Chapter 1

What is Data Science?

> The purpose of computing is insight, not numbers.
>
> – Richard W. Hamming

What is data science? Like any emerging field, it hasn't been completely defined yet, but you know enough about it to be interested or else you wouldn't be reading this book.

I think of data science as lying at the intersection of computer science, statistics, and substantive application domains. From computer science comes machine learning and high-performance computing technologies for dealing with scale. From statistics comes a long tradition of exploratory data analysis, significance testing, and visualization. From application domains in business and the sciences comes challenges worthy of battle, and evaluation standards to assess when they have been adequately conquered.

But these are all well-established fields. Why data science, and why now? I see three reasons for this sudden burst of activity:

- New technology makes it possible to capture, annotate, and store vast amounts of social media, logging, and sensor data. After you have amassed all this data, you begin to wonder what you can do with it.

- Computing advances make it possible to analyze data in novel ways and at ever increasing scales. Cloud computing architectures give even the little guy access to vast power when they need it. New approaches to machine learning have lead to amazing advances in longstanding problems, like computer vision and natural language processing.

- Prominent technology companies (like Google and Facebook) and quantitative hedge funds (like Renaissance Technologies and TwoSigma) have proven the power of modern data analytics. Success stories applying data to such diverse areas as sports management (*Moneyball* [Lew04]) and election forecasting (Nate Silver [Sil12]) have served as role models to bring data science to a large popular audience.

S.S. Skiena, *The Data Science Design Manual,*
Texts in Computer Science, https://doi.org/10.1007/978-3-319-55444-0_1

This introductory chapter has three missions. First, I will try to explain how good data scientists think, and how this differs from the mindset of traditional programmers and software developers. Second, we will look at data sets in terms of the potential for what they can be used for, and learn to ask the broader questions they are capable of answering. Finally, I introduce a collection of data analysis challenges that will be used throughout this book as motivating examples.

1.1 Computer Science, Data Science, and Real Science

Computer scientists, by nature, don't respect data. They have traditionally been taught that the algorithm was the thing, and that data was just meat to be passed through a sausage grinder.

So to qualify as an effective data scientist, you must first learn to think like a real scientist. Real scientists strive to understand the natural world, which is a complicated and messy place. By contrast, computer scientists tend to build their own clean and organized virtual worlds and live comfortably within them. Scientists obsess about discovering things, while computer scientists invent rather than discover.

People's mindsets strongly color how they think and act, causing misunderstandings when we try to communicate outside our tribes. So fundamental are these biases that we are often unaware we have them. Examples of the cultural differences between computer science and real science include:

- *Data vs. method centrism*: Scientists are data driven, while computer scientists are algorithm driven. Real scientists spend enormous amounts of effort collecting data to answer their question of interest. They invent fancy measuring devices, stay up all night tending to experiments, and devote most of their thinking to how to get the data they need.

  By contrast, computer scientists obsess about methods: which algorithm is better than which other algorithm, which programming language is best for a job, which program is better than which other program. The details of the data set they are working on seem comparably unexciting.

- *Concern about results*: Real scientists care about answers. They analyze data to discover something about how the world works. Good scientists care about whether the results make sense, because they care about what the answers mean.

  By contrast, bad computer scientists worry about producing plausible-looking numbers. As soon as the numbers stop looking grossly wrong, they are presumed to be right. This is because they are personally less invested in what can be learned from a computation, as opposed to getting it done quickly and efficiently.

- *Robustness*: Real scientists are comfortable with the idea that data has errors. In general, computer scientists are not. Scientists think a lot about possible sources of bias or error in their data, and how these possible problems can effect the conclusions derived from them. Good programmers use strong data-typing and parsing methodologies to guard against formatting errors, but the concerns here are different.

  Becoming aware that data can have errors is empowering. Computer scientists chant “garbage in, garbage out” as a defensive mantra to ward off criticism, a way to say *that’s not my job*. Real scientists get close enough to their data to smell it, giving it the sniff test to decide whether it is likely to be garbage.

- *Precision*: Nothing is ever completely true or false in science, while *everything* is either true or false in computer science or mathematics.

  Generally speaking, computer scientists are happy printing floating point numbers to as many digits as possible: $8/13 = 0.61538461538$. Real scientists will use only two significant digits: $8/13 \approx 0.62$. Computer scientists care what a number is, while real scientists care what it means.

Aspiring data scientists must learn to think like real scientists. Your job is going to be to turn numbers into insight. It is important to understand the *why* as much as the *how*.

To be fair, it benefits real scientists to think like data scientists as well. New experimental technologies enable measuring systems on vastly greater scale than ever possible before, through technologies like full-genome sequencing in biology and full-sky telescope surveys in astronomy. With new breadth of view comes new levels of vision.

Traditional *hypothesis-driven* science was based on asking specific questions of the world and then generating the specific data needed to confirm or deny it. This is now augmented by *data-driven* science, which instead focuses on generating data on a previously unheard of scale or resolution, in the belief that new discoveries will come as soon as one is able to look at it. Both ways of thinking will be important to us:

- Given a problem, what available data will help us answer it?
- Given a data set, what interesting problems can we apply it to?

There is another way to capture this basic distinction between software engineering and data science. It is that software developers are hired to build systems, while data scientists are hired to produce insights.

This may be a point of contention for some developers. There exist an important class of engineers who wrangle the massive distributed infrastructures necessary to store and analyze, say, financial transaction or social media data on a full Facebook or Twitter-level of scale. Indeed, I will devote Chapter 12 to the distinctive challenges of big data infrastructures. These engineers are building tools and systems to support data science, even though they may not personally mine the data they wrangle. Do they qualify as data scientists?

This is a fair question, one I will finesse a bit so as to maximize the potential readership of this book. But I do believe that the better such engineers understand the full data analysis pipeline, the more likely they will be able to build powerful tools capable of providing important insights. A major goal of this book is providing big data engineers with the intellectual tools to think like big data scientists.

1.2 Asking Interesting Questions from Data

Good data scientists develop an inherent curiosity about the world around them, particularly in the associated domains and applications they are working on. They enjoy talking shop with the people whose data they work with. They ask them questions: What is the coolest thing you have learned about this field? Why did you get interested in it? What do you hope to learn by analyzing your data set? Data scientists always ask questions.

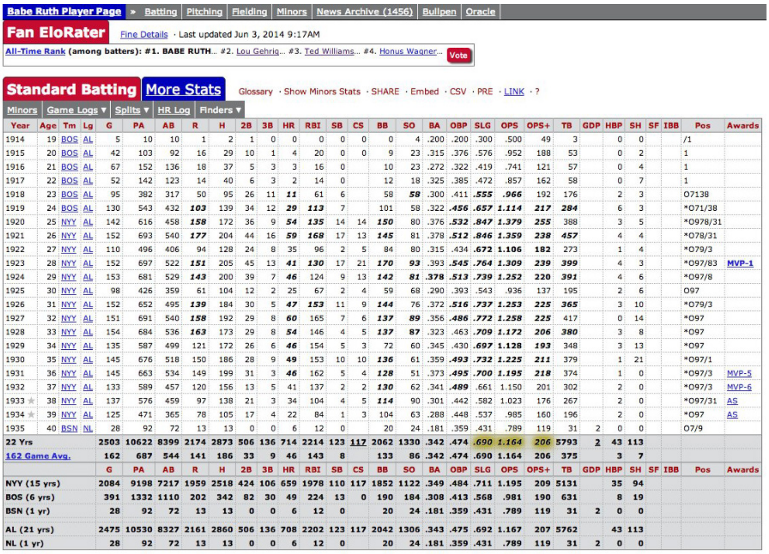
Good data scientists have wide-ranging interests. They read the newspaper every day to get a broader perspective on what is exciting. They understand that the world is an interesting place. Knowing a little something about everything equips them to play in other people's backyards. They are brave enough to get out of their comfort zones a bit, and driven to learn more once they get there.

Software developers are not really encouraged to ask questions, but data scientists are. We ask questions like:

- What things might you be able to learn from a given data set?
- What do you/your people really want to know about the world?
- What will it mean to you once you find out?

Computer scientists traditionally do not really appreciate data. Think about the way algorithm performance is experimentally measured. Usually the program is run on "random data" to see how long it takes. They rarely even look at the results of the computation, except to verify that it is correct and efficient. Since the "data" is meaningless, the results cannot be important. In contrast, real data sets are a scarce resource, which required hard work and imagination to obtain.

Becoming a data scientist requires learning to ask questions about data, so let's practice. Each of the subsections below will introduce an interesting data set. After you understand what kind of information is available, try to come up with, say, five interesting questions you might explore/answer with access to this data set.



Babe Ruth Player Page » Batting | Pitching | Fielding | Minors | News Archive (1456) | Bullpen | Oracle
Fan EloRater
Vote
Standard Batting
More Stats
Glossary · Show Minors Stats · SHARE · Embed · CSV · PRE · LINK · ?
Minors | Game Logs ▾ | Splits ▾ | HR Log | Finders ▾

Fine Details · Last updated Jun 3, 2014 9:17AM

All-Time Rank (among batters): #1. BABE RUTH... #2. Lou Gehrig... #3. Ted Williams... #4. Honus Wagner...

Year	Age	Tm	Lg	G	PA	AB	R	H	2B	3B	HR	RBI	SB	CS	BB	SO	BA	OBP	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	Pos	Awards
1914	19	BOS	AL	5	10	10	1	2	1	0	0	0	0	0	0	4	.200	.200	.300	.500	49	3		0	0			/1	
1915	20	BOS	AL	42	103	92	16	29	10	1	4	20	0	0	9	23	.315	.376	.576	.952	188	53		0	2			1	
1916	21	BOS	AL	67	152	136	18	37	5	3	3	16	0		10	23	.272	.322	.419	.741	121	57		0	4			1	
1917	22	BOS	AL	52	142	123	14	40	6	3	2	14	0		12	18	.325	.385	.472	.857	162	58		0	7			1	
1918	23	BOS	AL	95	382	317	50	95	26	11	11	61	6		58	58	.300	.411	.555	.966	192	176		2	3			O7138	
1919	24	BOS	AL	130	543	432	103	139	34	12	29	113	7		101	58	.322	.456	.657	1.114	217	284		6	3			*O71/38	
1920	25	NYY	AL	142	616	458	158	172	36	9	54	135	14	14	150	80	.376	.532	.847	1.379	255	388		3	5			*O978/31	
1921	26	NYY	AL	152	693	540	177	204	44	16	59	168	17	13	145	81	.378	.512	.846	1.359	238	457		4	4			*O78/31	
1922	27	NYY	AL	110	496	406	94	128	24	8	35	96	2	5	84	80	.315	.434	.672	1.106	182	273		1	4			*O79/3	
1923	28	NYY	AL	152	697	522	151	205	45	13	41	130	17	21	170	93	.393	.545	.764	1.309	239	399		4	3			*O97/83	MVP-1
1924	29	NYY	AL	153	681	529	143	200	39	7	46	124	9	13	142	81	.378	.513	.739	1.252	220	391		4	6			*O97/8	
1925	30	NYY	AL	98	426	359	61	104	12	2	25	67	2	4	59	68	.290	.393	.543	.936	137	195		2	6			O97	
1926	31	NYY	AL	152	652	495	139	184	30	5	47	153	11	9	144	76	.372	.516	.737	1.253	225	365		3	10			*O79/3	
1927	32	NYY	AL	151	691	540	158	192	29	8	60	165	7	6	137	89	.356	.486	.772	1.258	225	417		0	14			*O97	
1928	33	NYY	AL	154	684	536	163	173	29	8	54	146	4	5	137	87	.323	.463	.709	1.172	206	380		3	8			*O97	
1929	34	NYY	AL	135	587	499	121	172	26	6	46	154	5	3	72	60	.345	.430	.697	1.128	193	348		3	13			*O97	
1930	35	NYY	AL	145	676	518	150	186	28	9	49	153	10	10	136	61	.359	.493	.732	1.225	211	379		1	21			*O97/1	
1931	36	NYY	AL	145	663	534	149	199	31	3	46	162	5	4	128	51	.373	.495	.700	1.195	218	374		1	0			*O97/3	MVP-5
1932	37	NYY	AL	133	589	457	120	156	13	5	41	137	2	2	130	62	.341	.489	.661	1.150	201	302		2	0			*O97/3	MVP-6
1933 ★	38	NYY	AL	137	576	459	97	138	21	3	34	104	4	5	114	90	.301	.442	.582	1.023	176	267		2	0			*O97/31	AS
1934 ★	39	NYY	AL	125	471	365	78	105	17	4	22	84	1	3	104	63	.288	.448	.537	.985	160	196		2	0			*O97	AS
1935	40	BSN	NL	28	92	72	13	13	0	0	6	12	0		20	24	.181	.359	.431	.789	119	31	2	0	0			O7/9	
22 Yrs				2503	10622	8399	2174	2873	506	136	714	2214	123	117	2062	1330	.342	.474	.690	1.164	206	5793	2	43	113				
162 Game Avg.				162	687	544	141	186	33	9	46	143	8		133	86	.342	.474	.690	1.164	206	375		3	7				
				G	PA	AB	R	H	2B	3B	HR	RBI	SB	CS	BB	SO	BA	OBP	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	Pos	Awards
NYY (15 yrs)				2084	9198	7217	1959	2518	424	106	659	1978	110	117	1852	1122	.349	.484	.711	1.195	209	5131		35	94				
BOS (6 yrs)				391	1332	1110	202	342	82	30	49	224	13	0	190	184	.308	.413	.568	.981	190	631		8	19				
BSN (1 yr)				28	92	72	13	13	0	0	6	12	0		20	24	.181	.359	.431	.789	119	31	2	0	0				
AL (21 yrs)				2475	10530	8327	2161	2860	506	136	708	2202	123	117	2042	1306	.343	.475	.692	1.167	207	5762		43	113				
NL (1 yr)				28	92	72	13	13	0	0	6	12	0		20	24	.181	.359	.431	.789	119	31	2	0	0				

Figure 1.1: Statistical information on the performance of Babe Ruth can be found at `http://www.baseball-reference.com`.

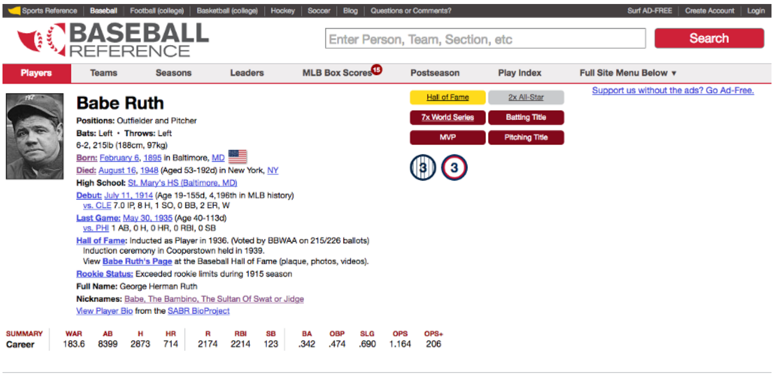
The key is thinking broadly: the answers to big, general questions often lie buried in highly-specific data sets, which were by no means designed to contain them.

1.2.1 The Baseball Encyclopedia

Baseball has long had an outsized importance in the world of data science. This sport has been called the national pastime of the United States; indeed, French historian Jacques Barzun observed that "Whoever wants to know the heart and mind of America had better learn baseball." I realize that many readers are not American, and even those that are might be completely disinterested in sports. But stick with me for a while.

What makes baseball important to data science is its extensive statistical record of play, dating back for well over a hundred years. Baseball is a sport of discrete events: pitchers throw balls and batters try to hit them – that naturally lends itself to informative statistics. Fans get immersed in these statistics as children, building their intuition about the strengths and limitations of quantitative analysis. Some of these children grow up to become data scientists. Indeed, the success of Brad Pitt's statistically-minded baseball team in the movie *Moneyball* remains the American public's most vivid contact with data science.

This historical baseball record is available at `http://www.baseball-reference.com`. There you will find complete statistical data on the performance of every player who even stepped on the field. This includes summary statistics of each season's batting, pitching, and fielding record, plus information about teams



Support us without the ads? Go Ad-Free.

Babe Ruth

Positions: Outfielder and Pitcher
Bats: Left • **Throws:** Left
6-2, 215lb (188cm, 97kg)
Born: February 6, 1895 in Baltimore, MD
Died: August 16, 1948 (Aged 53-192d) in New York, NY
High School: St. Mary's HS (Baltimore, MD)
Debut: July 11, 1914 (Age 19-155d, 4,196th in MLB history)
vs. CLE 7.0 IP, 8 H, 1 SO, 0 BB, 2 ER, W
Last Game: May 30, 1935 (Age 40-113d)
vs. PHI 1 AB, 0 H, 0 HR, 0 RBI, 0 SB
Hall of Fame: Inducted as Player in 1936. (Voted by BBWAA on 215/226 ballots)
Induction ceremony in Cooperstown held in 1939.
View **Babe Ruth's Page** at the Baseball Hall of Fame (plaque, photos, videos).
Rookie Status: Exceeded rookie limits during 1915 season
Full Name: George Herman Ruth
Nicknames: Babe, The Bambino, The Sultan Of Swat or Jidge
View Player Bio from the SABR BioProject

SUMMARY	WAR	AB	H	HR	R	RBI	SB	BA	OBP	SLG	OPS	OPS+
Career	183.6	8399	2873	714	2174	2214	123	.342	.474	.690	1.164	206

Figure 1.2: Personal information on every major league baseball player is available at `http://www.baseball-reference.com`.

and awards as shown in Figure 1.1.

But more than just statistics, there is metadata on the life and careers of all the people who have ever played major league baseball, as shown in Figure 1.2. We get the vital statistics of each player (height, weight, handedness) and their lifespan (when/where they were born and died). We also get salary information (how much each player got paid every season) and transaction data (how did they get to be the property of each team they played for).

Now, I realize that many of you do not have the slightest knowledge of or interest in baseball. This sport is somewhat reminiscent of cricket, if that helps. But remember that as a data scientist, it is your job to be interested in the world around you. Think of this as chance to learn something.

So what interesting questions can you answer with this baseball data set? Try to write down five questions before moving on. Don't worry, I will wait here for you to finish.

The most obvious types of questions to answer with this data are directly related to baseball:

- How can we best measure an individual player's skill or value?
- How fairly do trades between teams generally work out?
- What is the general trajectory of player's performance level as they mature and age?
- To what extent does batting performance correlate with position played? For example, are outfielders really better hitters than infielders?

These are interesting questions. But even more interesting are questions about demographic and social issues. Almost 20,000 major league baseball players have taken the field over the past 150 years, providing a large, extensively-documented cohort of men who can serve as a proxy for even larger, less well-documented populations. Indeed, we can use this baseball player data to answer questions like:

- Do left-handed people have shorter lifespans than right-handers? Handedness is not captured in most demographic data sets, but has been diligently assembled here. Indeed, analysis of this data set has been used to show that right-handed people live longer than lefties [HC88]!
- How often do people return to live in the same place where they were born? Locations of birth and death have been extensively recorded in this data set. Further, almost all of these people played at least part of their career far from home, thus exposing them to the wider world at a critical time in their youth.
- Do player salaries generally reflect past, present, or future performance?
- To what extent have heights and weights been increasing in the population at large?

There are two particular themes to be aware of here. First, the identifiers and reference tags (i.e. the metadata) often prove more interesting in a data set than the stuff we are supposed to care about, here the statistical record of play.

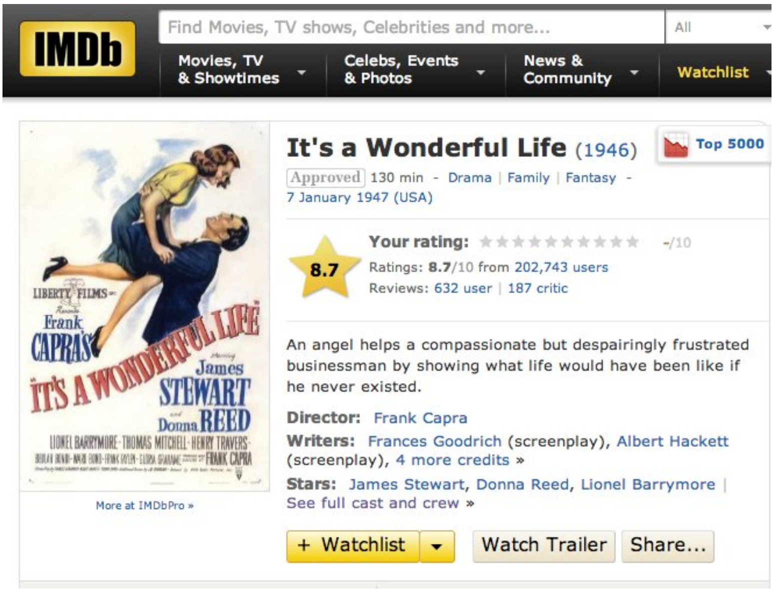
Second is the idea of a *statistical proxy*, where you use the data set you have to substitute for the one you really want. The data set of your dreams likely does not exist, or may be locked away behind a corporate wall even if it does. A good data scientist is a pragmatist, seeing what they can do with what they have instead of bemoaning what they cannot get their hands on.

1.2.2 The Internet Movie Database (IMDb)

Everybody loves the movies. The Internet Movie Database (IMDb) provides crowdsourced and curated data about all aspects of the motion picture industry, at `www.imdb.com`. IMDb currently contains data on over 3.3 million movies and TV programs. For each film, IMDb includes its title, running time, genres, date of release, and a full list of cast and crew. There is financial data about each production, including the budget for making the film and how well it did at the box office.

Finally, there are extensive ratings for each film from viewers and critics. This rating data consists of scores on a zero to ten stars scale, cross-tabulated into averages by age and gender. Written reviews are often included, explaining why a particular critic awarded a given number of stars. There are also links between films: for example, identifying which other films have been watched most often by viewers of *It's a Wonderful Life*.

Every actor, director, producer, and crew member associated with a film merits an entry in IMDb, which now contains records on 6.5 million people.



It's a Wonderful Life (1946)

Approved 130 min - Drama | Family | Fantasy - 7 January 1947 (USA)

Your rating: ★★★★★★★★★★ -/10

8.7

Ratings: **8.7**/10 from 202,743 users
Reviews: 632 user | 187 critic

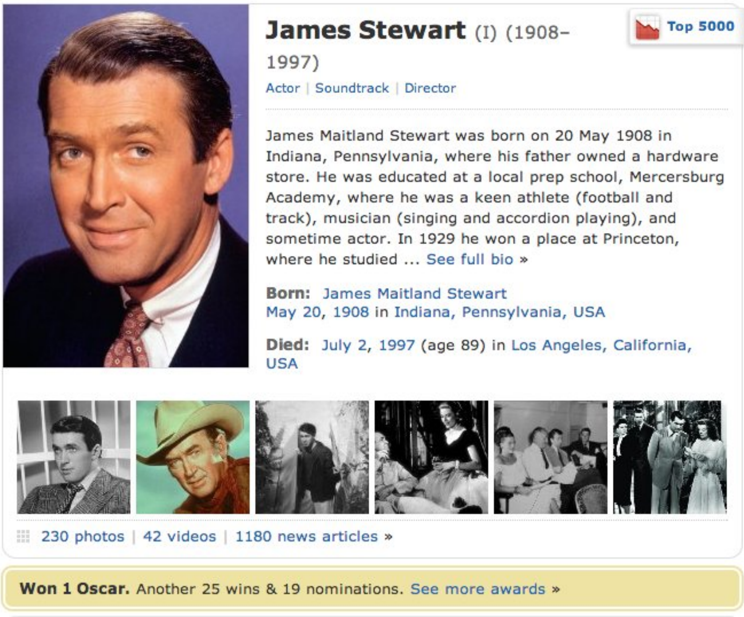
An angel helps a compassionate but despairingly frustrated businessman by showing what life would have been like if he never existed.

Director: Frank Capra

Writers: Frances Goodrich (screenplay), Albert Hackett (screenplay), 4 more credits »

Stars: James Stewart, Donna Reed, Lionel Barrymore | See full cast and crew »

Figure 1.3: Representative film data from the Internet Movie Database.



James Stewart (I) (1908–1997)

Actor | Soundtrack | Director

James Maitland Stewart was born on 20 May 1908 in Indiana, Pennsylvania, where his father owned a hardware store. He was educated at a local prep school, Mercersburg Academy, where he was a keen athlete (football and track), musician (singing and accordion playing), and sometime actor. In 1929 he won a place at Princeton, where he studied ... See full bio »

Born: James Maitland Stewart
May 20, 1908 in Indiana, Pennsylvania, USA

Died: July 2, 1997 (age 89) in Los Angeles, California, USA

230 photos | 42 videos | 1180 news articles »

Won 1 Oscar. Another 25 wins & 19 nominations. See more awards »

Figure 1.4: Representative actor data from the Internet Movie Database.

These happen to include my brother, cousin, and sister-in-law. Each actor is linked to every film they appeared in, with a description of their role and their ordering in the credits. Available data about each personality includes birth/death dates, height, awards, and family relations.

So what kind of questions can you answer with this movie data?

Perhaps the most natural questions to ask IMDb involve identifying the extremes of movies and actors:

- Which actors appeared in the most films? Earned the most money? Appeared in the lowest rated films? Had the longest career or the shortest lifespan?
- What was the highest rated film each year, or the best in each genre? Which movies lost the most money, had the highest-powered casts, or got the least favorable reviews.

Then there are larger-scale questions one can ask about the nature of the motion picture business itself:

- How well does movie gross correlate with viewer ratings or awards? Do customers instinctively flock to trash, or is virtue on the part of the creative team properly rewarded?
- How do Hollywood movies compare to Bollywood movies, in terms of ratings, budget, and gross? Are American movies better received than foreign films, and how does this differ between U.S. and non-U.S. reviewers?
- What is the age distribution of actors and actresses in films? How much younger is the actress playing the wife, on average, than the actor playing the husband? Has this disparity been increasing or decreasing with time?
- Live fast, die young, and leave a good-looking corpse? Do movie stars live longer or shorter lives than bit players, or compared to the general public?

Assuming that people working together on a film get to know each other, the cast and crew data can be used to build a social network of the movie business. What does the social network of actors look like? The Oracle of Bacon (`https://oracleofbacon.org/`) posits Kevin Bacon as the center of the Hollywood universe and generates the shortest path to Bacon from any other actor. Other actors, like Samuel L. Jackson, prove even more central.

More critically, can we analyze this data to determine the probability that someone will like a given movie? The technique of *collaborative filtering* finds people who liked films that I also liked, and recommends other films that *they* liked as good candidates for me. The 2007 Netflix Prize was a $1,000,000 competition to produce a ratings engine 10% better than the proprietary Netflix system. The ultimate winner of this prize (BellKor) used a variety of data sources and techniques, including the analysis of links [BK07].

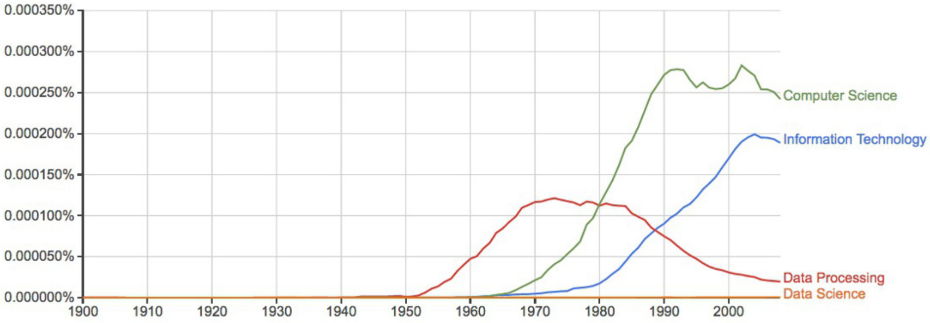


Figure 1.5: The rise and fall of data processing, as witnessed by Google Ngrams.

1.2.3 Google Ngrams

Printed books have been the primary repository of human knowledge since Gutenberg's invention of movable type in 1439. Physical objects live somewhat uneasily in today's digital world, but technology has a way of reducing everything to data. As part of its mission to organize the world's information, Google undertook an effort to scan all of the world's published books. They haven't quite gotten there yet, but the 30 million books thus far digitized represent over 20% of all books ever published.

Google uses this data to improve search results, and provide fresh access to out-of-print books. But perhaps the coolest product is *Google Ngrams*, an amazing resource for monitoring changes in the cultural zeitgeist. It provides the frequency with which short phrases occur in books published each year. Each phrase must occur at least forty times in their scanned book corpus. This eliminates obscure words and phrases, but leaves over two billion time series available for analysis.

This rich data set shows how language use has changed over the past 200 years, and has been widely applied to cultural trend analysis [MAV$^+$11]. Figure 1.5 uses this data to show how the word *data* fell out of favor when thinking about computing. *Data processing* was the popular term associated with the computing field during the punched card and spinning magnetic tape era of the 1950s. The Ngrams data shows that the rapid rise of *Computer Science* did not eclipse *Data Processing* until 1980. Even today, *Data Science* remains almost invisible on this scale.

Check out Google Ngrams at `http://books.google.com/ngrams`. I promise you will enjoy playing with it. Compare *hot dog* to *tofu*, *science* against *religion*, *freedom* to *justice*, and *sex* vs. *marriage*, to better understand this fantastic telescope for looking into the past.

But once you are done playing, think of bigger things you could do if you got your hands on this data. Assume you have access to the annual number of references for *all* words/phrases published in books over the past 200 years.

Google makes this data freely available. So what are you going to do with it?

Observing the time series associated with particular words using the Ngrams Viewer is fun. But more sophisticated historical trends can be captured by aggregating multiple time series together. The following types of questions seem particularly interesting to me:

- How has the amount of cursing changed over time? Use of the four-letter words I am most familiar with seem to have exploded since 1960, although it is perhaps less clear whether this reflects increased cussing or lower publication standards.

- How often do new words emerge and get popular? Do these words tend to stay in common usage, or rapidly fade away? Can we detect when words change meaning over time, like the transition of *gay* from *happy* to *homosexual*?

- Have standards of spelling been improving or deteriorating with time, especially now that we have entered the era of automated spell checking? Rarely-occurring words that are only one character removed from a commonly-used word are likely candidates to be spelling errors (e.g. *algorithm* vs. *algorthm*). Aggregated over many different misspellings, are such errors increasing or decreasing?

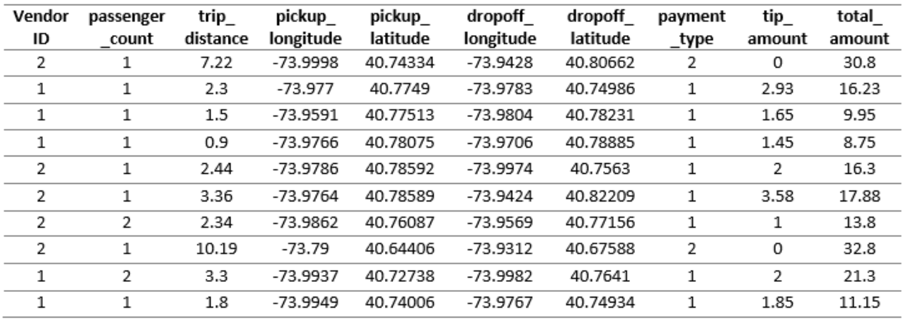
You can also use this Ngrams corpus to build a language model that captures the meaning and usage of the words in a given language. We will discuss word embeddings in Section 11.6.3, which are powerful tools for building language models. Frequency counts reveal which words are most popular. The frequency of word pairs appearing next to each other can be used to improve speech recognition systems, helping to distinguish whether the speaker said *that's too bad* or *that's to bad*. These millions of books provide an ample data set to build representative models from.

1.2.4 New York Taxi Records

Every financial transaction today leaves a data trail behind it. Following these paths can lead to interesting insights.

Taxi cabs form an important part of the urban transportation network. They roam the streets of the city looking for customers, and then drive them to their destination for a fare proportional to the length of the trip. Each cab contains a metering device to calculate the cost of the trip as a function of time. This meter serves as a record keeping device, and a mechanism to ensure that the driver charges the proper amount for each trip.

The taxi meters currently employed in New York cabs can do many things beyond calculating fares. They act as credit card terminals, providing a way



Vendor ID	passenger _count	trip_ distance	pickup_ longitude	pickup_ latitude	dropoff_ longitude	dropoff_ latitude	payment _type	tip_ amount	total_ amount
2	1	7.22	-73.9998	40.74334	-73.9428	40.80662	2	0	30.8
1	1	2.3	-73.977	40.7749	-73.9783	40.74986	1	2.93	16.23
1	1	1.5	-73.9591	40.77513	-73.9804	40.78231	1	1.65	9.95
1	1	0.9	-73.9766	40.78075	-73.9706	40.78885	1	1.45	8.75
2	1	2.44	-73.9786	40.78592	-73.9974	40.7563	1	2	16.3
2	1	3.36	-73.9764	40.78589	-73.9424	40.82209	1	3.58	17.88
2	2	2.34	-73.9862	40.76087	-73.9569	40.77156	1	1	13.8
2	1	10.19	-73.79	40.64406	-73.9312	40.67588	2	0	32.8
1	2	3.3	-73.9937	40.72738	-73.9982	40.7641	1	2	21.3
1	1	1.8	-73.9949	40.74006	-73.9767	40.74934	1	1.85	11.15

Figure 1.6: Representative fields from the New York city taxi cab data: pick up and dropoff points, distances, and fares.

for customers to pay for rides without cash. They are integrated with global positioning systems (GPS), recording the exact location of every pickup and drop off. And finally, since they are on a wireless network, these boxes can communicate all of this data back to a central server.

The result is a database documenting every single trip by all taxi cabs in one of the world's greatest cities, a small portion of which is shown in Figure 1.6. Because the New York Taxi and Limousine Commission is a public agency, its non-confidential data is available to all under the Freedom of Information Act (FOA).

Every ride generates two records: one with data on the trip, the other with details of the fare. Each trip is keyed to the medallion (license) of each car coupled with the identifier of each driver. For each trip, we get the time/date of pickup and drop-off, as well as the GPS coordinates (longitude and latitude) of the starting location and destination. We do not get GPS data of the route they traveled between these points, but to some extent that can be inferred by the shortest path between them.

As for fare data, we get the metered cost of each trip, including tax, surcharge and tolls. It is traditional to pay the driver a tip for service, the amount of which is also recorded in the data.

So I'm talking to you. This taxi data is readily available, with records of over 80 million trips over the past several years. What are you going to do with it?

Any interesting data set can be used to answer questions on many different scales. This taxi fare data can help us better understand the transportation industry, but also how the city works and how we could make it work even better. Natural questions with respect to the taxi industry include:

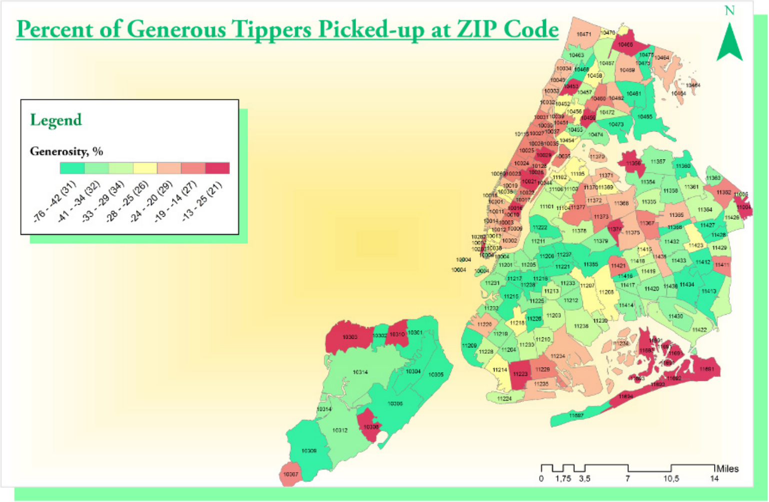


Figure 1.7: Which neighborhoods in New York city tip most generously? The relatively remote outer boroughs of Brooklyn and Queens, where trips are longest and supply is relatively scarce.

- How much money do drivers make each night, on average? What is the distribution? Do drivers make more on sunny days or rainy days?

- Where are the best spots in the city for drivers to cruise, in order to pick up profitable fares? How does this vary at different times of the day?

- How far do drivers travel over the course of a night's work? We can't answer this exactly using this data set, because it does not provide GPS data of the route traveled between fares. But we do know the last place of drop off, the next place of pickup, and how long it took to get between them. Together, this should provide enough information to make a sound estimate.

- Which drivers take their unsuspecting out-of-town passengers for a "ride," running up the meter on what should be a much shorter, cheaper trip?

- How much are drivers tipped, and why? Do faster drivers get tipped better? How do tipping rates vary by neighborhood, and is it the rich neighborhoods or poor neighborhoods which prove more generous?

  I will confess we did an analysis of this, which I will further describe in the war story of Section 9.3. We found a variety of interesting patterns [SS15]. Figure 1.7 shows that Manhattanites are generally cheapskates relative to large swaths of Brooklyn, Queens, and Staten Island, where trips are longer and street cabs a rare but welcome sight.

But the bigger questions have to do with understanding transportation in the city. We can use the taxi travel times as a sensor to measure the level of traffic in the city at a fine level. How much slower is traffic during rush hour than other times, and where are delays the worst? Identifying problem areas is the first step to proposing solutions, by changing the timing patterns of traffic lights, running more buses, or creating high-occupancy only lanes.

Similarly we can use the taxi data to measure transportation flows across the city. Where are people traveling to, at different times of the day? This tells us much more than just congestion. By looking at the taxi data, we should be able to see tourists going from hotels to attractions, executives from fancy neighborhoods to Wall Street, and drunks returning home from nightclubs after a bender.

Data like this is essential to designing better transportation systems. It is wasteful for a single rider to travel from point a to point b when there is another rider at point $a+\epsilon$ who also wants to get there. Analysis of the taxi data enables accurate simulation of a ride sharing system, so we can accurately evaluate the demands and cost reductions of such a service.

1.3 Properties of Data

This book is about techniques for analyzing data. But what is the underlying stuff that we will be studying? This section provides a brief taxonomy of the properties of data, so we can better appreciate and understand what we will be working on.

1.3.1 Structured vs. Unstructured Data

Certain data sets are nicely structured, like the tables in a database or spreadsheet program. Others record information about the state of the world, but in a more heterogeneous way. Perhaps it is a large text corpus with images and links like Wikipedia, or the complicated mix of notes and test results appearing in personal medical records.

Generally speaking, this book will focus on dealing with structured data. Data is often represented by a *matrix*, where the rows of the matrix represent distinct items or records, and the columns represent distinct properties of these items. For example, a data set about U.S. cities might contain one row for each city, with columns representing features like state, population, and area.

When confronted with an unstructured data source, such as a collection of tweets from Twitter, our first step is generally to build a matrix to structure it. A *bag of words* model will construct a matrix with a row for each tweet, and a column for each frequently used vocabulary word. Matrix entry $M[i,j]$ then denotes the number of times tweet i contains word j. Such matrix formulations will motivate our discussion of linear algebra, in Chapter 8.

1.3.2 Quantitative vs. Categorical Data

Quantitative data consists of numerical values, like height and weight. Such data can be incorporated directly into algebraic formulas and mathematical models, or displayed in conventional graphs and charts.

By contrast, *categorical data* consists of labels describing the properties of the objects under investigation, like gender, hair color, and occupation. This descriptive information can be every bit as precise and meaningful as numerical data, but it cannot be worked with using the same techniques.

Categorical data can usually be coded numerically. For example, gender might be represented as $male = 0$ or $female = 1$. But things get more complicated when there are more than two characters per feature, especially when there is not an implicit order between them. We may be able to encode hair colors as numbers by assigning each shade a distinct value like gray hair = 0, red hair = 1, and blond hair = 2. However, we cannot really treat these values as numbers, for anything other than simple identity testing. Does it make any sense to talk about the maximum or minimum hair color? What is the interpretation of my hair color minus your hair color?

Most of what we do in this book will revolve around numerical data. But keep an eye out for categorical features, and methods that work for them. Classification and clustering methods can be thought of as generating categorical labels from numerical data, and will be a primary focus in this book.

1.3.3 Big Data vs. Little Data

Data science has become conflated in the public eye with *big data*, the analysis of massive data sets resulting from computer logs and sensor devices. In principle, having more data is always better than having less, because you can always throw some of it away by sampling to get a smaller set if necessary.

Big data is an exciting phenomenon, and we will discuss it in Chapter 12. But in practice, there are difficulties in working with large data sets. Throughout this book we will look at algorithms and best practices for analyzing data. In general, things get harder once the volume gets too large. The challenges of big data include:

- *The analysis cycle time slows as data size grows*: Computational operations on data sets take longer as their volume increases. Small spreadsheets provide instantaneous response, allowing you to experiment and play *what if?* But large spreadsheets can be slow and clumsy to work with, and massive-enough data sets might take hours or days to get answers from.

  Clever algorithms can permit amazing things to be done with big data, but staying small generally leads to faster analysis and exploration.

- *Large data sets are complex to visualize*: Plots with millions of points on them are impossible to display on computer screens or printed images, let alone conceptually understand. How can we ever hope to really understand something we cannot see?

- *Simple models do not require massive data to fit or evaluate*: A typical data science task might be to make a decision (say, whether I should offer this fellow life insurance?) on the basis of a small number of variables: say age, gender, height, weight, and the presence or absence of existing medical conditions.

  If I have this data on 1 million people with their associated life outcomes, I should be able to build a good general model of coverage risk. It probably wouldn't help me build a substantially better model if I had this data on hundreds of millions of people. The decision criteria on only a few variables (like age and martial status) cannot be too complex, and should be robust over a large number of applicants. Any observation that is so subtle it requires massive data to tease out will prove irrelevant to a large business which is based on volume.

Big data is sometimes called *bad data*. It is often gathered as the by-product of a given system or procedure, instead of being purposefully collected to answer your question at hand. The result is that we might have to go to heroic efforts to make sense of something just because we have it.

Consider the problem of getting a pulse on voter preferences among presidential candidates. The big data approach might analyze massive Twitter or Facebook feeds, interpreting clues to their opinions in the text. The small data approach might be to conduct a poll, asking a few hundred people this specific question and tabulating the results. Which procedure do you think will prove more accurate? The right data set is the one most directly relevant to the tasks at hand, not necessarily the biggest one.

> *Take-Home Lesson*: Do not blindly aspire to analyze large data sets. Seek the *right* data to answer a given question, not necessarily the biggest thing you can get your hands on.

1.4 Classification and Regression

Two types of problems arise repeatedly in traditional data science and pattern recognition applications, the challenges of classification and regression. As this book has developed, I have pushed discussions of the algorithmic approaches to solving these problems toward the later chapters, so they can benefit from a solid understanding of core material in data munging, statistics, visualization, and mathematical modeling.

Still, I will mention issues related to classification and regression as they arise, so it makes sense to pause here for a quick introduction to these problems, to help you recognize them when you see them.

- *Classification*: Often we seek to assign a label to an item from a discrete set of possibilities. Such problems as predicting the winner of a particular sporting contest (team A or team B?) or deciding the genre of a given movie (comedy, drama, or animation?) are *classification* problems, since each entail selecting a label from the possible choices.

- *Regression*: Another common task is to forecast a given numerical quantity. Predicting a person's weight or how much snow we will get this year is a *regression* problem, where we forecast the future value of a numerical function in terms of previous values and other relevant features.

Perhaps the best way to see the intended distinction is to look at a variety of data science problems and label (classify) them as regression or classification. Different algorithmic methods are used to solve these two types of problems, although the same questions can often be approached in either way:

- Will the price of a particular stock be higher or lower tomorrow? (classification)
- What will the price of a particular stock be tomorrow? (regression)
- Is this person a good risk to sell an insurance policy to? (classification)
- How long do we expect this person to live? (regression)

Keep your eyes open for classification and regression problems as you encounter them in your life, and in this book.

1.5 Data Science Television: The Quant Shop

I believe that hands-on experience is necessary to internalize basic principles. Thus when I teach data science, I like to give each student team an interesting but messy forecasting challenge, and demand that they build and evaluate a predictive model for the task.

These forecasting challenges are associated with events where the students must make testable predictions. They start from scratch: finding the relevant data sets, building their own evaluation environments, and devising their model. Finally, I make them watch the event as it unfolds, so as to witness the vindication or collapse of their prediction.

As an experiment, we documented the evolution of each group's project on video in Fall 2014. Professionally edited, this became *The Quant Shop*, a television-like data science series for a general audience. The eight episodes of this first season are available at `http://www.quant-shop.com`, and include:

- *Finding Miss Universe* – The annual Miss Universe competition aspires to identify the most beautiful woman in the world. Can computational models predict who will win a beauty contest? Is beauty just subjective, or can algorithms tell who is the fairest one of all?

- *Modeling the Movies* – The business of movie making involves a lot of high-stakes data analysis. Can we build models to predict which film will gross the most on Christmas day? How about identifying which actors will receive awards for their performance?

- *Winning the Baby Pool* – Birth weight is an important factor in assessing the health of a newborn child. But how accurately can we predict junior's weight before the actual birth? How can data clarify environmental risks to developing pregnancies?

- *The Art of the Auction* – The world's most valuable artworks sell at auctions to the highest bidder. But can we predict how many millions a particular J.W. Turner painting will sell for? Can computers develop an artistic sense of what's worth buying?

- *White Christmas* – Weather forecasting is perhaps the most familiar domain of predictive modeling. Short-term forecasts are generally accurate, but what about longer-term prediction? What places will wake up to a snowy Christmas this year? And can you tell one month in advance?

- *Predicting the Playoffs* – Sports events have winners and losers, and bookies are happy to take your bets on the outcome of any match. How well can statistics help predict which football team will win the Super Bowl? Can Google's PageRank algorithm pick the winners on the field as accurately as it does on the web?

- *The Ghoul Pool* – Death comes to all men, but when? Can we apply actuarial models to celebrities, to decide who will be the next to die? Similar analysis underlies the workings of the life insurance industry, where accurate predictions of lifespan are necessary to set premiums which are both sustainable and affordable.

Figure 1.8: Exciting scenes from data science television: *The Quant Shop*.

- *Playing the Market* – Hedge fund quants get rich when guessing right about tomorrow's prices, and poor when wrong. How accurately can we predict future prices of gold and oil using histories of price data? What other information goes into building a successful price model?

I encourage you to watch some episodes of *The Quant Shop* in tandem with reading this book. We try to make it fun, although I am sure you will find plenty of things to cringe at. Each show runs for thirty minutes, and maybe will inspire you to tackle a prediction challenge of your own.

These programs will certainly give you more insight into these eight specific challenges. I will use these projects throughout this book to illustrate important lessons in how to do data science, both as positive and negative examples. These projects provide a laboratory to see how intelligent but inexperienced people not wildly unlike yourself thought about a data science problem, and what happened when they did.

1.5.1 Kaggle Challenges

Another source of inspiration are challenges from Kaggle (`www.kaggle.com`), which provides a competitive forum for data scientists. New challenges are posted on a regular basis, providing a problem definition, training data, and a scoring function over hidden evaluation data. A leader board displays the scores of the strongest competitors, so you can see how well your model stacks up in comparison with your opponents. The winners spill their modeling secrets during post-contest interviews, to help you improve your modeling skills.

Performing well on Kaggle challenges is an excellent credential to put on your resume to get a good job as a data scientist. Indeed, potential employers will track you down if you are a real Kaggle star. But the real reason to participate is that the problems are fun and inspiring, and practice helps make you a better data scientist.

The exercises at the end of each chapter point to expired Kaggle challenges, loosely connected to the material in that chapter. Be forewarned that Kaggle provides a misleading glamorous view of data science as applied machine learning, because it presents extremely well-defined problems with the hard work of data collection and cleaning already done for you. Still, I encourage you to check it out for inspiration, and as a source of data for new projects.

1.6 About the War Stories

Genius and wisdom are two distinct intellectual gifts. *Genius* shows in discovering the right answer, making imaginative mental leaps which overcome obstacles and challenges. *Wisdom* shows in avoiding obstacles in the first place, providing a sense of direction or guiding light that keeps us moving soundly in the right direction.

Genius is manifested in technical strength and depth, the ability to see things and do things that other people cannot. In contrast, wisdom comes from experience and general knowledge. It comes from listening to others. Wisdom comes from humility, observing how often you have been wrong in the past and figuring out why you were wrong, so as to better recognize future traps and avoid them.

Data science, like most things in life, benefits more from wisdom than from genius. In this book, I seek to pass on wisdom that I have accumulated the hard way through *war stories*, gleaned from a diverse set of projects I have worked on:

- *Large-scale text analytics and NLP*: My Data Science Laboratory at Stony Brook University works on a variety of projects in big data, including sentiment analysis from social media, historical trends analysis, deep learning approaches to natural language processing (NLP), and feature extraction from networks.

- *Start-up companies*: I served as co-founder and chief scientist to two data analytics companies: General Sentiment and Thrivemetrics. General Sentiment analyzed large-scale text streams from news, blogs, and social media to identify trends in the sentiment (positive or negative) associated with people, places, and things. Thrivemetrics applied this type of analysis to internal corporate communications, like email and messaging systems.

  Neither of these ventures left me wealthy enough to forgo my royalties from this book, but they did provide me with experience on cloud-based computing systems, and insight into how data is used in industry.

- *Collaborating with real scientists*: I have had several interesting collaborations with biologists and social scientists, which helped shape my understanding of the complexities of working with real data. Experimental data is horribly noisy and riddled with errors, yet you must do the best you can with what you have, in order to discover how the world works.

- *Building gambling systems*: A particularly amusing project was building a system to predict the results of jai-alai matches so we could bet on them, an experience recounted in my book *Calculated Bets: Computers, Gambling, and Mathematical Modeling to Win* [Ski01]. Our system relied on web scraping for data collection, statistical analysis, simulation/modeling, and careful evaluation. We also have developed and evaluated predictive models for movie grosses [ZS09], stock prices [ZS10], and football games [HS10] using social media analysis.

- *Ranking historical figures*: By analyzing Wikipedia to extract meaningful variables on over 800,000 historical figures, we developed a scoring function to rank them by their strength as historical memes. This ranking does a great job separating the greatest of the great (Jesus, Napoleon, Shakespeare, Mohammad, and Lincoln round out the top five) from lesser mortals, and served as the basis for our book *Who's Bigger?: Where Historical Figures Really Rank* [SW13].

All this experience drives what I teach in this book, especially the tales that I describe as war stories. *Every one of these war stories is true.* Of course, the stories improve somewhat in the retelling, and the dialogue has been punched up to make them more interesting to read. However, I have tried to honestly trace the process of going from a raw problem to a solution, so you can watch how it unfolded.

1.7 War Story: Answering the Right Question

Our research group at Stony Brook University developed an NLP-based system for analyzing millions of news, blogs and social media messages, and reducing this text to trends concerning all the entities under discussion. Counting the number of mentions each name receives in a text stream (volume) is easy, in principle. Determining whether the connotation of a particular reference is positive or negative (sentiment analysis) is hard. But our system did a pretty good job, particularly when aggregated over many references.

This technology served as the foundation for a social media analysis company named General Sentiment. It was exciting living through a start-up starting up, facing the challenges of raising money, hiring staff, and developing new products.

But perhaps the biggest problem we faced was answering the right question. The General Sentiment system recorded trends about the sentiment and volume for *every* person, place, and thing that was ever mentioned in news, blogs, and social media: over 20 million distinct entities. We monitored the reputations of celebrities and politicians. We monitored the fates of companies and products. We tracked the performance of sports teams, and the buzz about movies. We could do anything!

But it turns out that no one pays you to do anything. They pay you to do *something*, to solve a particular problem they have, or eliminate a specific pain point in their business. Being able to do anything proves to be a terrible sales strategy, because it requires you to find that need afresh for each and every customer.

Facebook didn't open up to the world until September 2006. So when General Sentiment started in 2008, we were at the very beginning of the social media era. We had lots of interest from major brands and advertising agencies which *knew* that social media was ready to explode. They *knew* this newfangled thing was important, and that they had to be there. They *knew* that proper analysis of social media data could give them fresh insights into what their customers were thinking. But they didn't know exactly what it was they really wanted to know.

One aircraft engine manufacturer was very interested in learning how much the kids talked about them on Facebook. We had to break it to them gently that the answer was zero. Other potential customers demanded proof that we were more accurate than the Nielsen television ratings. But of course, if you wanted Nielsen ratings then you should buy them from Nielsen. Our system provided different insights from a completely different world. But you had to know what you wanted in order to use them.

We did manage to get substantial contracts from a very diverse group of customers, including consumer brands like Toyota and Blackberry, governmental organizations like the Hawaii tourism office, and even the presidential campaign of Republican nominee Mitt Romney in 2012. Our analysts provided them insights into a wide variety of business issues:

- What did people think about Hawaii? (Answer: they think it is a very nice place to visit.)
- How quickly would Toyota's sentiment recover after news of serious brake problems in their cars? (Answer: about six months.)
- What did people think about Blackberry's new phone models? (Answer: they liked the iPhone much better.)
- How quickly would Romney's sentiment recover after insulting 47% of the electorate in a recorded speech? (Answer: never.)

But each sale required entering a new universe, involving considerable effort and imagination on the part of our sales staff and research analysts. We never managed to get two customers in the same industry, which would have let us benefit from scale and accumulated wisdom.

Of course, the customer is always right. It was our fault that we could not explain to them the best way to use our technology. The lesson here is that the world will not beat a path to your door just for a new source of data. You must be able to supply the right questions before you can turn data into money.

1.8 Chapter Notes

The idea of using historical records from baseball players to establish that left-handers have shorter lifespans is due to Halpern and Coren [HC88, HC91], but their conclusion remains controversial. The percentage of left-handers in the population has been rapidly growing, and the observed effects may be a function of survivorship bias [McM04]. So lefties, hang in there! Full disclosure: I am one of you.

The discipline of quantitative baseball analysis is sometimes called *sabermetrics*, and its leading light is a fellow named Bill James. I recommend budding data scientists read his *Historical Baseball Abstract* [Jam10] as an excellent example of how one turns numbers into knowledge and understanding. *Time Magazine* once said of James: "Much of the joy of reading him comes from the extravagant spectacle of a first-rate mind wasting itself on baseball." I thank `http://sports-reference.com` for permission to use images of their website in this book. Ditto to Amazon, the owner of IMDb.

The potential of ride-sharing systems in New York was studied by Santi et. al. [SRS$^+$14], who showed that almost 95% of the trips could have been shared with no more than five minutes delay per trip.

The Lydia system for sentiment analysis is described in [GSS07]. Methods to identify changes in word meaning through analysis of historical text corpora like Google Ngram are reported in [KARPS15].

1.9 Exercises

Identifying Data Sets

1-1. *[3]* Identify where interesting data sets relevant to the following domains can be found on the web:

(a) Books.

(b) Horse racing.

(c) Stock prices.

(d) Risks of diseases.

(e) Colleges and universities.

(f) Crime rates.

(g) Bird watching.

For each of these data sources, explain what you must do to turn this data into a usable format on your computer for analysis.

1-2. *[3]* Propose relevant data sources for the following *The Quant Shop* prediction challenges. Distinguish between sources of data that you are sure *somebody* must have, and those where the data is clearly available to you.

(a) *Miss Universe.*

(b) *Movie gross.*

(c) *Baby weight.*

(d) *Art auction price.*

(e) *White Christmas.*

(f) *Football champions.*

(g) *Ghoul pool.*

(h) *Gold/oil prices.*

1-3. *[3]* Visit `http://data.gov`, and identify five data sets that sound interesting to you. For each write a brief description, and propose three interesting things you might do with them.

Asking Questions

1-4. *[3]* For each of the following data sources, propose three interesting questions you can answer by analyzing them:

(a) Credit card billing data.

(b) Click data from `http://www.Amazon.com`.

(c) White Pages residential/commercial telephone directory.

1-5. *[5]* Visit Entrez, the National Center for Biotechnology Information (NCBI) portal. Investigate what data sources are available, particularly the Pubmed and Genome resources. Propose three interesting projects to explore with each of them.

1-6. *[5]* You would like to conduct an experiment to establish whether your friends prefer the taste of regular Coke or Diet Coke. Briefly outline a design for such a study.

1-7. *[5]* You would like to conduct an experiment to see whether students learn better if they study without any music, with instrumental music, or with songs that have lyrics. Briefly outline the design for such a study.

1-8. *[5]* Traditional polling operations like Gallup use a procedure called random digit dialing, which dials random strings of digits instead of picking phone numbers from the phone book. Suggest why such polls are conducted using random digit dialing.

Implementation Projects

1-9. *[5]* Write a program to scrape the best-seller rank for a book on Amazon.com. Use this to plot the rank of all of Skiena's books over time. Which one of these books should be the next item that you purchase? Do you have friends for whom they would make a welcome and appropriate gift? :-)

1-10. *[5]* For your favorite sport (baseball, football, basketball, cricket, or soccer) identify a data set with the historical statistical records for all major participants. Devise and implement a ranking system to identify the best player at each position.

Interview Questions

1-11. *[3]* For each of the following questions: (1) produce a quick guess based only on your understanding of the world, and then (2) use Google to find supportable numbers to produce a more principled estimate from. How much did your two estimates differ by?

(a) How many piano tuners are there in the entire world?

(b) How much does the ice in a hockey rink weigh?

(c) How many gas stations are there in the United States?

(d) How many people fly in and out of LaGuardia Airport every day?

(e) How many gallons of ice cream are sold in the United States each year?

(f) How many basketballs are purchased by the National Basketball Association (NBA) each year?

(g) How many fish are there in all the world's oceans?

(h) How many people are flying in the air right now, all over the world?

(i) How many ping-pong balls can fit in a large commercial jet?

(j) How many miles of paved road are there in your favorite country?

(k) How many dollar bills are sitting in the wallets of all people at Stony Brook University?

(l) How many gallons of gasoline does a typical gas station sell per day?

(m) How many words are there in this book?

(n) How many cats live in New York city?

(o) How much would it cost to fill a typical car's gas tank with Starbuck's coffee?

(p) How much tea is there in China?

(q) How many checking accounts are there in the United States?

1-12. *[3]* What is the difference between regression and classification?

1-13. *[8]* How would you build a data-driven recommendation system? What are the limitations of this approach?

1-14. *[3]* How did you become interested in data science?

1-15. *[3]* Do you think data science is an art or a science?

Kaggle Challenges

1-16. Who survived the shipwreck of the Titanic?

`https://www.kaggle.com/c/titanic`

1-17. Where is a particular taxi cab going?

`https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i`

1-18. How long will a given taxi trip take?

`https://www.kaggle.com/c/pkdd-15-taxi-trip-time-prediction-ii`