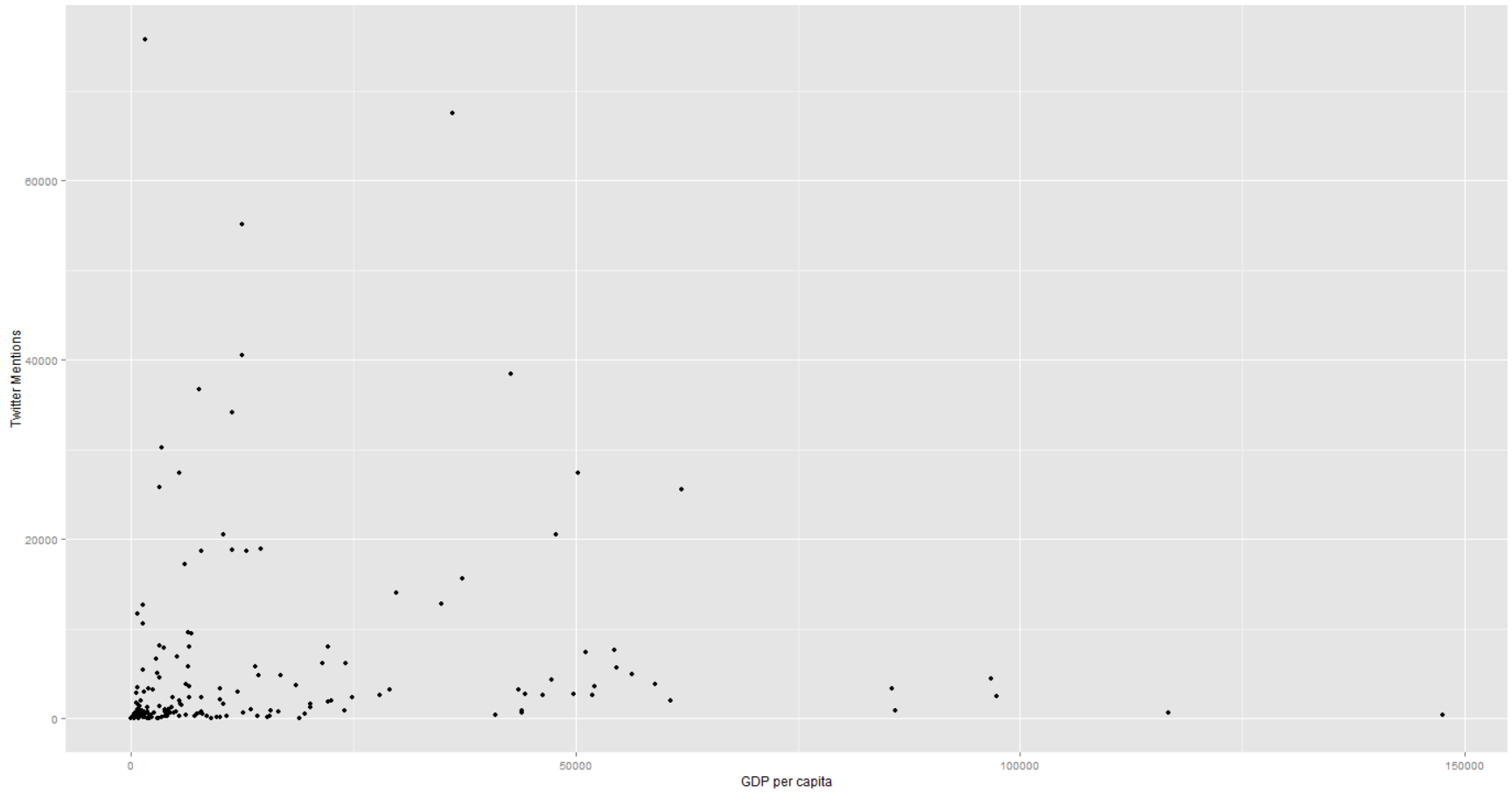

Figure 17: Total Twitter mentions in a 10 day period versus GDP per capita by country in 2014.

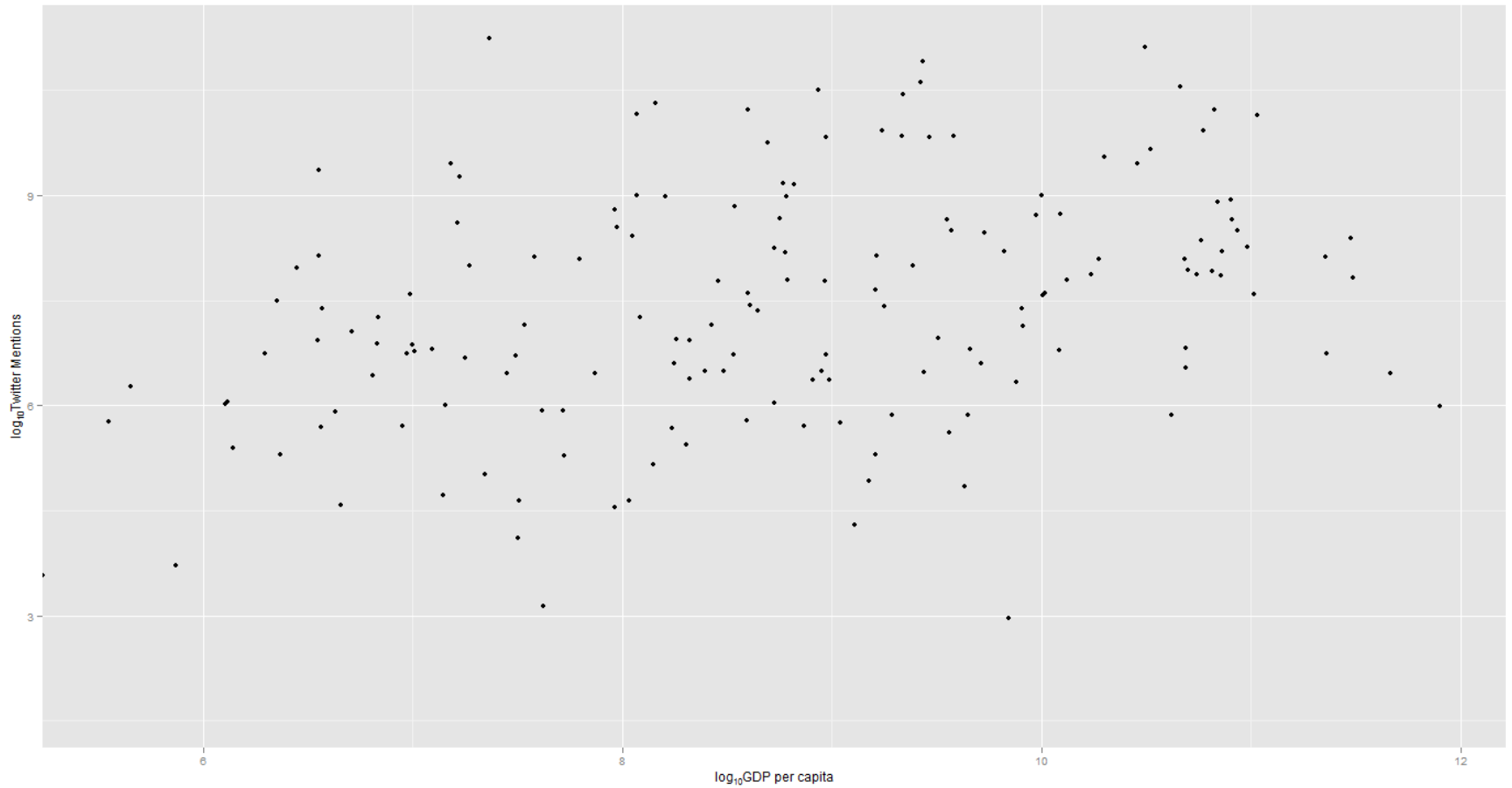

Figure 18: Total Twitter mentions in a 10 day period versus GDP per capita by country in 2014 (log transformed).

CHAPTER 5

CONCLUSION

The Internet has become a valuable resource for collecting data for all sorts of research. I discussed two tools to help make data collection more efficient. The first, scraping data from a web site, is useful for both collecting primary data and for a simpler approach to using secondary data. My case study was scraping Google's ngram frequency graph for specific data to avoid having to download the entire several terabyte database. The second tool, using an API to collect data, is most often used for secondary data collected by another source, making it easier to access. My case study for this was using the Twitter Streaming API to monitor tweets mentioning specific words.

Scraping data from a site is a powerful way to collect data and can be used if an API is not available to access data directly. The first part of my case study focused on a tool I created that scraped public data from the Google Books Ngram Corpus web site. The tool itself was made for public use, which creates new obstacles to overcome for usability. The case study measured mentions of countries in literature throughout the years. These mentions were used to quantify the prestige of a country, with more mentions relating to more prestige.

An API is an interface to connect to a data set offered by a company. It allows tools to be programmed to retrieve smaller amounts of data directly from the larger data set. For the second part of my case study, I used the Twitter Streaming API, which connects to a live stream of tweets. I monitored this feed for mentions of countries to measure prestige.

Once I had my mentions, I compared these data to other conventional means of measuring prestige, such as GDP and population. While there was some correlation comparing all countries' ngram frequency to GDP (r=0.85), grouping the countries revealed correlations as high as r=0.88 (Northwest Europe and USA) and as low as r=0.75 (Africa), meaning that the data fit pretty closely with the fitted regression line. A linear model of GDP and mentions on Twitter had a p-value of < 0.001.

Even though the focus of the thesis was the methods of data collection, the case study results revealed an important method of measuring prestige of a country. Each tool has positives and negatives, such as repeatability for the ngram scraping and content analysis of the tweets for the Twitter Streaming API. All in all, both methods were successful in achieving the original goals and can be duplicated for other data collection projects.

Importance

In the age of big data, scientists have access to more data than they can process. Tools can make retrieving a piece of this data more efficient. Two of these methods are web scraping and utilizing an API. I made tools using these methods to access two huge data repositories, the Google Books Ngram Corpus and the Twitter Streaming API. Both data sets are available for public use if you have the means to access the immense amount of data. The tools I developed brought both data sets down to a more manageable level.

The first method of web scraping can collect data from web sites. The site I used was the Google Ngrams Viewer, which shows a small amount of data in visual format. Within the source of the page, though, the frequencies are shown and able to be collected. Without

this time-saving method, a researcher would have to download the entire dataset and try to develop tools to process it. Since it's over 2 terabytes, storage alone can be difficult, but searching it requires a great deal of processing power as well.

While web scraping is a powerful method for collecting data, many data repositories offer an API to make it easier to access the data. This method is often much more efficient to deploy, as APIs are designed to open data up for easy access. Typically, an API allows a researcher to send a request for specific data from a much larger set. The searching and processing is done from the repository server which is often very sophisticated and expensive. This reduces the need for such hardware from the researcher using the API, lowering costs for the research.

The case study used in this thesis highlights the value of such tools. Scraping the Google Ngrams Viewer prevented the purchase and deployment of a computer with a huge storage capacity and processing power. Not only that, it can be used by the public to do similar research. Without the Twitter Streaming API, it would have been very difficult to collect the tweets for the project. We could have deployed a scraping program, but the likelihood of missing tweets would have been very high increasing missingness.

Observations

Both methods have positives and negatives for their use. Both methods require programming acumen, for instance. If the targeted data repository offers an API that allows access to everything needed, that is almost always going to be the better method to use. Probably the largest negative to the API is that you are limited by what the API offers. Web

scraping, on the other hand, allows a developer to access nearly anything that can be viewed via a web browser.

Web scraping, the more versatile method for collecting data from the Internet, is much more difficult to deploy. Web sites don't always have organized methods for displaying data, making parsing the response difficult. Also, some data repositories and sites refuse to give permission to scrape their sites, even though the data is often public. APIs are designed to access the data from a site, so they are often more organized and the data is much more regular. Also, gaining permission is often much easier since they are intending the data to be accessed by other people.

A tool designed to scrape data from a site also has the complication that the tool will stop working if the site changes its layout. An API can be written in such a way that major changes to the site will not impact the use of the API. Sites that don't want people to harvest their information, even though it is public, can make scraping more difficult by changing their site periodically.

When it comes to creating and deploying tools that use these methods, probably the most important thing to do is document its creation. Comments in the code help the creator and anyone else looking at the code to see the intent of each section. Also, a tool that harnesses an API is much easier to develop and deploy in most cases. If an API is not available or is limited in its usefulness, then resort to web scraping.

Overview

Accessing large data sets is a valuable tactic to do research in today's scientific environment. The two methods described in this thesis offer tools to increase the efficiency

of collecting data from the Internet. Web scraping is versatile and powerful, but can be difficult to develop and deploy. Using an API is easier to deploy, but can be limiting to what can be accessed. Both are powerful tools to collect scientific data. The case study presented here demonstrates the power of these methods by measuring mentions of countries in two different data sources that can be difficult to access without these methods.

REFERENCES

Adamic, Lada A., and Eytan Adar. "Friends and neighbors on the web." Social networks 25.3 (2003): 211-230.

Aitamurto, Tanja, and Seth C. Lewis. "Open innovation in digital journalism: Examining the impact of Open APIs at four news organizations." new media & society 15.2 (2013): 314-331.

Alonso-Rorís, Víctor M., et al. "Information extraction in semantic, highly-structured, and semi-structured web sources." Polibits 49 (2014): 69-75.

Angela, M. O. "The precious and the precocious: Understanding cumulative disadvantage and cumulative advantage over the life course." The Gerontologist 36.2 (1996): 230-238.

Anholt, Simon. "Anholt nation brands index: how does the world see America?." Journal of Advertising Research 45.3 (2005): 296-304.

Anholt, Simon. *Places: Identity, image and reputation.* Springer, 2010.

Ashton, Kevin. "That 'internet of things' thing." RFiD Journal 22.7 (2009): 97-114.

Atzori, Luigi, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey." Computer networks 54.15 (2010): 2787-2805.

Bargh, John A., and Katelyn YA McKenna. "The Internet and social life." Annu. Rev. Psychol. 55 (2004): 573-590.

Barr, Juliana, et al. "Clinical practice guidelines for the management of pain, agitation, and delirium in adult patients in the intensive care unit." Critical care medicine 41.1 (2013): 263-306.

Beaudette, D. E., and A. T. O'Geen. "Soil-Web: an online soil survey for California, Arizona, and Nevada." Computers & Geosciences 35.10 (2009): 2119-2128.

Bernburg, Jön Gunnar, and Marvin D. Krohn. "Labeling, life chances, and adult crime: The direct and indirect effects of official intervention in adolescence on crime in early adulthood." Criminology 41.4 (2003): 1287-1318.

Binstock, Georgina, et al. "Influences on the knowledge and beliefs of ordinary people about developmental hierarchies." International journal of comparative sociology (2013): 0020715213506726.

Bollen, Johan, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market." Journal of Computational Science 2.1 (2011): 1-8.

Boulos, Maged NK, Inocencio Maramba, and Steve Wheeler. "Wikis, blogs and podcasts: a new generation of Web-based tools for virtual collaborative clinical practice and education." BMC medical education 6.1 (2006): 41.

Brysbaert, Marc, Emmanuel Keuleers, and Boris New. "Assessing the usefulness of Google Books' word frequencies for psycholinguistic research on word processing." Frontiers in Psychology 2 (2011).

Burton, Scott H., et al. ""Right time, right place" health communication on Twitter: value and accuracy of location information." Journal of medical Internet research 14.6 (2012).

Connelly, Roxanne, Vernon Gayle, and Paul S. Lambert. "A Review of occupation-based social classifications for social survey research." Methodological Innovations 9 (2016): 2059799116638003.

Cook, Colleen, Fred Heath, and Russel L. Thompson. "A meta-analysis of response rates in web-or internet-based surveys." Educational and psychological measurement 60.6 (2000): 821-836.

Crystal, Stephen, Dennis G. Shea, and Adriana M. Reyes. "Cumulative advantage, cumulative disadvantage, and evolving patterns of late-life inequality." The Gerontologist (2016): gnw056.

Dewan, Rajiv M., Marshall L. Freimer, and Yabing Jiang. "A temporary monopolist: Taking advantage of information transparency on the web." Journal of Management Information Systems 24.2 (2007): 167-194.

Dillman, Don A. *Mail and internet surveys: The tailored design method.* Vol. 2. New York: Wiley, 2000.

DiMaggio, Paul. "Adapting computational text analysis to social science (and vice versa)." Big Data & Society 2.2 (2015): 2053951715602908.

Dinnie, Keith. *Nation branding: Concepts, issues, practice, 2nd edition.* Routledge, 2016.

Dodds, Peter Sheridan, et al. "Temporal patterns of happiness and information in a global social network: Hedonometrics and Twitter." PloS one 6.12 (2011): e26752.

Evans, James A., and Jacob G. Foster. "Metaknowledge." Science (2011).

Fan, Ying. "Branding the nation: What is being branded?." Journal of vacation marketing 12.1 (2006): 5-14.

Fetscherin, Marc. "The determinants and measurement of a country brand: the country brand strength index." International Marketing Review 27.4 (2010): 466-479.

Field, Kenneth, and James O'Brien. "Cartoblography: Experiments in Using and Organising the Spatial Context of Micro-blogging." Transactions in GIS 14.s1 (2010): 5-23.

Gill, Stephen R., and David Law. "Global hegemony and the structural power of capital." International Studies Quarterly (1989): 475-499.

Glez-Peña, Daniel, et al. "Web scraping technologies in an API world." Briefings in bioinformatics 15.5 (2014): 788-797.

Golder, Scott A., and Michael W. Macy. "Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures." Science 333.6051 (2011): 1878-1881.

Goldkind, Lauri, and Lea Wolf. "A digital environment approach: Four technologies that will disrupt social work practice." Social work (2014): swu045.

Goldman, Lauren M. "Trending Now: The Use of Social Media Websites in Public Shaming Punishments." Am. Crim. L. Rev. 52 (2015): 415.

Groves, Robert M., and Lars Lyberg. "Total survey error: Past, present, and future." Public opinion quarterly 74.5 (2010): 849-879.

Hauser, Robert M., and John Robert Warren. "4. Socioeconomic Indexes for Occupations: A Review, Update, and Critique." Sociological methodology 27.1 (1997): 177-298.

Henrich, Joseph, and Francisco J. Gil-White. "The evolution of prestige: Freely conferred deference as a mechanism for enhancing the benefits of cultural transmission." Evolution and human behavior 22.3 (2001): 165-196.

Jansen, Bernard J., et al. "Twitter power: Tweets as electronic word of mouth." Journal of the American society for information science and technology 60.11 (2009): 2169-2188.

Java, Akshay, et al. "Why we twitter: understanding microblogging usage and communities." Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. ACM, 2007.

Kaplan, Andreas M., and Michael Haenlein. "Users of the world, unite! The challenges and opportunities of Social Media." Business horizons 53.1 (2010): 59-68.

Kavanaugh, Andrea L., et al. "Social media use by government: From the routine to the critical." Government Information Quarterly 29.4 (2012): 480-491.

Kutateladze, Besiki L., et al. "Cumulative disadvantage: Examining racial and ethnic disparity in prosecution and sentencing." Criminology 52.3 (2014): 514-551.

Lin, Henry C., et al. "Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions." Computer Aided Surgery11.5 (2006): 220-230.

Lin, Yuri, et al. "Syntactic annotations for the google books ngram corpus." Proceedings of the ACL 2012 system demonstrations. Association for Computational Linguistics, 2012.

Lopes, Giza, et al. "Labeling and cumulative disadvantage: The impact of formal police intervention on life chances and crime during emerging adulthood." Crime & Delinquency 58.3 (2012): 456-488.

Malhotra, Anshu, et al. "Studying user footprints in different online social networks." Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012). IEEE Computer Society, 2012.

Martin, Ingrid M., and Sevgin Eroglu. "Measuring a multi-dimensional construct: country image." Journal of business research 28.3 (1993): 191-210.

Marwick, Alice E. "I tweet honestly, I tweet passionately: Twitter users, context collapse, and the imagined audience." New media & society 13.1 (2011): 114-133.

Melegh, Attila, et al. "Perceptions of societal developmental hierarchies in Europe and beyond: a Bulgarian perspective." European sociological review29.3 (2013): 603-615.

Melegh, Attila. "Between global and local hierarchies: population management in the first half of the twentieth century." Demográfia English Edition 53.5 (2010): 51-77.

Merton, Robert K. "The Matthew effect in science." Science 159.3810 (1968): 56-63.

Michel, Jean-Baptiste, et al. "Quantitative analysis of culture using millions of digitized books." science 331.6014 (2011): 176-182.

Nakao, Keiko, and Judith Treas. Computing 1989 occupational prestige scores. publisher not identified, 1990.

Nakao, Keiko, and Judith Treas. The 1989 socioeconomic index of occupations: Construction from the 1989 occupational prestige scores. Chicago: National Opinion Research Center, 1992.

Oussalah, M., et al. "A software architecture for Twitter collection, search and geolocation services." Knowledge-Based Systems 37 (2013): 105-120.

Petersen, Alexander M., et al. "Languages cool as they expand: Allometric scaling and the decreasing need for new words." Scientific Reports 2 (2012).

Preis, Tobias, et al. "Quantifying the advantage of looking forward."Scientific reports 2 (2012).

Scanfeld, Daniel, Vanessa Scanfeld, and Elaine L. Larson. "Dissemination of health information through social networks: Twitter and antibiotics." American journal of infection control 38.3 (2010): 182-188.

Schau, Hope Jensen, and Mary C. Gilly. "We are what we post? Self-presentation in personal web space." Journal of consumer research 30.3 (2003): 385-404.

Schmidt, William C. "World-Wide Web survey research: Benefits, potential problems, and solutions." Behavior Research Methods, Instruments, & Computers 29.2 (1997): 274-279.

Seabrook, Jamie A., and William R. Avison. "Socioeconomic status and cumulative disadvantage processes across the life course: implications for health outcomes." Canadian Review of Sociology/Revue canadienne de sociologie 49.1 (2012): 50-68.

Signorini, Alessio, Alberto Maria Segre, and Philip M. Polgreen. "The use of Twitter to track levels of disease activity and public concern in the US during the influenza A H1N1 pandemic." PloS one 6.5 (2011): e19467.

Steinmetz, George. "The Sociology of Empires, Colonies, and Postcolonialism." Annual Review of Sociology 40 (2014): 77-103.

Taira, Koji. "Japan, an imminent hegemon?." The Annals of the American Academy of Political and Social Science (1991): 151-163.

Tat-Kei Ho, Alfred. "Reinventing local governments and the e-government initiative." Public administration review 62.4 (2002): 434-444.

Thornton, Arland, et al. "Knowledge and beliefs about national development and developmental hierarchies: The viewpoints of ordinary people in thirteen countries." Social science research 41.5 (2012): 1053-1068.

Treiman, Donald J. *Occupational prestige in comparative perspective*. Elsevier, 2013.

Wallerstein, Immanuel. *The capitalist world-economy. Vol. 2*. Cambridge University Press, 1979.

Whitmore, Andrew, Anurag Agarwal, and Li Da Xu. "The Internet of Things—A survey of topics and trends." Information Systems Frontiers 17.2 (2015): 261-274.

Yang, Y., L. T. Wilson, and J. Wang. "Development of an automated climatic data scraping, filtering and display system." Computers and Electronics in Agriculture 71.1 (2010): 77-87.

APPENDIX A

NGRAM CORPUS/LANGUAGE CHOICES

American English
British English
Chinese (simplified)
English
English Fiction
French
German
Hebrew
Italian
Russian
Spanish

APPENDIX B

TWITTER STREAMING API FIELDS

text
retweet_count
favorited
truncated
id_str
in_reply_to_screen_name
source
retweeted
created_at
in_reply_to_status_id_str
in_reply_to_user_id_str
lang
listed_count
verified
location
user_id_str
description
geo_enabled
user_created_at
statuses_count
followers_count
favourites_count
protected
user_url
name
time_zone
user_lang
utc_offset
friends_count
screen_name
country_code
country
place_type
full_name
place_name
place_id
place_lat
place_lon
lat
lon
expanded_url
url

APPENDIX C

COUNTRY LISTS

World War 1 countries:
United Kingdom
France
Russia
Italy
United States
Germany
Austria
Hungary
Bulgaria
Turkey

World War 2 countries:
United Kingdom
France
Australia
Canada
United States
New Zealand
India
Russia
China
Germany
Italy
Japan

G7 countries:
France
Germany
Italy
Japan
United Kingdom
United States
Canada

G8plus5 countries:
France
Germany
Italy
Japan
United Kingdom
United States
Canada
Russia
China
Brazil
India
Mexico
South Africa

G20 countries:
France
Germany
Italy
Japan
United Kingdom
United States
Canada
Russia
China
Brazil
India
Mexico
South Africa
Argentina
South Korea
Indonesia
Saudi Arabia
Australia

Northwestern European countries (25)

Australia
Austria
Belgium
Canada
Denmark
Estonia
Finland
France
Germany
Greenland
Guernsey
Iceland
Ireland
Latvia
Liechtenstein
Lithuania
Luxembourg
Monaco
Netherlands
New Zealand
Norway
Sweden
Switzerland
United Kingdom
United States

Eastern European countries (25)

Albania
Andorra
Belarus
Bulgaria
Croatia
Czech Republic
Gibraltar
Greece
Holy See
Hungary
Italy
Kosovo
Macedonia
Malta
Moldova
Montenegro
Poland
Portugal
Romania
Russia
Serbia
Slovakia
Slovenia
Spain
Ukraine

Latin America countries (34)

Argentina
Aruba
Bahamas
Barbados
Belize
Bermuda
Bolivia
Brazil
Chile
Colombia
Costa Rica
Cuba
Dominican Republic
Ecuador
El Salvador
French Guiana
Grenada
Guadeloupe
Guatemala
Guyana
Haiti
Honduras
Jamaica
Martinique
Mexico
Montserrat
Nicaragua
Panama
Paraguay
Peru
Puerto Rico
Suriname
Uruguay
Venezuela

Asian countries (58)

Afghanistan
Armenia
Azerbaijan
Bahrain
Bangladesh
Bhutan
Brunei
Cambodia
China
Cyprus
Fiji
French Polynesia
Guam
India
Indonesia
Iran
Iraq
Israel
Japan
Kazakhstan
Kiribati
North Korea
South Korea
Kuwait
Kyrgyzstan
Laos
Lebanon
Malaysia
Maldives
Micronesia
Mongolia
Myanmar
Nauru
Nepal
New Caledonia
Niue
Pakistan
Palau
Palestine
Papua New Guinea
Philippines
Pitcairn
Qatar
Saudi Arabia
Singapore
Sri Lanka
Syria
Taiwan
Tajikistan
Thailand
Tokelau
Tonga
Turkmenistan
Tuvalu
United Arab Emirates
Uzbekistan
Vanuatu
Viet Nam
Yemen

African countries (46)

Algeria
Angola
Benin
Botswana
Burkina Faso
Burundi
Cameroon
Central African Republic
Comoros
Ivory Coast
Djibouti
Egypt
Equatorial Guinea
Eritrea
Ethiopia
Gabon
Gambia
Ghana
Kenya
Lesotho
Liberia
Libya
Madagascar
Malawi
Mali
Mauritania
Mauritius
Mayotte
Morocco
Mozambique
Namibia
Nigeria
Rwanda
Senegal
Seychelles
Sierra Leone
Somalia
South Africa
Sudan
Swaziland
Tanzania
Tunisia
Uganda
Western Sahara
Zambia
Zimbabwe

Final list of 189 country names used

Afghanistan
Albania
Algeria
Andorra
Angola
Argentina
Armenia
Aruba
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh
Barbados
Belarus
Belgium
Belize
Benin
Bermuda
Bhutan
Bolivia
Botswana
Brazil
Brunei
Bulgaria
Burkina Faso
Burundi
Cambodia
Cameroon
Canada
Central African Republic
Chile
China
Colombia
Comoros
Costa Rica
Croatia
Cuba
Cyprus
Czech Republic
Denmark
Djibouti
Dominican Republic
Ecuador
Egypt
El Salvador
Equatorial Guinea
Eritrea
Estonia
Ethiopia
Fiji
Finland
France
French Guiana
French Polynesia
Gabon
Gambia
Germany
Ghana
Gibraltar
Greece
Greenland
Grenada
Guadeloupe
Guam
Guatemala
Guernsey
Guyana
Haiti
Holy See
Honduras
Hungary
Iceland
India
Indonesia
Iran
Iraq
Ireland
Israel
Italy
Ivory Coast
Jamaica
Japan
Kazakhstan
Kenya
Kiribati
Kosovo
Kuwait
Kyrgyzstan
Laos
Latvia
Lebanon
Lesotho
Liberia
Libya
Liechtenstein
Lithuania
Luxembourg
Macedonia
Madagascar
Malawi
Malaysia
Maldives
Mali
Malta
Martinique
Mauritania
Mauritius
Mayotte
Mexico
Micronesia
Moldova
Monaco
Mongolia
Montenegro
Montserrat
Morocco
Mozambique
Myanmar
Namibia
Nauru
Nepal
Netherlands
New Caledonia
New Zealand
Nicaragua
Nigeria
Niue
North Korea
Norway
Pakistan
Palau
Palestine
Panama

Papua New Guinea
Paraguay
Peru
Philippines
Pitcairn
Poland
Portugal
Puerto Rico
Qatar
Romania
Russia
Rwanda
Saudi Arabia
Senegal
Serbia
Seychelles
Sierra Leone
Singapore
Slovakia
Slovenia
Somalia
South Africa
South Korea
Spain
Sri Lanka
Sudan
Suriname
Swaziland
Sweden
Switzerland
Syria
Taiwan
Tajikistan
Tanzania
Thailand
Tokelau
Tonga
Tunisia
Turkmenistan
Tuvalu
Uganda
Ukraine
United Arab Emirates
United Kingdom
United States
Uruguay
Uzbekistan
Vanuatu
Venezuela
Viet Nam
Western Sahara
Yemen
Zambia
Zimbabwe

List of countries used in regression tables

Angola
Argentina
Australia
Austria
Belgium
Brazil
Canada
Chile
China
Colombia
Czech Republic
Denmark
Ecuador
Egypt
Estonia
Finland
France
Germany
Hungary
India
Indonesia
Iran
Ireland
Italy
Japan
Kenya
Lituania
Malaysia
Mexico
Netherlands
New Zealand
Nigeria
Norway
Peru
Poland
Romania
Russia
Saudi Arabia
Singapore
South Africa
South Korea
Spain
Sweden
Switzerland
Tailand
United Arab Emirates
United Kingdom
United States

APPENDIX D

R CODE FOR COLLECTING TWITTER DATA

```
library(ROAuth)
library(streamR)

# 1a) First we set the Twitter parameters (can skip to 1b)
requestURL <- "https://api.twitter.com/oauth/request_token"
accessURL <- "https://api.twitter.com/oauth/access_token"
authURL <- "https://api.twitter.com/oauth/authorize"

# THESE NEXT TWO LINES ARE ACCOUNT SPECIFIC!!
consumerKey <- "KrjtebqOJgTgaFS4zxgSdC2E8"
consumerSecret <- "L02T4ahCludE5w7YDNvTfgSkv1IhRur6Q4aIhF1roNysYo85CO"

# Put the above together
my_oauth <- OAuthFactory$new(consumerKey=consumerKey,
                             consumerSecret=consumerSecret,
                             requestURL=requestURL,
                             accessURL=accessURL, authURL=authURL)

# Do the actual connection. You will have to agree
# to share this information on the Twitter site and
# enter in the responding code here.
my_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem",
                                        package = "RCurl"))

# Once this is done, you can save the credentials to use again
save(my_oauth, file="credentials.RData")

# 1b) You can SKIP all the above and just load what you used before
load(file="credentials.RData")

# 2) Setting the parameters
# First, we set the list of countries
countryList <- read.csv("countries.txt")

# Next, we set where the data files will be saved
mainDir <- getwd() # Can use setwd("/path/to/R/files")
dataDir <- "Data" # The folder with the saved data files
procDir <- "Processed" # The folder that will hold processed files
filePre <- "countries temp " # The prefix of the files we're processing

# Set collection time in HOURS
colTime <- 240 # Set to whatever you want. Ex: 240 => 10 days
curTime <- Sys.time() # Get the current time
expTime <- curTime + (colTime * 3600) # Set to current + collection time
loopCount <- 1 # Initialize the count

while(Sys.time() < expTime ) # Loop until the time has expired
{
  # Name the file and put it in the right place
```

```
  fileName <- file.path(mainDir,dataDir,
                        paste(filePre, loopCount, ".txt", sep = ""))
  # Increment the loop count
  loopCount <- loopCount + 1

  # Collect the data in 6 minute increments
  # The stream has to reconnect each time, so keep 30 secs or longer.
  # The stream should be held open long enough to collect at least a few
  # tweets, but short enough to avoid a large file size. You will need
  # to experiment to get an appropriate file size, but try for 1500 tweets
  filterStream( file=fileName,
                track=countryList, timeout=360, oauth=my_oauth )
}
```

APPENDIX E

R CODE FOR PROCESSING TWITTER DATA

```
library(streamR)
library(foreign)

# This is the PROCESSING of the already collected data

# 1) Set some parameters

mainDir <- getwd() # Can us setwd("/path/to/R/files")
setwd("D:/Twitter")
dataDir <- "Data/Data 8" # The folder with the saved data files
procDir <- "Processed/8" # The folder that will hold processed files
filePre <- "countries 08 " # The prefix of the files we're processing

# The list of countries we're monitoring
countryList <- read.csv("countries.txt")

# Find the data file list
fList <- list.files(path=file.path(mainDir, dataDir), pattern=filePre)

# Ensure folders are made for each country in the list
for (i in 1:nrow(countryList)) {
  # Step through the countries and make folders
  # If they already exist, we'd get a warning, so turn them off
  dir.create(file.path(mainDir, procDir, countryList[i,]), showWarnings =
FALSE)
}


# 2) Processing the data into specific countries

# Get file, parse file, go through each country saving file
for (j in 1:length(fList)) { #length(fList)
  # If there is data, we first parse the tweets in the selected file
  if (file.info(file.path(mainDir,dataDir,fList[j]))$size > 700) {
    tempList <- parseTweets(file.path(mainDir,dataDir,fList[j]))

    # Next we step through the countries to create files

#    rowNums <- grep("Afghanistan",tempList$text)
#    tempCList <- tempList[rowNums,]
#    cfile <- gsub("countries", "Afghanistan", fList[j])
#    write.csv(tempCList, file.path(mainDir, procDir, "Afghanistan", cfile))
#  }
#}

    for (i in 1:nrow(countryList)) {
      # We get the subset of tweets containing the country name
      rowNums <- grep(countryList[i,],tempList$text)
      tempCList <- tempList[rowNums,]
```

```
      # Finally, we write the list using the same name
      # For naming, we replace 'countries' with the country name
      cfile <- gsub("countries", countryList[i,],fList[j])
      # location: main folder/processing folder/country name
      write.csv(tempCList, file.path(mainDir, procDir, countryList[i,],
cfile))
    } # Finish stepping through one file, all countries
  } # End IF - one tweet is about 4000, so 700 is a very safe number for
error messages
} # Finish stepping through all files
```

APPENDIX F

CORPUSEXTRACT TOOL PHP CODE

```
<html>
<body>

<?php
function cleanse_input($input)
{
  $comRep = array(" , "," ,",", ");
  $plusRep = array(" + "," +","+ ");
  while (strpos($input,'  ')>0) $input = str_replace('  ',' ',$input); //Turn
all spaces into a single space
  while ((strpos($input,', ')>0) || (strpos($input,' ,')>0)) $input =
str_replace($comRep,',',$input); //Turn all comma spaces into comma
  while ((strpos($input,'+ ')>0) || (strpos($input,' +')>0)) $input =
str_replace($plusRep,'+',$input); //Turn all plus spaces into plus
  $input = str_replace ('+','%2B', $input); // Turn '+' into '%2B'
  $input = str_replace (' ','+', $input); // Turn spaces into '+'
  $input = preg_replace('/\s+/', ',', $input); //Turn all white space into
commas
  while (strpos($input,",,")>0) $input = str_replace (",,",",",$input); //
Remove extraneous commas
  $input = preg_replace('(^\+)', '', $input); //Turn all leading '+' into
nothing
  $input = preg_replace("/(\+)(\,)?$/", '', $input); //Turn all ending '+'
into nothing
  $input = stripslashes($input); // Remove slashes
  return $input;
}

function cleanse_output($input)  // Output needs to be returned back into a
viewable form
{
  $input = str_replace ('+',' ', $input); // Turn '+' into space
  $input = str_replace ('%2B','+', $input); // Turn '%2B' into '+'
  return $input;
}

function getNGramData($url,$yrstart,$yrend)
{

  // create curl resource
  $ch = curl_init();

  // set url
  curl_setopt($ch, CURLOPT_URL, $url);

  //set options like returning the transfer as a string
  curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
  curl_setopt($ch, CURLOPT_FAILONERROR, 1);
```

```
  // $output contains the output string
  $output = curl_exec($ch);

  // Split the ngrams
  $processOutput = substr($output, strpos($output, "var data =
"),strpos($output, "}];"));
  $outputArray = explode("\"ngram\":",$processOutput);
  array_shift($outputArray); // Shifted because the first item isn't an ngram

  // Now that each ngram is in an index, let's parse.
  // List of noise that will need to be removed from the input:

$search_list=array(":[","],","]},{","]}];if(data.length>0){ngrams.drawD3Chart
(data,$yrstart,$yrend,1.0,");

  // Initialize the arrays
  $arrayCount = 0;
  $finalArray = array();

  // Step through each ngram and build an array
  foreach ($outputArray as $csvEntry) {
    $splitEntry = explode("\"",$csvEntry);
    $dataIndex = 0;

    // Step through each area looking for the label and data
    foreach($splitEntry as $field) {
      if(strpos($field,": [")!==false) break;
      $dataIndex++;
    }

    // Clean the data area
    $splitEntry[$dataIndex] = preg_replace('/\s+/', '',
$splitEntry[$dataIndex]); //Remove all white space
    $splitEntry[$dataIndex] =
str_replace($search_list,"",$splitEntry[$dataIndex]); //Remove all noise

    $dataValues = explode(",",$splitEntry[$dataIndex]); // Now that we've
found the data, split each piece out
    $finalArray[$arrayCount][0] = $splitEntry[1]; // Make the first entry the
label
    foreach ($dataValues as $piece) $finalArray[$arrayCount][] = $piece; //
Put each piece into the array

    $arrayCount++; // Go to next array
  }
  // close curl resource to free up system resources
  curl_close($ch);
  return ($finalArray);
}

function smoothArray ($baseArray, $smoothNum) {
  $returnArray = array ();
  foreach($baseArray as $key => $val) {
    // Initialize numerator and denominator
    $numer = 0;
    $denom = 0;
```

```
    for($i=(0-$smoothNum);$i<=$smoothNum;$i++) {
      if(array_key_exists(($key+$i),$baseArray)) {
        $numer=$numer+$baseArray[$key+$i];
        $denom++;
      }
    }
    $returnArray[$key]=($numer/$denom);
  }
  return($returnArray);
}

$corpusArray = array (
  17  =>  "googlebooks-eng-us-all-totalcounts-20120701.txt",
  18  =>  "googlebooks-eng-gb-all-totalcounts-20120701.txt",
  23  =>  "googlebooks-chi-sim-all-totalcounts-20120701.txt",
  15  =>  "googlebooks-eng-all-totalcounts-20120701.txt",
  16  =>  "googlebooks-eng-fiction-all-totalcounts-20120701.txt",
  19  =>  "googlebooks-fre-all-totalcounts-20120701.txt",
  20  =>  "googlebooks-ger-all-totalcounts-20120701.txt",
  24  =>  "googlebooks-heb-all-totalcounts-20120701.txt",
  22  =>  "googlebooks-ita-all-totalcounts-20120701.txt",
  25  =>  "googlebooks-rus-all-totalcounts-20120701.txt",
  21  =>  "googlebooks-spa-all-totalcounts-20120701.txt",
  5  =>  "googlebooks-eng-us-all-totalcounts-20090715.txt",
  6  =>  "googlebooks-eng-gb-all-totalcounts-20090715.txt",
  11  =>  "googlebooks-chi-sim-all-totalcounts-20120701.txt",
  0  =>  "googlebooks-eng-all-totalcounts-20090715.txt",
  4  =>  "googlebooks-eng-fiction-all-totalcounts-20090715.txt",
  1  =>  "googlebooks-eng-1M-totalcounts-20090715.txt",
  7  =>  "googlebooks-fre-all-totalcounts-20090715.txt",
  8  =>  "googlebooks-ger-all-totalcounts-20090715.txt",
  9  =>  "googlebooks-heb-all-totalcounts-20090715.txt",
  12  =>  "googlebooks-rus-all-totalcounts-20090715.txt",
  10  =>  "googlebooks-spa-all-totalcounts-20090715.txt"
);

if (isset($_GET)) { // If there is input, process it
  // First we clean it
  $content = (isset($_GET['content']) ? cleanse_input($_GET['content']) :
"");
  $yrstart = (isset($_GET['yrstart']) ? cleanse_input($_GET['yrstart']) :
1500);
  $yrend = (isset($_GET['yrend']) ? cleanse_input($_GET['yrend']) : 2008);
  $corpus = (isset($_GET['corpus']) ? cleanse_input($_GET['corpus']) : 15);
  $smoothing = (isset($_GET['smoothing']) ? cleanse_input($_GET['smoothing'])
: 3);
  $case = (isset($_GET['case']) ? "&case_insensitive=on" : "");

  // Test variables for range, and reset if out of limits
  if ($yrstart < 1500) $yrstart = 1500;
  if ($yrstart > 2008) $yrstart = 2008;
  if ($yrend < 1500) $yrend = 1500;
  if ($yrend > 2008) $yrend = 2008;
  if ($corpus < 0) $corpus = 0;
  if ($corpus > 25) $corpus = 25;
  if ($smoothing < 0) $smoothing = 0;
  if ($smoothing > 50) $smoothing = 50;
```

```
  // Parse content into 12 search chunks (Since Google Ngrams only runs on 12
ngram chunks)
  $contentArray = explode(",",$content); // Temporarily hold each search item
  $newContentArray = array_chunk($contentArray, 12); // Break into 12 piece
searches
  $contentArray = array(); // Reset this array
  foreach($newContentArray as $dataChunk) { // create a new content array
with the comma separated pieces
    $contentArray[] = implode(",",$dataChunk);
  }
} // End if (for testing the GET contents)

// Now we design the HTML form
?>

<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
method="get">
<table>
  <tr>
    <td>Content:</td>
    <td><textarea rows="4" cols="50" name="content"><?php echo
cleanse_output($content); ?></textarea></td>
  </tr>
  <tr>
    <td></td><td><input type="checkbox" name="case"<?php if($case!="") echo "
checked=\"checked\"";?>>Case insensitive</td>
  </tr>
  <tr>
    <td>Year Start:</td>
    <td><input type="text" name="yrstart" value="<?php echo $yrstart; ?>">
*1500-2008; blank = 1500</td>
  </tr>
  <tr>
    <td>Year End:</td>
    <td><input type="text" name="yrend" value="<?php echo $yrend; ?>"> *1500-
2008; blank = 2008</td>
  </tr>
  <tr>
    <td>Corpus:</td>
    <td><select name="corpus">
          <option value="17"<?php if ($corpus=="17") echo " selected";?>>
American English</option>
          <option value="18"<?php if ($corpus=="18") echo " selected";?>>
British English</option>
          <option value="23"<?php if ($corpus=="23") echo " selected";?>>
Chinese (simplified)</option>
          <option value="15"<?php if ($corpus=="15") echo " selected";?>>
English</option>
          <option value="16"<?php if ($corpus=="16") echo " selected";?>>
English Fiction</option>
          <option value="19"<?php if ($corpus=="19") echo " selected";?>>
French</option>
          <option value="20"<?php if ($corpus=="20") echo " selected";?>>
German</option>
          <option value="24"<?php if ($corpus=="24") echo " selected";?>>
Hebrew</option>
```

```
          <option value="22"<?php if ($corpus=="22") echo " selected";?>>
Italian</option>
          <option value="25"<?php if ($corpus=="25") echo " selected";?>>
Russian</option>
          <option value="21"<?php if ($corpus=="21") echo " selected";?>>
Spanish</option>
        </select>
    </td>
  </tr>
  <tr>
    <td>Smoothing:</td>
    <td><input type="text" name="smoothing" value="<?php echo $smoothing;
?>">*0-50; blank=3</td>
  </tr>
</table>
<input type="submit">
</form>
<p>
<a href="https://books.google.com/ngrams/info">Need help with Google Ngram
Viewer?</a>
<br>
<?php
if(isset($_GET['content'])) {
  // Get the total counts per year
  // Folder containing the files
  $location = $_SERVER['DOCUMENT_ROOT'] . "/www1/repository/ngramcounts/";
  // Initialize the array of year:total
  $yeartotal = array ();
  $row = 1;
  if (($handle = fopen($location.$corpusArray[$corpus], "r")) !== FALSE) {
    while (($data = fgetcsv($handle, 0, ",")) !== FALSE) {
      $num = count($data);
      for ($c=0; $c < $num; $c++) {
        if (strpos($data[$c],"\t")!= FALSE) {
          $tempkey=substr($data[$c],strpos($data[$c],"\t"));
          $yeartotal[(int)$tempkey] = $data[$c+1];
        }
      }
    }
    $row++;
  }
  for ($i=$yrstart;$i<$yrend;$i++) if (!(array_key_exists($i,$yeartotal)))
$yeartotal[$i] = 0;
  $yearSmooth = smoothArray($yeartotal,$smoothing);
  fclose($handle);

  // Retrieve data and build the array
  $step = 0; // initialize the steps for the links
  echo "Click here to visit this search on Google (shown in blocks of 12 as
per the Google limit):";
  foreach($contentArray as $contentItem) {
    $url =
"http://books.google.com/ngrams/graph?content=$contentItem$case&year_start=$y
rstart&year_end=$yrend&corpus=$corpus&smoothing=$smoothing";
    foreach (getNGramData($url,$yrstart,$yrend) as $ngram) $displayArray[] =
$ngram; // Does the work
    echo " <a href=\"$url\"> "; // Outputs the links to Google
```

```
    echo $step+1 . "</a>";
    $step++;
  }

  // Begin table display
  if (!(isset($displayArray))) echo "<br><strong>No results found.</strong>";
  else {
    echo "<br><br>Table output:<br>\n";
    echo "<table>\n";
    $maxArray = max(array_map('count',$displayArray));
    $lenArray = count($displayArray);
    for ($i=0; $i < $maxArray; $i++) {
      echo "  <tr>\n";
      for ($j=0; $j <= ($lenArray+1); $j++) {
        if ($j==0){
          if ($i==0) {
            echo "    <td>Year</td>\n";
          }
          else {
            echo "    <td>" . ($yrstart+$i-1) . "</td>\n";
          }
        }
        if ($j==$lenArray) {
          if ($i==0) {
            echo "    <td nowrap>Actual Total</td>\n";
          }
          else {
            echo "    <td>" . $yeartotal[$yrstart+$i-1] . "</td>\n";
          }
        }
        elseif($j==$lenArray+1) {
          if ($i==0) {
            echo "    <td nowrap>Smooth Total</td>\n";
          }
          else {
            echo "    <td>" . $yearSmooth[$yrstart+$i-1] . "</td>\n";
          }
        }
        else {
          if ($i>0) {
            $itemCount = round($yeartotal[$yrstart+$i-
1]*$displayArray[$j][$i]);
            echo "    <td nowrap>$itemCount</td>\n";
          }
          else echo "    <td nowrap>{$displayArray[$j][$i]}</td>\n";

        }
      }
      echo "  </tr>\n";
    }
    echo "\n</table>\n\n";

    // Begin CSV display and building CSV file
    echo "<br><br>CSV output:<br>\n";
    echo "<table>\n";
    $maxArray = max(array_map('count',$displayArray));
    $lenArray = count($displayArray);
```

```
    for ($i=0; $i < $maxArray; $i++) {
      echo "  <tr>\n    <td nowrap>";
      for ($j=0; $j <= $lenArray+1; $j++) {
        if ($j==0){
          if ($i==0) {
            echo "Year";
          }
          else {
            echo ($yrstart+$i-1);
          }
        }
        if ($j==$lenArray) {
          if ($i==0) {
            echo ",Actual Total";
          }
          else {
            echo "," . $yeartotal[$yrstart+$i-1];
          }
        }
        elseif ($j==$lenArray+1) {
          if ($i==0) {
            echo ",Smooth Total";
          }
          else {
            echo "," . $yearSmooth[$yrstart+$i-1];
          }
        }
        else {
          if ($i>0) {
            $itemCount = round($yeartotal[$yrstart+$i-
1]*$displayArray[$j][$i]);
            echo ",$itemCount";
          }
          else echo "," . $displayArray[$j][$i];

        }
      }
      echo "</td>\n  </tr>\n";
    }
    echo "\n</table>\n\n";
  }
}

?>

<p>
  If you are experiencing any unexpected results, please click on the link of
the search that
  is responding unexpectedly and copy the URL and send it to:
  <a href="mailto:jwillers@shawndorius.com">jwillers@shawndorius.com</a>.<br>
  Alternatively, go to <a href="https://books.google.com/ngrams">Ngram
Viewer</a>, enter your search,
  make sure it performs as expected, and email the working Ngram URL.<br><br>
  A NOTE ON CORPUS USE: Only the 2012 corpus for a language is accessible at
this time. The 2009 corpora
  have been deprecated.
</p>
```

```
</body>
</html>
```