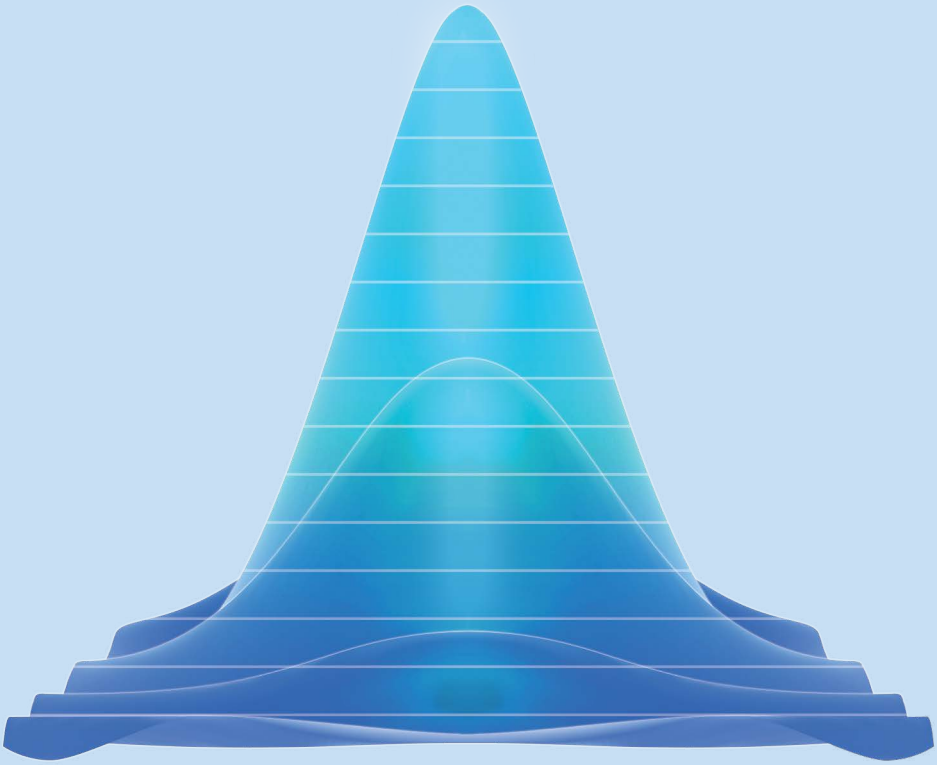
THE DATA SCIENCE HANDBOOK



ADVICE AND INSIGHTS FROM 25 AMAZING DATA SCIENTISTS

FOREWORD BY JAKE **KLAMKA**

DJ **Patil,** Hilary **Mason,** Pete **Skomoroch,** Riley **Newman,** Jonathan **Goldman,** Michael **Hochster,** George **Roumeliotis,** Kevin **Novak,** Jace **Kohlmeier,** Chris **Moody,** Erich **Owens,** Luis **Sanchez,** Eithon **Cadag,** Sean **Gourley,** Clare **Corthell,** Diane **Wu,** Joe **Blitzstein,** Josh **Wills,** Bradley **Voytek,** Michelangelo **D'Agostino,** Mike **Dewar,** Kunal **Punera,** William **Chen,** John **Foreman,** Drew **Conway**

BY CARL **SHAN** HENRY **WANG** WILLIAM **CHEN** MAX **SONG**

To our family, friends and mentors.
Your support and encouragement is the fuel for our fire.

CONTENTS

CONTENTS

CONTENTS

PREFACE

In the past five years, data science has made an impact in almost every major area of human endeavour. From commerce, to education, to energy, and of course, software and the Internet, data science has created immense value across the world. In fact, in early 2015 the President of the United States announced the new role of Chief Data Scientist to the White House and appointed DJ Patil, one of the interviewees in this book, to the position.

Like many innovations in the world, the birth of this industry was started by a few motivated people. Over the last few years, they founded, developed and advocated for the value that data analytics can bring to every industry. In The Data Science Handbook, you will have the opportunity to meet many of these founding data scientists, hear first hand accounts of the incredible journeys they took, and read where they believe the field is headed.

The road to becoming a data scientist is not always an easy one. When I tried to transition from experimental particle physics to industry, resources were few and far between. In fact, although a need for data science existed in companies, the job title had not even been created. I spent a lot of time teaching myself, working on various startup projects, and later saw many of my friends from academia run into the same challenges.

I observed a groundswell of incredibly gifted and highly trained researchers who were excited about moving into data-driven roles, yet were missing key pieces of knowledge about how to do so. As a result, they had trouble transferring their incredible quantitative and data analysis research skills to a career in industry. Meanwhile, having lived and worked in Silicon Valley, I also saw that there was very strong demand from technology companies who wanted to hire these exact people.

To help others bridge the gap between academia and industry, I founded the Insight Data Science Fellows Program in 2012. Insight is a training fellowship that helps quantitative PhDs transition from academia to industry. Over the last few years, we've helped hundreds of Insight Fellows, from fields like physics, computational biology, neuroscience, math, and engineering, transition from a background in academia to become leading data scientists at companies like Facebook, Airbnb, LinkedIn, The New York Times, Memorial Sloan Kettering Cancer Center and nearly a hundred other companies.

In my personal journey to both entering the technology field as well as creating a community for others to do the same, one key resource I found to be tremendously useful was conversations with those who had successfully made the transition. As I developed Insight, I have had the chance to engage with some of Silicon Valley's best data scientists who are mentors to the program:

Jonathan Goldman created one of the first data products at LinkedIn—People You May Know—which transformed the growth trajectory of the company. DJ Patil built and grew the data science team at LinkedIn into a powerhouse and co-coined the term "Data Scientist." Riley Newman worked on developing product analytics that was instrumental in Airbnb's growth. Jace Kohlmeier led the data team at Khan Academy that helped optimize learning for millions of students.

Unfortunately, it's hard to get face-to-face time with these remarkable people. At Insight, to maintain an exceptionally high quality and personal time with these mentors, we select only a small group of talented scientists and engineers three times per year.

However, The Data Science Handbook provides readers with a way to have these in-depth conversation at scale.

By reading the interviews in The Data Science Handbook, you will have the experience of learning from the leaders in data science at your own pace, no matter where you are in the world. Each interview is an in-depth conversation, covering the personal stories of these data scientists from their initial experiences that helped them find their own path to a career in data science.

It's not just the early data science leaders who can have a big impact on the field. There is also new talent entering, with the opportunity for each and every new member to push the field forward. When I met the authors of this book, they were still college students and aspiring data scientists, full of the same questions that those beginning in data science have.

Through 18 months of hard work, they have done the legwork in seeking out some of the best data scientists around the country, and asking them for their advice and guidance. This book is the result of that work, containing over 100 hours of collected wisdom with people otherwise inaccessible to most of us (imagine having to compete with President Obama to talk with DJ Patil!).

By reading these extended, informal interviews, you will get to sit down with industry trailblazers like DJ Patil, Jonathan Goldman and Pete Skomoroch, who were all part of the early, core LinkedIn data science teams. You will meet with Hilary Mason and Drew Conway, who were instrumental in creating the thriving New York data science community. You will hear advice from the next generation of data science leaders, like Diane Wu and Chris Moody, both Insight Alumni, who are now blazing new trails at MetaMinds and Stitch Fix.

You will meet data scientists who are having a big impact in academia, including Bradley Voytek from UC San Diego and Joe Blitzstein from Harvard. You will meet data scientists

in startups, such as Clare Corthell from Mattermark and Kunal Punera of Bento Labs, who will share how they use data science as a core competitive advantage.

The data scientists in the Data Science Handbook, along with dozens of others, have helped create the very industry that is now having such a tremendous impact on the world. Here in this book, they discuss the mindset that allowed them to create this industry, address misconceptions about the field, share stories of specific challenges and victories, and talk about what they look for when building their teams.

I hope that by reading their stories, hearing how they think, and learning their vision for the future of data science, you will come to think of ways you can have an impact, and perhaps even advance the field yourself.

Jake Klamka
Founder
Insight Data Science Fellows Program
Insight Data Engineering Fellows Program
Insight Health Data Science Fellows Program

INTRODUCTION

Welcome to The Data Science Handbook!

In the following pages, you will find in-depth interviews with 25 remarkable data scientists. They hail from a wide selection of backgrounds, disciplines, and industries. Some of them, like DJ Patil and Hilary Mason, were part of the trailblazing wave of data scientists who catapulted the field into national attention. Others are at the start of their careers, such as Clare Corthell, who made her own path to data science by creating the Open Source Data Science Masters, a self-guided curriculum built on freely available internet resources.

How We Hope You Can Use This Book

In assembling this book, we wanted to create something that could both last the test of time as well as address your interest in data science no matter what background you may have. We crafted our book so that it can be something you come back to again and again, to re-read at different stages in your career as a data professional.

Below, we've listed the knowledge our book can offer. While each interview is fascinating in its own right, and covers a large portion of the knowledge spectrum, we've highlighted a few interviews to give you a quick start:

- **As an aspiring data scientist** - you'll find concrete examples and advice of how to transition into the industry.
    - *Suggested interviews:* William Chen, Clare Corthell, Diane Wu
- **As a working data scientist** - you'll find suggestions on how to become more effective and grow in your career.
    - *Suggested interviews:* Josh Wills, Kunal Punera, Jace Kohlmeier
- **As a leader of a data science team** - you'll find time-tested advice on how to hire other data scientists, build a team, and work with product and engineering.
    - *Suggested interviews:* Riley Newman, John Foreman, Kevin Novak
- **As an entrepreneur or business owner** - you'll find insights on the future of data science and the opportunities on the horizon.
    - *Suggested interviews:* Sean Gourley, Jonathan Goldman, Luis Sanchez
- **As a data-curious citizen** - you'll find narratives and histories of the field, from some of the first data pioneers.
    - *Suggested interviews:* DJ Patil, Hilary Mason, Drew Conway, Pete Skomoroch

In collecting, curating and editing these interviews, we focused on having a deep and stimulating conversation with each data scientist. Much of what's inside is being told publicly for the first time. You'll hear about their personal backgrounds, worldviews, career trajectories and life advice.

In the following pages, you'll learn how these data scientists navigated questions such as:

- Why is data science so important in today's world and economy?
- How does one master the triple disciplines of programming, statistics and domain expertise to become an effective data scientist?
- How do you transition from academia, or other fields, to a position in data science?
- What separates the work of a data scientists from a statistician, and a software engineer? How can they work together?
- What should you look for when evaluating data science roles at companies?
- What does it take to build an effective data science team?
- What mindsets, techniques and skills distinguishes a great data scientist from the merely good?
- What lies in the future for data science?

After you read these interviews, we hope that you will see the road to becoming a data scientist is as diverse and varied as the discipline itself. Good luck on your own journey, and and feel free to get in touch with us at contact@thedatasciencehandbook.com!

— Carl, Henry, William and Max

DJ PATIL VP of Product at RelateIQ

The Importance of Taking Chances and Giving Back

DJ Patil is co-coiner of the term 'Data Scientist' and co-author of the Harvard Business Review article: "Data Scientist: Sexiest Job of the 21st Century."

Fascinated by math at an early age, DJ completed a B.A. in Mathematics at University of California, San Diego and a PhD in Applied Mathematics at University of Maryland where he studied nonlinear dynamics, chaos theory, and complexity. Before joining the tech world, he did nearly a decade of research in meteorology, and consulted for the Department of Defense and Department of Energy. During his tech career, DJ has worked at eBay as a Principal Architect and Research Scientist, and at LinkedIn as Head of Data Products, where he co-coined the term "Data Scientist" with Jeff Hammerbacher and built one of the premier data science teams. He is now VP of Product at RelateIQ, a next generation, data-driven customer relationship management (CRM) software. Most recently RelateIQ was acquired by Salesforce.com for its novel data science technology.

In his interview, DJ talks about the importance of taking chances, seeking accelerations in learning, working on teams, rekindling curiosity, and giving back to the community that invests in you.

Since we interviewed him, DJ has gone on to be appointed by President Barack Obama as the first United States Chief Data Scientist.

Something that touched a lot of people from your presentations is your speech on failure. It's surprising to see someone as accomplished as yourself talk about failure. Can you tell us a bit more about that?

Something most people struggle with when starting their career is how they enter the job market correctly. The first role you have places you in a "box" that other people use to infer what skills you have. If you enter as a salesperson you're into sales, if you enter as a media person you're into media, if you enter as a product person you're into products etc. Certain boxes make more sense to transition in or out of than other ones.

The academic box is a tough one because automatically, by definition, you're an academic. The question is: Where do you go from there? How do you jump into a different box? I think we have a challenge that people and organizations like to hire others like

themselves. For example, at Ayasdi (a topological machine learning company) there's a disproportionate amount of mathematicians and a surprising number of topologists.

For most people who come from academia, the first step is that someone has to take a risk on you. Expect that you're going to have to talk to lots and lots of people. It took me 6 months before eBay took a chance on me. Nobody just discovers you at a cafe and says "Hey, by the way you're writing on that piece of napkin, you must be smart!" That's not how it works, you must put yourself in positions where somebody can actually take a risk on you, before they can give you that opportunity.

Nobody just discovers you at a cafe and says "Hey, by the way you're writing on that piece of napkin, you must be smart!" That's not how it works, you must put yourself in positions where somebody can actually take a risk on you, before they can give you that opportunity.

And to do that, you must have failed many times, to the point where some people are not willing to take a risk on you. You don't get your lucky break without seeing a lot of people slamming doors in your face. Also, it's not like the way that you describe yourself is staying the same; your description is changing and evolving every time you talk to someone. You are doing data science in that way. You're iterating on how you are presenting yourself and you're trying to figure out what works.

Finally someone takes a chance on you, but once you've found somebody, the question is how do you set yourself up for success once you get in? I think one of the great things about data science is it's ambiguous enough now, so that a lot of people with extra training fit the mold naturally. People say, "Hey, sure you can be a data scientist! Maybe your coding isn't software engineering quality coding, but your ability to learn about a problem and apply these other tools is fantastic."

Nobody in the company actually knows what these tools are supposed to be, so you get to figure it out. It gives you latitude. The book isn't written yet, so it's really exciting.

What would you suggest as the first step to putting yourself out there and figuring out what one should know? How does one first demonstrate one's value?

It first starts by proving you can do something, that you can make something.

I tell every graduate student to do the following exercise: when I was a grad student I went around to my whole department and said, "I want to be a mathematician. When I say the word mathematician, what does that mean to you? What must every mathematician know?"

I did it, and the answers I got were all different. What the hell was I supposed to do? No one had a clear definition of what a mathematician is! But I thought, there must be some underlying basis. Of course, there's a common denominator that many people came from. I said, okay, there seem to be about three or four different segmentations. The segmentation I thought was the most important was the segmentation that gave you the best optionality to change if it ended up being a bad idea.

As a result of that, I took a lot of differential equations classes, and a bunch of probability classes, even though that wasn't my thing. I audited classes, I knew how to code, I was learning a lot about physics — I did everything I could that was going to translate to something that I could do more broadly.

Many people who come out of academia are very one-dimensional. They haven't proven that they can make anything, all they've proven is that they can study something that nobody (except maybe their advisor and their advisor's past two students) cares about. That's a mistake in my opinion. During that time, you can solve that hard PhD caliber problem AND develop other skills.

For example, aside from your time in the lab, you can be out interacting with people, going to lectures that add value, attending hackathons, learning how to build things. It's the same reason that we don't tell someone, "First, you have to do research and then you learn to give a talk." These things happen together. One amplifies the other.

It first starts by proving you can do something, that you can make something.

So my argument is that people right now don't know how to make things. And once you make it, you must also be able to tell the story, to create a narrative around why you made it.

With that comes the other thing that most academics are not good at. They like to tell you, rather than listen to you, so they don't actually listen to the problem. In academia, the first thing you do is sit at your desk and then close the door. There's no door anywhere in Silicon Valley; you're out on the open floor. These people are very much culture shocked when people tell them, "No you must be working, collaborating, engaging, fighting, debating, rather than hiding behind the desk and the door."

I think that's just lacking in the training, and where academia fails people. They don't get a chance to work in teams; they don't work in groups.

Undergrad education, however is undergoing some radical transformations. We're seeing that shift if you just compare the amount of hackathons, collaboration, team projects

that exist today versus a few years ago. It's really about getting people trained and ready for the work force. The Masters students do some of that as well but the PhDs do not. I think it's because many academics are interested in training replicas of themselves rather than doing what's right for society and giving people the optionality as individuals to make choices.

How does collaboration change from academic graduate programs to working in industry?

People make a mistake by forgetting that data science is a team sport. People might point to people like me or Hammerbacher or Hilary or Peter Norvig and they say, oh look at these people! It's false, it's totally false, there's not one single data scientist that does it all on their own. data science is a team sport, somebody has to bring the data together, somebody has to move it, someone needs to analyse it, someone needs to be there to bounce ideas around.

People make a mistake by forgetting that data science is a team sport.

Jeff couldn't have done this without the rest of the infrastructure team at Facebook, the team he helped put together. There are dozens and dozens of people that I could not have done it without, and that's true for everyone! Because it's a bit like academia, people see data scientists as solo hunters. That's a false representation, largely because of media and the way things get interpreted.

Do you think there's going to be this evolution of people in data science who work for a few years, then take those skills and then apply them to all sorts of different problem domains, like in civics, education and health care?

I think it's the beginning of a trend. I hope it becomes one. Datakind is one of the first examples of that, and so is data science for Social Good. One of the ones that's personally close to my heart is something called Crisis Text Line. It comes out of DoSomething.org — they started this really clever texting campaign as a suicide prevention hotline and the result is we started getting these text messages that were just heart wrenching.

There were calls that said "I've been raped by my father," "I'm going to cut myself," "I'm going to take pills," really just tragic stuff. Most teens nowadays do not interact by voice - calling is tough but texting is easy. The amount of information that is going back and forth between people who need help and people who can provide help through Crisis Text Line is astonishing.

How do we do it? How does it happen? There are some very clever data scientists there who are drawn to working on this because of its mission, which is to help teens in crisis.

There's a bunch of technology that is allowing us to do things that couldn't be done five, six years ago because you'd need this big heavyweight technology that cost a lot of money. Today, you can just spin up your favorite technology stack and get going.

These guys are doing phenomenal work. They are literally saving lives. The sophistication that I see from such a small organization in terms of their dashboards rivals some of the much bigger, well-funded types of places. This is because they're good at it. They have access to the technology, they have the brain power. We have people jumping in who want to help, and we're seeing this as not just a data science thing but as a generational thing where all technologists are willing to help each other as long as it's for a great mission.

Jennifer Aaker just wrote about this in a *New York Times* op-ed piece — that the millennial generation is much more mission driven. What defines happiness for them is the ability to help others. I think that there is a fundamental shift happening. In my generation it's ruled by empathy. In your generation, it's about compassion. The difference between empathy and compassion is big. Empathy is understanding the pain. Compassion is about taking away the pain away from others, it's about solving the problem. That small subtle shift is the difference between a data scientist that can tell you what the graph is doing versus telling you what action you need to do from the insight. That's a force multiplier by definition.

Compassion is also critical for designing beautiful and intuitive products, by solving the pain of the user. Is that how you chose to work in product, as the embodiment of data?

I think the first thing that people don't recognize is that there are a number of people who have started very hard things who also have very deep technical backgrounds.

Take Fry's Electronics for example. John Fry, the founder, is a mathematician. He built a whole castle for one of the mathematical associations out in Morgan Hill, that's how much of patron of the arts he is for them. Then you can look at Reed Hastings of Netflix, he's a mathematician. My father and his generation, all of the old Silicon Valley crew were all hardcore scientists. I think it just goes on to show - you look in these odd places and you see things you would not have guessed.

I think there's two roles that have been interesting to me in companies: the first is you're starting something from scratch and the second is you're in product. Why those two roles? If you start the company you're in product by definition, and if you're in product you're making. It's about physically making something. Then the question is, how do you make? There's a lot of ways and weapons you can use to your advantage. People

say there is market assessment, you can do this detailed market assessment, you can identify a gap in the market right there and hit it.

There's marketing products, where you build something and put a lot of whizbang marketing, and the marketing does phenomenally. There are engineering products which are just wow — you can say this is just so well engineered, this is phenomenal, nobody can understand it, but it's great, pure, raw engineering. There is designing products, creating something beautifully. And then, there's data.

The type of person I like best is the one who has two strong suits in these domains, not just one. Mine, personally, are user experience (UX) and data. Why user experience and data? Most people say you have to be one or the other, and that didn't make sense to me because the best ways to solve data problems are often with UX. Sometimes, you can be very clever with a UX problem by surfacing data in a very unique way.

For example, People You May Know (a viral feature at LinkedIn that connected the social graph between professionals) solved a design problem through data. You would join the site, and it would recommend people to you as you onboard on the website. But People You May Know feels creepy if the results are too good, even it it was just a natural result of an algorithm called triangle closing. They'd ask, "How do you know that? I just met this person!" To fix this, you could say something like "You both know Jake." Then it's obvious. It's a very simplistic design element that fixes the data problem. My belief is that by bringing any two elements together, it's no longer a world of one.

Because of the pace at which the world changes, the only way to prepare yourself is by having that dynamic range.

Another way to say this is, how do you create versatility? How do you make people with dynamic range, which is the ability to be useful in many different contexts? The assumption is our careers are naturally changing at a faster rate than we've ever seen them change before. Look at the pace at which things are being disrupted. It's astonishing. When I first got here eBay was the crazy place to be and now they're on a turnaround. Yahoo went from being the mammoth place to now attempting a turnaround. We've had companies that just totally disappeared.

I see a spectrum of billion dollar companies coming and going. We're seeing something very radical happening. Think about Microsoft. Who wouldn't have killed for a role in Microsoft ten years ago? It was a no brainer. But not anymore.

Because of the pace at which the world changes, the only way to prepare yourself is by

having that dynamic range. I think what we're realizing also is that different things give you different elements of dynamic range. Right now data is one of those because it's so scarce. People are getting the fact that this is happening. It gives a disproportionate advantage to those who are data savvy.

You mentioned earlier that when you were looking to become a mathematician you picked a path that optimized for optionality. As a data scientist, what type of skills should one be building to expand or broaden their versatility?

I think what data gives you is a unique excuse to interact with many different functions of a business. As a result, you tend to be more in the center and that means you get to understand what lots of different functions are, what other people do, how you can interact with them. In other words, you're constantly in the fight rather than being relegated to the bench. So you get a lot of time on the field. That's what changes things.

One of the first things I tell new data scientists when they get into the organization is that they better be the first ones in the building and the last ones out.

The part here I think people often miss is that they don't know how much work this is. Take an example from RelateIQ. I'm in the product role (although they say I'm supposed to be the head of product here, I think of these things as team sports and that we're all in it together), and I work over a hundred hours a week easily. If I had more time I'd go for longer hours. I think one of the things that people don't recognize is how much net time you just have to put in. It doesn't matter how old you are or how good you are, you have to put in your time.

You're not putting in your time because of some mythical ten thousand hours thing (I don't buy that argument at all, I think it's false because it assumes linear serial learning rather than parallelized learning that accelerates). You put in your time because you can learn a lot more about disparate things that fit into the puzzle together. It's like a stew, it only becomes good if it's been simmering for long time.

One of the first things I tell new data scientists when they get into the organization is that they better be the first ones in the building and the last ones out. If that means four hours of sleep, get used to it. It's going to be that way for the first six months, probably a year plus.

That's how you accelerate on the learning curve. Once you get in there, you're in the conversations. You want to be in those conversations where people are suffering at two in the morning. You're worn down. They are worn down. All your emotional barriers come down and now you're really bonding. There's a reason they put Navy Seals through

training hell. They don't put them in hell during their first firefight. You go into a firefight completely unprepared and you die. You make them bond before the firefight so you can rely on each other and increase their probability of survival in the firefight. It's not about bonding during the firefight, it's about bonding before.

That's what I would say about the people you talked to at any of the good data places. They've been working 10x harder than most places, because it is do or die. As a result, they have learned through many iterations. That's what makes them good.

What can you do on a day-to-day basis that can make you a good data scientist?

I don't think we know. I don't think we have enough data on it. I don't think there's enough clarity on what works well and what doesn't work well. I think you can definitely say some things increase the probability of personal success. That's not just about data science, it's about listening hard, being a good team player, picking up trash, making sure balls don't get dropped, taking things off people's plates, being there for the team rather than as an individual, and focusing on delivering value for somebody or something.

If you watch kids running around a track, and the parents want to leave, the kids always answer, "One more! One more!" You watch an adult run laps, and they are thinking, "How many more do I have to do?"

When you do that, you have a customer (could be internal, external, anybody). I think that's what gives you the lift. Besides the usual skills, the other thing that's really important is the ability to make, storytell, and create narratives. Also, never losing the feeling of passion and curiosity.

I think people that go into academia early, go in with passion. You know that moment when you hear a lecture about something, and you're saying, "Wow! That was mind blowing!" That moment on campus when you're saying, "Holy crap, I never saw it coming." Why do we lose that?

Here is a similar analogy. If you watch kids running around a track, and the parents want to leave, the kids always answer, "One more! One more!" You watch an adult run laps, and they are thinking, "How many more do I have to do?" You count down the minutes to the workout, instead of saying, "Wow, that was awesome!"

I feel that once you flip from one to the other you've lost something inherently. You have to really fight hard to fill your day with things that are going to invigorate you on those fronts. One more conversation, one more fight, one more thing. When you find those

environments, that's rare. When you're around people who are constantly inspiring you with tidbits of information, I feel like that's when you're lucky.

Is all learning the same? What value can you bring as a young data scientist to people who have more knowledge than yourself?

There's a difference between knowledge and wisdom. I think that's one of the classic challenges with academia. You can take a high school kid who can build an app better than a person with a doctorate who works in algorithms, and it's because of their knowledge of the app ecosystem. Wisdom also goes the other way: if you're working on a very hard academic problem, you can look at it and say, "That's going to be $O(n^2)$".

I'm a firm believer in the apprentice model

I was very fortunate when I was at eBay, as I happened to get inserted in a team where there was a lot of wisdom. Even though eBay was moving very slowly in things we were doing, I was around a lot of people who had a disproportionate amount of wisdom, so I was the stupidest guy with the least amount of tours of duty. But at the same time, I was able to add value because I saw things in ways that they had never seen. So we had to figure out where that wisdom aligned and where it didn't.

The other side of that was at LinkedIn, when you're on that exponential curve trajectory with a company. People say, "Well you were only at the company for three plus years," but I happened to be there when it grew from couple hundred to a couple thousand people. Being in a place where you see that crazy trajectory is what gives you wisdom, and that's the type of thing that I think compounds massively.

Many young people today are confronted with this problem related to knowledge and wisdom. They have to decide: Do they do what they're deeply passionate about in the field they care most about? Or do they do the route that provides them with the most immediate amount of growth? Do they go compound the knowledge of skills, or do they build wisdom in that domain?

It's a good and classic conundrum. I've gone with it as a non-linear approach: you go where the world takes you. The way I think about it is, wherever you go, make sure you're around the best people in the world.

I'm a firm believer in the apprentice model, I was very fortunate that I got to train with people like James Yorke who coined with the term "chaos theory." I was around Sergey Brin's dad. I was around some really amazing people and their conversations are some of the most critical pieces of input in my life, I think I feel very grateful and fortunate to be

around these people. Being around people like Reid Hoffman, Jeff Weiner is what makes you good and that gives you wisdom.

So for that tradeoff, if you're going to be around somebody that's phenomenal at Google, great! If you're going to be around someone super phenomenal in the education system, great! Just make sure whatever you are doing, you're accelerating massively. The derivative of your momentum better be changing fast in the positive direction. It's all about derivatives.

What do you think about risk taking, and defining oneself?

Everyone needs to chart their own destiny. The only I thing I think is for certain is that as an individual, you get to ask the questions, and by asking the questions and interpreting the answers, you decide the narrative that is appropriate for you. If the narrative is wrong, it's your narrative to change. If you don't like what you're doing, you get to change it.

It may be ugly, maybe hard or painful but the best thing is when you're younger, you get to take crazy swings at bats that you don't get to take later on. I couldn't do half the stuff I was doing before, and I'm very envious of people who get to. And that's a part of life, there's the flip side of when you do have family, or responsibilities, that you're paying for that next generation. Your parents put a lot on the line to try to stay in a town with great schools, and they may not have taken the risk that they would've normally taken to do these things.

If the narrative is wrong, it's your narrative to change. If you don't like what you're doing, you get to change it.

That's part of the angle by which you play. It's also the angle which is the difference between what it means as an individual and team player. Sometimes you can't do the things that you want to do. It's one of the reasons I've become less technical. Take someone like Monica Rogati or Peter Skomoroch, two amazing data scientists and engineers at LinkedIn. What's a better use of my time? Taking a road block out of their way or me spending time debugging or coding something on my own?

In the role I have, in the position and what was expected of me, my job was to remove hurdles from people, my job was to construct the narrative to give other people runway to execute, their job was to execute and they did a hell of a good job at it.

You have talked about your research as a way to give back to the public that invested in you. Is there an aspect of the world that you feel like could really use

the talent and skills of data scientists to improve it for the better?

I think we're starting to see elements of it. The Crisis Text Line is a huge one. That's why I put a lot of my time and energy into that one. But there are so many others: national security, basic education, government, Code for America. I think about our environment, understanding weather, understanding those elements, I would love to see us tackle harder problems there.

Only work on simple things; simple things become hard, hard things become intractable.

It's hard to figure out how you can get involved in these things, they make it intentionally closed off. And that's one of the cool things about data, it is a vehicle to open things up. I fell into working on weather because the data was available and I said to myself, "I can do this!" As a result, you could say I was being a data scientist very early on by downloading all this crazy data and taking over the computers in the department. The data allowed me to become an expert in the weather, not because I spent years studying it, because I was playing around and that gave me the motivation to spend years studying it.

From rekindling curiosity, to exploring data, to exploring available venues, it seems like a common thread in your life is about maximizing your exposure to different opportunities. How do you choose what happens next?

You go where the barrier of entry is low. I don't like working on things where it's hard. My PhD advisor gave me a great lesson — he said only work on simple things; simple things become hard, hard things become intractable.

So work on simple things?

Just simple things.

HILARY MASON Founder at Fast Forward Labs

On Becoming a Successful Data Scientist

Hilary is the Founder of Fast Forward Labs, a machine intelligence research company, and the Data Scientist in Residence at Accel. Previously, she was the Chief Scientist at bitly, where she led a team that studied attention on the internet in realtime, doing a mix of research, exploration, and engineering. She also co-founded HackNY and DataGotham, and is a member of NYCResistor.

What do you do as a data scientist in residence?

I do three things. First, I occasionally help the partners talk through an interesting technology or company. Second, I work with companies in the Accel portfolio. I help them when they run into an interesting or challenging data question. Finally, I help Accel think through what the next generation of data companies might look like.

Do you expect this to be a growing trend, the fact that VC firms are hiring data scientists in residence?

We're at a point where there are very few people who've spent years building data science organizations in a company or building data-driven products. Having people with even just a few years of expertise in doing that is valuable.

I don't expect that this will be nearly as difficult in the future as it is now. Because data science is so new — there are only a few people who have been doing this for a long time. Therefore it really helps a VC firm to have access to someone who they can send to one of their companies when that company has some questions. Right now, the expertise is fairly hard to come by, but it's not impossible. In the coming years, I think more and more people will take this expertise for granted.

What can you tell our readers about the data community in New York City?

We're not a tech city. We are a city of finance, publishing, media, fashion, food and more. It's a city of everything else. We see data in everything here. We have people in New York

doing data work across every domain you can imagine. It's absolutely fascinating.

You'll see people who talk about their work in the Mayor's office, people talking about their academic work, people in health care using data to cure cancer, and people talking about journalism. You can see both startups and big companies all talking about how they use data.

DataGotham is our attempt to highlight this diversity. We started it as a public flag that we planted and said, "*Whatever you do, if you care about data, come here and meet other people who also feel the same way.*" I think we've done a good job with that. The best way to get a sense of New York's data community is to come.

How else do you think data science will change? What will happen to data science in the next five years?

Five years is a long time. If you think back five years, data science barely existed, and it's still evolving rapidly. It will change a lot in these next five. I'm not going to say what is certain to happen in the next five years, but I'll make a few guesses.

We see data in everything here. We have people in New York doing data work across every domain you can imagine. It's absolutely fascinating.

One change is that some of the delightful chaos will go away. I know fantastic data scientists who have degrees in computer science, physics, math, statistics, economics, psychology, political science, journalism and more. People have switched to data science with a passion and an interest. They didn't come from an academic program. That's already changing — you can enroll in Master's degree programs in data science now.

Perhaps some of the creativity that happens when you have people from so many different backgrounds will result in a more rigid understanding of what a data scientist actually is. That's both a good and bad thing.

The second change is, well, let's just say that if I'm still writing Java code in five years I'm going to punch a wall! Our tooling *has* to get a lot better, and it already is starting to. This is a fake prediction because I know things are already happening in this area.

Five years ago, the most interesting data companies were building infrastructure, different kinds of databases. They were working on special tools for managing time series data. Now, the base infrastructure is mature and we're seeing companies that are making it easier to work with those pieces of infrastructure. So you get a great dashboard

and you can plug in your queries, which go behind the scenes and run map-reduce jobs. You won't be spending 40 hours manually parallelizing algorithms and hating your life anymore. I think that will continue to expand.

Culture is also a big part of the practice. I think data culture will continue to grow, even among people who aren't data scientists. This means that within lots of companies, you will begin to see people whose job titles don't say "data scientist," but they will be doing very similar things. They won't need to ask a statistician to count something in a database anymore — they can do it themselves. That's exciting to me. I do believe that data gives people the power to make better decisions, so the more people who have access to it, the better.

How do you think the role of a data scientist will change in a world where every company has data-minded people?

Data scientists will keep asking the questions. It's not always entirely obvious what you should be counting, even for fairly trivial business problems. It's also not entirely obvious how to interpret the results. Data scientists can become the coach, the person who really understands the problem they're trying to solve.

Data scientists and data teams do a variety of things beyond just business intelligence. They also do algorithmic engineering, build new features, collect new data sets, and open up potential futures for the product or business. I don't think data scientists will be out of work anytime soon.

You emphasize communication and storytelling a lot when you talk about data science. Can you elaborate more on this?

A data scientist is someone who sits down with a question and gathers some data to answer it, or someone who starts with a data set and asks questions to learn more about it. They do some math, write some code, do the analysis, and then come to a conclusion. Then what?

They need to take what they've learned and communicate it to people who were not involved in the analytical process. Creating a story that's compelling and exciting for people, while still respecting the truth of the data, is hard to do. This skill gets neglected in many technical programs, as it's taken for granted that if you can do something you can explain it. However, I don't think it's that easy.

Why isn't it easy? Why is explaining something in a simple manner so difficult?

It's hard because it requires a lot of empathy. You have to understand something that's

very technical and complex, then explain it to someone who doesn't come from the same background. You have to know how they think so you can translate it into something they can understand. You also have to do it for people who generally have short attention spans, who are impatient, and who are not ready to spend hours studying.

I do believe that data gives people the power to make better decisions, so the more people who have access to it, the better.

So you need to come up with a solution that uses language or a visualization to facilitate their understanding after you've invested all of this time building a complex model. When you think about it, it's amazing that we can take our complex technical understanding of something and then write it down in such a short, concise way to communicate it to someone who doesn't share the same knowledge or interests. That's amazing.

When you think of it that way, it's not a surprise at all that storytelling is hard. It's like art. You're trying to take a really intense emotion or complex phenomenon and express it in a way that people will understand intuitively.

You've said before that some of the most exciting data science opportunities are in startups. Given your experience with Bitly and advising startups, can you elaborate more on that?

I'll explain with the disclaimer that I'm obviously slightly biased. The most exciting data opportunity is when you have the flexibility to collect data. Often you're collecting data accidentally as a side effect of another product you were trying to build.

Bitly is the classic example of this — short URLs make it easy to share on social networks. You end up collecting this amazing data set about what people are sharing and what people are clicking on across all these social networks. But nobody really set out in the beginning to build the world's greatest URL shortener to discover how popular Kim Kardashian is. Bitly's founder John Borthwick calls this accidental side effect "data exhaust," which is a lovely phrase for it.

That said, if you're in academia, you don't have the benefit of having a product there already collecting data. There's an extra project to do before you even do the work you actually care about. You have to struggle to collect your own data, or go to a company and beg for their data. That's really difficult, because most companies have no incentive to share data at all. In fact, they have a very strong disincentive given privacy liability. So, as an academic, you find yourself in a difficult position unless you're one of those people who are able to build good partnerships (which some people are).

If you're at a larger company, the data you have is probably either stuck in a bunch of incompatible databases or so highly controlled that it will take a huge political effort to get the data into a place where it becomes useful.

Startups are the perfect place where you have a product that's generating it's own data. As a data scientist you have input into how the product changes, so you can ask, "*Can we collect this other thing?*" or "*Do you think if we tried this we might learn something else?*" It's very open as to what you do with it.

I love that aspect that we can learn something interesting from the data. It's a fun process and a good place to be.

What advice would you give our readers who are interested in joining a data science startup? How should one choose where to work at?

Startups are the perfect place where you have a product that's generating it's own data.

Try to learn more about the startup culture. Startups generally have great cultures — one reason is because startups are much more free to have wide variability in those cultures. You'll find that some startups might be a great fit for you, while some of them might feel uncomfortable. There's nothing wrong with you, it's just a company that's not a good match.

This is just good advice in general. When you're looking at working in a small company, make sure it's a group of people that you're comfortable working with and that the social environment is one that you're going to feel happy and comfortable in.

That said, a lot of companies are hiring their first data scientists. Most data scientists have no experience in a job, so it's very hard to find someone who can come in and do a job well that nobody has done before. I would make sure that whoever you're working for — whether it's your COO, CTO or CEO — has a pretty clear understanding of what they want you to do. At least they should be someone you think you could collaborate with in figuring out where you should invest your time.

Can you elaborate more on prioritization and investing time?

You've got an infinite list of questions you can look into — how do you pick the ones that are going to have the biggest impact? How do you do that in an environment where you might have your CEO demanding slides for a board meeting, your head of sales demanding data, etc., and you have a project that you think is really exciting — but no

one else quite gets it yet because they haven't really sat with you and gone into the data?

If you're looking for your first job as a data scientist, I would make sure you have a manager who can manage that process with you. If you're going to be that manager, it's not as easy as it looks from the outside. That is a skill you have to develop. If you're going to be a manager, I'd recommend that you think about those sets of problems -- how to process them and how to communicate them in a way that fits with the process that the rest of the company is using.

What other advice do you have?

Look for good data sets. When I interview people for a data science job, they will already have spent a few hours with people on the team. I'll say, 'Y*ou know what we do now. What is the first thing that comes to your mind when you're thinking 'why haven't these guys even thought about this?'"* I don't really care what the answer is, but I want to know that they're capable of thinking about what the data set is and coming up with ideas on their own for what they would like to see.

Most of the answers I've have to that question were things we had already thought of. I don't expect people to come up with genius ideas in the interview, but just to show that they have that creative ability can be really helpful. If you're looking at a company or product to potentially work for and you can't come up with things you would want to work on, that's a problem. You should find something you're a little more excited about.

Do you have more advice on prioritization and making an impact within a company?

During my time at Bitly and in general, we have a series of questions we ask about every data project we work on. The questions would help not just with personal prioritization but also with helping other people in the company understand what was going on.

You've got an infinite list of questions you can look into — how do you pick the ones that are going to have the biggest impact?

The first question is, can we define the question we're interested in? You'd think it would be obvious that it's helpful to write down the question in plain language so that anyone can understand what you're trying to do.

The second question is, how do we know when we've won? What are the error metrics by which we evaluate our solution to this question? If we're working on an algorithm where there are no quantitative error metrics, you at least have to write down that there are none.

The third question is, assuming we can solve this perfectly, what's the first thing we will do with it? I ask that question to ensure that every project is immediately relevant to the business or product. It's not just an irrelevant exercise because we're curious about something. The first thing you'll do with it should also have some longer term implications about what you understand about the data.

For each data project you're working on, you need to ask yourself these questions: what are you working on? How will I know when it's done? What does it impact? If you ask yourself these questions, you always know you're making a good decision about how you're spending your time.

Do you have an example of using these questions to understand a project?

One project you might be working on might be, "*Does our user behavior in Turkey differ from user behavior in the United States?*" That might be an immediately relevant question, maybe because of a sales deal with someone in Turkey.

The longer term goal would be to understand if geography affects user behaviour, and if so, how? You should always be balancing those near-term and long-term rewards, building your library of information of what you know from your data.

The last question is, assuming that everything works perfectly and everyone in the world uses our solution, how does it change human behavior? That question is important because I want to make sure that people are always working on the highest-impact problems.

Another question I ask sometimes is, what is the most evil thing that could be done with this? If I were an evil mad scientist in my volcano lair and I had this technology or knowledge, what could I do with it? You get way more creative ideas for what to actually do with it, very few of which are evil. That's a fun thought experiment to do.

You've given great advice on how data scientists can choose a startup. I wanted to flip that question around — what general advice would you give to new startups that are building their data science team?

This is always a challenge, and often, people have different ideas of what a data scientist coming into the company will do. So this means that first the founders and management team should really understand what they need now.

You're sure that you want some business analytics, product analytics, and metrics. Maybe you have an idea to do something cool with the data — perhaps something that's

well understood like a recommendation engine, or maybe even something that's more creative. But it's hard to find someone who can do all of these things and potentially can grow to manage a team of people.

The things you can do when you're hiring is look for people who learn quickly, are really creative, are flexible, and who can work with your engineering team because that's where they're going to sit. They need to be best friends with whoever is running the infrastructure that holds the data, and they need to be able to work with the product and business side as well.

For each data project you're working on, you need to ask yourself these questions: what are you working on? How will I know when it's done? What does it impact?

That means that you might want to hire somebody who doesn't have 20 years of data experience but who you think can learn really quickly and grow with the product, with the understanding for that person that eventually a team might come around them or they might hire a manager.

So much of hiring well in small companies is finding the right person at the right time for that company. There's no one formula that really describes it — it has to be a good match on both sides.

What advice do you have for students who are choosing between smaller companies and larger companies?

I would say it's worth looking at the smaller companies. The advice I have there is find someone who you'll work for who you think would be a great mentor for a year. Don't just go to a small company because it sounds good. Go to one where you think, "*This is somewhere I can learn from for a year. I think I'll be happy here for about that long.*"

Then after a year, you can re-evaluate. Am I still learning? Am I doing work that I love? And if not, you can move on to your next learning opportunity. But the first few years out of school will help you learn the skills you'll need later. Go to places where you can learn things. That's the best way to think about it.

What other advice do you have for students choosing between companies?

I know when you look at job offers, it's really easy to evaluate them based on how much money you're going to make and where you're going to live. I'm a big fan of living somewhere you like, because otherwise you're miserable all the time, because it's not all about the money. It's most important to be working in an environment where you have

challenging work with people you can learn from.

For example, I once did an internship in AT&T Labs Research, and I loved working there. It was an amazing place full of really amazing people. But I hated living in New Jersey and commuting on the Garden State Parkway. You need to find that right balance of making sure you're in a place where you're going to be happy, but also learning a lot.

Whether you're making 10 or 20 grand more now, versus years later, it doesn't make a difference. As long as you're making enough to have a decent place to live, eat well, enjoy your life when you're not at work, I wouldn't pay too much attention to the salary.

What advice would you give to aspirational data scientists?

A lot of people are afraid to get started because they're afraid they're going to do something stupid and people will make fun of them. Yes, you will do something stupid, but people are actually nicer than you think and the ones who make fun of you don't matter.

My recommendation is that if you're interested in data science, try it! There are a lot of data sets out there. I have a Bitly list of about 100 public research-quality datasets, which you can see here: bitly.com/bundles/hmason/1. You also have access to a bunch of public APIs. You can be creative.

Go to places where you can learn things.

Try to do a project that plays to your strengths. In general, I divide the work of a data scientist into three buckets: Stats, Code, and Storytelling/Visualization. Whichever one of those you're best at, do a project that highlights that strength. Then, do a project using whichever one of those you're worst at. This helps you grow, learn something new, and figure out what you need to learn next. Keep going from there.

This has a bunch of advantages. For one thing, you know what data science is actually like. A lot of data scientists spend their time cleaning data and writing Hadoop scripts. It's not all fun — you should experience that.

Second, it gives you something to show people. You can tell people what cool things you're trying out — people get really excited about that. They're not going to say you tried and you suck, they're going to say, "Wow, you actually did something. That's cool!" This can help you get a job.

A great example of this is my friend Hilary Parker who works at Etsy on their analytics

team. Before she got the job there, she did this fantastic analysis of how Hilary is the most poisoned baby name in U.S. history. The popularity of the name Hilary was growing until Bill Clinton got elected, when it just plummeted. Slowly now it's getting more and more popular again (obviously I love this example because my name is also Hilary). She put it on her blog and ended up getting published in *New York Magazine* — I believe it really helped her land a job by showing that she really knew what she was doing.

I really just encourage people to start putting things up on their blogs and on Github, and not to be discouraged. It takes optimism and stubbornness to do this well.

PETE SKOMOROCH

Principal Data Scientist at Data Wrangling

Software is Eating the World, and It's Replacing it With Data

Ever since he was young, Pete Skomoroch was interested in science. This led him to double major in mathematics and physics at Brandeis University, where he discovered he enjoyed tinkering with mathematical models and engineering. After graduating, Pete honed his technical skills at Juice Analytics, MIT Lincoln Laboratory and AOL Search.

Pete eventually ended up as a Principal Data Scientist at LinkedIn, where he led teams of Data Scientists focused on Reputation, Inferred Identity and Data Products. He was lead Data Scientist and creator of LinkedIn Skills & Endorsements, one of the fastest growing new products in LinkedIn's history.

He is also the founder of Data Wrangling, which offers consulting services for data mining and predictive analytics.

You're one of the people who've been around data science since the beginning. How have you seen it evolve?

The creation of the data scientist role was originally intended to address some challenges at large social networks. Many software companies at the time had separate teams. There were production engineers, research scientists writing papers and developing prototypes, and data analysts working with offline data warehouses. The classic R&D model required a lot of overhead as ideas were passed from one team to another to be re-implemented. The latency to get an idea into production and iteratively improve it in this way was too high, especially for startups.

The data scientist role was intended to bridge the gap between theory and practice by having scientists who could write code and collaborate with engineering teams to build new product features and systems. At LinkedIn, we wanted to hire scientists and engineers who could develop products and work with large production datasets, not just hand off prototypes. I think the original concept has evolved over the last few years as organizations found it difficult to hire candidates with the full skill set. Simultaneously, as data science became more popular, it evolved into an umbrella term that describes a large number of very different roles. In my case, I was a Research Engineer at AOL

Search and was originally hired as a Research Scientist at LinkedIn before my job title was changed to Data Scientist. In the following years, many business analysts and statisticians also rebranded as data scientists.

Today, depending on the company, a data scientist could be a person who fits that original hybrid scientist-engineer role, or they could be statisticians, business analysts, research scientists, infrastructure engineers, marketers, or data visualization experts. In some organizations, things have come full circle as these skills are held by separate specialized individuals that work together on a data team.

There is nothing wrong with any of these roles and you need all of them for a large modern organization to get the most out of data. That said, I think there is value in having people who fit the original definition, who are interdisciplinary, and can cross boundaries to build new products and platforms. Confusion often arises when companies either don't know which type of role they need for their organization or which type of data scientist they are interviewing.

What [Jeff Hammerbacher] really wanted on his team was a MacGyver of Data Analysis who could work with data, write code in Java and actually implement the algorithms, do some statistics, and really have a good intuition of what would drive strategic objectives.

Can you talk about your story, and how you ended up where you are?

I was really interested in science from an early age. When I started at LinkedIn, I was a research scientist, and before that, I had been a research engineer at AOL Search. The flavor of that role was more like the R&D labs that were doing machine learning research and crunching search query data, but there was a strong pull for us to do more production coding involving product.

I remember a talk that Jeff Hammerbacher gave in which he mentioned that what he really wanted on his team was a MacGyver of Data Analysis who could work with data, write code in Java and actually implement the algorithms, do some statistics, and really have a good intuition of what would drive strategic objectives.

I think that was the kernel of the idea that Data Scientist is a different role. When we are interviewing, we don't want to select for people who are just business analysts who can't code, and we don't want people who are pure engineers who don't have any science or math background. We want people at that intersection. I think that was really the genesis of data science, it is cross-disciplinary.

Some of your undergrad research was about neuroscience, can you tell us a bit more about that?

I was really interested in neuroscience, and physics and electronics. When I went to Brandeis, I found that I actually liked mathematical modeling, data crunching, cracking codes, building models and programming versus doing lots of bio lab work. I felt my real aptitude was digging into the data and coming up with theoretical models, which is what drew me to physics.

When giving advice on undergraduate coursework... [I'd say] take as many physics and math classes as you can, but also learn some computer science.

I graduated college in 2000 while the dotcom boom was still happening. My family was just scraping by financially, so it was really compelling for me to go into industry although I ultimately planned to go back to grad school. I had used Matlab, Mathematica, some C, and Assembly in physics classes and learned Visual Basic in an internship, but I wasn't a strong programmer at that point. In retrospect, that is one thing I would have done differently in undergrad. If I had taken more computer science classes, I probably would have ramped up faster at startups.

When giving advice on undergraduate coursework, I'd echo Yann Lecun, who is now heading AI Research at Facebook and did pioneering work in neural networks. I agree with his advice to take as many physics and math classes as you can, but also learn some computer science.

How did computer science play into your post-college job?

A big piece of what a data scientist is really doing is creating models. It's not just about taking data and loading into a black box machine learning algorithm and running it, but actually modeling something about an organization, a company or a product. It's difficult to find the underlying factors and phenomena that are really predictive and prescriptive vs. something that is just a correlation.

So, when I was looking at jobs coming out of college in 2000, I interviewed at a few places, and one that looked really interesting was a small startup in Kendall Square called Technology Strategy Inc., which eventually rebranded as ProfitLogic, Inc. Our early clients included casinos and some of my coworkers were working on interesting projects optimizing slot machines or spotting cheaters. In the early days we did a lot of consulting work and as it turned out there was a lot of interest from fashion retailers, who wanted things like better inventory allocation and markdown price optimization.

What we were doing was essentially an early version of data science. We would get tapes delivered weekly from big retailers like Macy's or JC Penny or Walmart, and the data would be loaded into our own data warehouses. Then we would run statistical models using a combination of C++ and Python to adjust prices and build predictive sales forecasts at the item level. The ultimate idea was that you could save a lot of time and maximize profit by automatically setting prices using a data driven approach. By taking these optimal price trajectories instead of relying only on intuition, you could make more profit and get more inventory through the system.

My initial role there was similar to a grad student in a research lab. Eventually, I became a hybrid product manager and engineer on the data and algorithm side. I would often be in the office all night, making sure that the weekly model run was working, scrutinizing thousands of charts and logs for model issues. Over time, I started to see areas for improvement and develop my own algorithms for seasonality and other forecast improvements. I was working with people across the engineering teams, the database team and research scientists. That's where I first encountered this pain point of bridging between those areas.

In my case, what I found was that I needed to build up my programming and computer science skills to become more self sufficient. I started out as an analyst building models and then moved into the software engineering organization.

How did you get good at these things? Did you take your own time to learn, or is it more like you just embedded yourself within the groups at the company that you were doing these things at?

I think the only way to excel is to take the extra time. I would go home and read every O'Reilly book I could get my hands on, working through textbooks and side projects. I would do what I could to learn at work, and I was always pushing to work on areas beyond what I was doing before. I'd advise people to take the time to level up early on in their careers, maybe sleep a couple hours less while you can handle it.

The only way to level up was to do real coursework and be around people who were actually doing it.

As I was reading and building models, it seemed like machine learning was a better answer than heuristics or other approaches commonly used in forecast models. I was learning that on my own, but I felt like the only way to level up was to do real coursework and be around people who were actually doing it. There was a job opportunity at MIT's Lincoln Lab working in biodefense, and a big benefit for me was that I could also take graduate courses in that role. I took a fantastic neural networks course with Sebastian

Seung, the author of Connectome, and a machine learning course with Leslie Kaelbling, along with some math courses and an optimization theory course.

My story during that time period is a bit of an unusual one. I would often wake up, go to work in Lexington, go to the MIT library, stay up all night eating from the vending machines and working on problem sets, and go back to work the next day without sleeping. Then I would go home and crash, and then I would repeat that process. I was a zombie for a couple of years and if I could do anything differently, I would balance that much better. Yes, you have to put in your time, but try to balance it. Staying up all night coding is the same thing. Sometimes you maybe have to do it but if you're doing it all the time, you are eventually going to burn out and you are nowhere near as effective as you think you are.

That said, I don't want to make it seem like there is a magic path through this. To get to the point where you can gain the right skills this field does take a lot of hard work and I wouldn't minimize that.

The amount of stuff you have done is unbelievable. I think telling the story of how hard everything was, it's not that you had everything handed to you. That is critical in communicating how people think.

I think there are two parts. Being smart only gets you so far. You have to work hard because anything worth doing is worth doing well and you're better off just digging in. There is this psychological factor of grit that is important.

If you go into management, I advise not giving up coding completely. Own a feature or something that keeps you in the loop.

That is what I would encourage people to think about. Stretch yourself, because if you only work on things that you know well, you're going to plateau. That is part of what makes doing a new startup so appealing. If you go into management, I advise not giving up coding completely. Own a feature or something that keeps you in the loop, so that you're up to speed with the development tools, the build process, the code base, the latest tricks and languages. All these things are important because the further you get from the nuts and bolts, the harder it is to make intelligent decisions. The technology changes rapidly, especially in data science.

Can you talk about your experience at Lincoln Lab? What was it like, especially as you were moving there from the private sector?

There was a mixture of biologists, physicists, hardware engineers and software engineers.

I've always been drawn to the intersection of fields. One project involved a machine learned model for a biosensor. It started as a simple threshold alarm algorithm, and I took it a step further to mathematically model the biochemical processes statistically and apply machine learning on top of that parameterized model.

Anyway, I thought it was interesting that machine learning doesn't just have to be a black box. You can get better results if you have a more intuitive sense or physical sense of what you are modeling and build those features into the model. Often, a custom model is what you need to really nail it. On the other hand, if the answer only has to be 80% accurate, you may want to do something more lightweight.

Afterwards, I moved to DC while my wife was in grad school, but after a few years in defense I wanted to try a job in consumer internet. The most interesting role around DC in terms of machine learning at the time was at AOL Search. The experience working with large datasets at MIT helped me land a role on a great team there mining search query data, and many of my coworkers from that team went on to work at Twitter via the Summize acquisition. There were a lot of management changes at AOL during that time, and I did my best to adapt while things were uncertain, installing an early Hadoop cluster there and experimenting with mapreduce techniques.

There were all these interesting things developing around the same time in the startup world, including the early development of Amazon EC2 and Hadoop, and so I viewed that lack of direction as an opportunity. AOL was very much a content company and I wanted to look at how they could do better in terms of content based on data: Based on search data, what can we decipher about what people are actually interested in, what's trending? And so the first step is to assess, how are you doing versus your competitors? AOL grew through acquisitions, so it wasn't like everything was on a central system. I actually had to crawl internal AOL properties and external sites as well.

Externally, there were signs that data was going to be a big deal, but internally they were dismantling the R&D team, so I knew that wasn't a good place to stay. Another company that I had been talking to in the area was called Juice Analytics. They were primarily known for data visualization, but it was an appealing opportunity to me because I could apply this intersection of skills I'd been developing to product development. So I joined Juice, and we built and shipped a SaaS software product built on Django and EC2. It took about a year, and we were crunching search queries and doing some clustering and pattern recognition to come up with a better picture of your site's search topics instead of just the top ten queries or whatever you got at the time in Google Analytics. That was a great experience of end-to-end product development.

Ultimately, I think it was a failure in terms of product-market fit, but I learned a lot from

that process. As a data scientist in an engineering driven company, you probably go through engineering boot camp, get up to speed with the tech stack, and then you can actually do some engineering to solve your own data problems. When you think about it, that's the way you get leverage in the world that we live in now.

What do you mean when you talk about leverage?

Imagine you have an idea on how to improve your company's product. Say you come in and say, I have this great idea. Everybody will love it and it will make billions of dollars and improve the lives of millions of people. But if you are just describing the idea and you can't implement at least some rough version of it, you are at a disadvantage. That's why I think one of the highest leverage things you can do right now is gain some engineering and computer science skills.

So how did you move from Lincoln Lab to Silicon Valley?

After the experience at ProfitLogic, I was bit by the startup bug and ultimately planned to move out to California. After my wife completed her master's in 2009, we said okay, we're just going out there. The previous year in DC, I became increasingly active on Twitter and I found it really fantastic for finding people with similar interests, especially when you were outside the Bay Area. For data, one of the key people I met was named Mike Driscoll. He's the CEO at Metamarkets, but at the time he had a blog called Dataspora and he did data-related consulting. We contemplated doing an O'Reilly book back then called *Big Data Power Tools* to a) survey these different tools that you should know and b) offer case studies with tips and tricks for practitioners. My vision was that you would hand that book to a new hire and just have them read through it and be ready to hit the ground running. Fast forward to today, and it's really great to see that this is actually happening through a variety of courses, textbooks, meetups and data science bootcamps like the Insight Data Science Fellows program.

One of the highest leverage things you can do right now is gain some engineering and computer science skills.

I think that now a lot of large Fortune 500 companies see the success of consumer internet companies like Google, Facebook, Twitter, Amazon, etc., and they say, "I'm not sure what they are doing, but it seems to be working. I want that. How do I innovate and build products like that?" I think there is a bit of a misconception out there that building dashboards of business metrics like Google will turn you into Google, when really it was a huge amount of engineering infrastructure and algorithmic product development that got them to where they are today. I think a lot of the people who want to get into data

science say, “That is really amazing, how does Google know everything?”.

Or, perhaps “How does Target know I’m pregnant?”

That’s a darker version of that question, but even there it’s interesting to note that the algorithms were really just detecting people following instructions from other software systems. If you are pregnant, there are tons of websites and medical guides that tell you exactly what to purchase and which vitamins to take each week. When you know that, it’s not so surprising that such regimented purchase patterns are detectable.

That said, a lot of data science does seem like magic. How do they create these magical experiences? Even Uber seems like magic (I know that isn’t all necessarily data science), but there is something impressive about getting the cars there fast enough when you push a button that it feels like magic. Fortune 500 companies and big organizations want that magic. And they have some sense that it is happening through data, but they’re not quite sure how. I wasn’t sure either when I started in the field, but it was just clear to me that we were just scratching the surface of what we can do involving engineering and data.

What sort of opportunities did you find at LinkedIn that took advantage of your quantitative background?

The younger a company is, the easier it is to propose new things. When I started working there, LinkedIn had some structured data around titles and companies and company pages, but they didn’t really have any notion of topics or skills. I had just done a bunch of Wikipedia topic mining to build a site called trendingtopics.org, and I thought, with all of these member profiles, I should be able to do some topic mining of the skills that people have. And then I’ll have that structured data set. I thought you should be able to tag people like websites in del.icio.us (which I was a big fan of) and then we would have all this rich data to do better recommendations and matching.

I made a quick proposal to my manager DJ Patil, and I got a time window of six to seven weeks to crank out a prototype. This was back in 2009 and at first, I didn’t think that LinkedIn would have enough data in the connection graph to say how good somebody was at something. But even in early versions, there was a lot of signal in the data and the project was green lighted based on that prototype. At that point, my picture of where this thing was going evolved and I thought that the ultimate value was going to be in the reputation data tied to each skill.

What ultimately led to further enhancements like endorsements was the overarching goal to develop products that fulfilled strategic goals to get people back on the site,

grow engagement, grow profile data, and help improve job matching, ad matching, and other algorithms. The ultimate goal for me was to add a layer of links anchored by skills across profiles, and do for the social and professional graph what Google had done for web pages, allowing people to find and by found.

Can you talk more about what it's like developing new features or products at larger established companies, versus the startups you've worked at in the past?

There was a formal process to bringing new ideas to production at LinkedIn because there may be a big difference between the technologies you used to prototype your idea, and those that LinkedIn is built with. The same thing likely applies for any big tech company at this point. You have to get projects approved and they have to get a budget because you need specialized people on the projects in different organizations: web designers, web developers, frontend engineers, ops people. It takes more of cross-team village to build a product versus a startup where you are a small group wearing a bunch of different hats doing a bunch of different things.

The younger a company is, the easier it is to propose new things.

The spirit at the time during when we built the first version of skills was still that we would try to wear many hats. That said, we wanted to ship product quickly and the way to get that done is to get the right resources lined up so you can really execute. I think one of the worst things you can do is sign up for a project when you know you are not set up for success and you are not resourced properly.

Another important reality to face is that you need to hit product-market fit. You could have a very smart idea as a data scientist, but there is more to succeeding than just having a smart idea. One common problem is that the idea might not align with the company objectives. Another is that many startups that just fail because they are a technology in search of a problem. When you hear there is a shortage of data scientists, I actually believe the most difficult people to find are those that have a more human, intuitive sense of the customer and knack for getting to product-market fit.

How do you develop this "intuition" for product-market fit?

When I interview people, it often manifests itself in somebody who is driven and who has done some novel, creative side projects. When you are building stuff on your own, you often see that your original idea doesn't actually have enough thought put into it. I also like to see when people have worked either in different disciplines or in different areas of domain expertise. An example of a concrete question that would come up in an interview to test for this intuition would be: "If you had access to all of our data, what would you do?"

I think that rather than going from the bottom up and thinking, "What is something cool I can do with this data?" it is sometimes a better approach to think strategically from the top down. "What are the top priorities for this company and what are we going after? What technology or product trends are opening up new opportunities? Who are our customers? What is the market, and how could I do this differently with data?"

This seems to capture the sentiment expressed by Steve Jobs when he said, "People think focus means saying yes to the thing you've got to focus on. But that's not what it means at all. It means saying no to the hundred other good ideas that there are."

Right, and the same thing goes for managing a data team. In the same way, when you're staffing or building a product, think about how well it matches the priorities of the company. LinkedIn could do a million different things, but you want to focus on things that actually align with the strategic goals of the company. There are many things that would align with the vision, but that doesn't mean it's the right thing to do. So you need to prioritize, and you need to do it in the context of all the other things you could possibly do.

It sounds like a great deal of understanding the ins-and-outs of data science is learning how to focus.

Yes. It may not take seven years of focus like a PhD, but you probably need at least a year to do anything of really significant value. If you are coming out of school and you want to work on a data team, you need to find good mentors. You need people who are training you up on the engineering stack, who are sharing the common tools, and helping push projects through management layers. I hear a lot of complaints where lone data scientists feel like they have no support structure. It is really hard to operate without a team on your side, because I think the personality type of scientists is often not the most assertive when dealing with business stakeholders.

Can you talk more about the growing importance of data science within companies?

I think data teams are building really important things. They are actually going about it in a very deliberate way and they're using reason, theory and evidence. People from science backgrounds are well suited for this, because you're building up a theory of what you think will happen if you were to make certain changes to the product. I think that that is really at the core of the skillset that you want in engineering product development and data science to make informed decisions.

I think that data science is going to become this discipline that drives decision-making

and product development. In order for data to have the biggest impact, it needs to be in the early phases of product development rather than just added as an afterthought.

It also involves giving feedback to the product and engineering team about the quality, type and quantity of data that will be collected and affected given certain product decisions. It's incredibly important to have someone sitting in the room and advocating for the data team every time a new product feature is proposed. That may be easier if data science itself rolls up within the engineering or product organization, or has an advocate reporting to the CEO like a Chief Scientist or Chief Data Officer.

If you are coming out of school and you want to work on a data team, you need to find good mentors.

Of the people we have talked to, you can offer a unique perspective on how to effectively manage a data science team because you are so engineering focused as well. There are a lot of managers who are very people focused or they sort of try to massage the politics of the company to get things done, but you seem to want to stick very closely to the nuts and bolts of a company. So what do you find to be effective in creating a data science team?

Jeff Weiner had this framework for prioritizing decisions around vision, strategy, mission and objectives. He used it as a leadership framework and a way to rally people behind a vision. Of the things that I think an effective engineering manager needs to have, one of them is expertise. If you don't understand what the people on your team are doing, you're going to have a hard time making the right calls. Beyond that, you need to be an advocate for what is right for the company and by proxy what's right for your team.

A good leader for a data science team understands some data science, has some vision to see what the right path is, brings the right people in, gets the resources, and then gets out of the way and gets other people out of their way. If your team is being thrashed around and pulled in different directions, it will be hard to stay focused.

There's a great talk by an MIT professor named Fred Kofman who has a book on business strategy called *Conscious Business*. He says when most people are asked what their job is, they reply with their job title. But that's a very limited way to think about the role you play within a team or company. If you use the analogy of a soccer team, the various players all may have different roles. For example, as the goalie your job is to simply stop the ball. As an attacker your goal is to score the most number of points. But if you are completely optimizing for these local metrics, your team still might not win! So I think that what can really make teams successful is everyone really believes in what they're doing, believes in the mission and feels like they're enabled to accomplish that.

How did your perspective change throughout your own life?

Earlier in my career, I thought what was blocking me from more success was not having the engineering skills. Over time, as LinkedIn went from 300 to 5000 employees, what's often difficult for organizations at that scale is communication and coordination issues. What I often would see blocking people was of that nature. It was less pure engineering ability, and more: "How do you get stuff to ship? How do you get resources? How do you get priority?" If I were given total freedom, I would actually just enjoy building stuff and building algorithms, but when you want to maximize impact and success at the company, I think more of what was blocking me at that stage was having to navigate structure within the company.

He says when most people are asked what their job is, they reply with their job title. But that's a very limited way to think about the role you play within a team or company.

My two cents of advice on that would be engineering, engineering, engineering. Because in that environment, or at Facebook or Google, optimizing and getting that right is really going to enable you much more than other alternatives. In a larger company, you're always going to have challenges of organization and your throughput isn't going to be as high by definition. That's why startups exist.

I really like what you said about the fact that even if you wanted to build, build, build, it would be more effective for a team to unblock their processes. That tied in very closely with the analogy you have with soccer players. Some soccer players want the personal glory but the best soccer players are the one who realize it's the team winning that is more important than fulfilling their own goals.

I think it's a balancing act, and I would add one other thing. My opinion has been the best way to show the way is just to do good work. But it's not enough to just do good work; you also have to talk about it. That's something else you can pull from science because a big part of science is communicating. There's value in that, both from a recruiting respective, but also for training the next generation. I think that's the way we build and weave on top of each other's experiences, it's all connected. I would balance talking about your projects with building. I would say work hard, work for a long time, and then talk about what you did and go on to the next step.

Given all of your experience and perspective, what do you think is going to be the future of how data is used in the world?

Four to five years ago, I think investors were a good proxy for the future. They might not

come up with all the ideas, but they hear a lot about what people think and are cued into where things are headed. It was very early for the data space and people were building low-level backend technologies. Over time, interest started to shift to what gets built on top of these backend technologies.

I think what people are really thinking about now is how to replicate the Google and Netflix approach and map it onto the rest of the world. There is a much bigger wave coming of building tools and applications on top of all of this data and infrastructure. There now exist data companies in oil & gas, and health care and other areas, taking on sorts of different verticals.

I'm looking forward to seeing a set of data companies like this. I think all the data companies that are building better platforms and better tools are making everyone's life easier and I want to see more of that, but I also want to see more industries disrupted in a way where it makes society more efficient and people's lives better.

I think the other wave we're hitting is that of social data. All the social data that is being generated is really instrumenting the world and people's behaviors in an entirely new way. Everyone has a Facebook account, a LinkedIn account or a Twitter account, which provides immediate context about the person. We're never lived in a time where there's so much context about you readily available to make your daily experience better. The other key component of this is that we all have mobile computers in our pockets, generating all this ambient data.

We're going to see more smart software at the intersection of those two trends. For example, why does it take four hours to book a flight right now? There are all these workflows, suboptimal systems, and paperwork which could be much easier using mobile and social data. In movies like *Her*, you are starting to see where the rise of Google Now, Siri and things like that could be headed. One thing I think is really interesting is this entire field of intelligent systems. That's a common thread in things I've worked on.

I think that having these techniques and this intelligence in that sea of data acting on your behalf is the next stage. You have context, you have alerting, you have all these disaggregated unbundled verticals like Pandora, but I think next you're going to see this really cool future where you're going to express a desire and intent and something else is going to make it happen. I think that's what I'm most excited about, and why I think for data scientists, the world is your oyster.

MIKE DEWAR

Data Scientist at *The New York Times* R&D Lab

Data Science in Journalism

Mike Dewar is a Data Scientist at the New York Times R&D Lab. Mike holds a PhD from the University of Sheffield, UK, where he studied the modelling of complex systems using data. His current work now focuses on building tools to study behaviour.

Before joining The New York Times, Mike worked at the New York tech company bit.ly, and completed postdoctoral positions at Sheffield, Edinburgh and Columbia Universities. In this interview, you'll read Mike's stories about fruit fly necrophilia, how The New York Times looks into the future and ways that data science is affecting journalism.

Mike is a data ambassador for the non-profit organization DataKind, and has published widely on signal processing, machine learning and data visualization.

Can you trace your career path for our readers? What got you interested in data science? What got you interested in bitly and The New York Times, and what projects have you done that you can share with our readers?

I got my PhD in Modelling Complex Systems from the University of Sheffield in the UK. The department is called Automatic Control and Systems Engineering, which in the US is sometimes called Controls or Cybernetics — it's the study of feedback, modelling, and control.

My PhD looked at modelling spatial-temporal systems. The idea is that you would collect data from the physical space and then build dynamic models of how the system evolved through time using the data you collected.

Then I did a few postdoctoral positions. I did a post-doc at the University of Sheffield for a year. We worked with Unilever and I looked at modelling how people were brushing their teeth. By attaching sensors to a toothbrush with accelerometers and positional sensing, they collected all this data about how people brushed their teeth — it was a very strange gig.

I did that for a year, spent some time writing up the papers for the PhD, and then I decamped to Edinburgh University, where I worked in the School of Informatics, studying

the behaviour of fruit flies. The biologists would alter the brain of the fruit fly and observe their changes in behaviour. In courtship behavior, specifically, the changes were easy to see. If you place a male fruit fly in a small space with a female fruit fly, even the dead body of a female fruit fly, it will mate with her. Well, it will definitely try at least, which is a bit grim.

So, there were loads of fun modelling of sequences and some nice machine learning. I even got to learn how to prepare mutant fruit flies. Most of this work was done at Edinburgh but also included a little bit of work at Harvard, at the Longwood Campus. Then I got the gig at Columbia, which was in the Applied Physics and Applied Math Department. That was with Professor Chris Wiggins, who you might have come across in your studies of data science.

Essentially, leaving academia is a moment where you have to decide if you want to be a professor or not, and I think I'd already decided that was not quite what I wanted to do.

He and Hilary Mason wrote a blog post which outlined various steps of data science, namely: "Obtain, scrub, explore, model, interpret." The steps outlined this idea of a data science flow being practical and producing tangible outputs. Chris was thinking a lot about that with Hilary while I was studying T-cells.

There are lots of different types of T-cells - the population of these different T-cells in your body changes before, during and after an infection (this is how immunization works). So after an infection, you have "memory" T-cells in your body. The group at Columbia was very interested in how T-cells change to this "memory" state.

They collected lots of genetic data and looked for different genes that were responsible for changing the state of these populations of cells. You'd be working with 8 microarrays, but each microarray would have 25,000 genes on it. You had a very strange machine learning problem, whereby you had very little data to go on but it felt like you had a lot because of all the features.

It was through Chris Wiggins that I met Hilary Mason, who was my boss at Bitly. I had also become engaged to a girl who lives in New York, so when it came time to start thinking about what was next after my post doctorate at Columbia, it was important to me to stay in New York. But life as a postdoctoral student in New York sucks because it's quite expensive here. At the same time, the idea of "big data" was just coming to the forefront. There were numerous social media companies that were just starting to think about what they might do with all their data. I was interested in behaviour and making tools for studying behaviour, so Hilary showed up at just the right moment when I wanted to

pay the rent, stay in New York, study behaviour, and use lots of data while doing it.

So I jumped ship and went to Bitly as a data scientist. I think I'm probably amongst the first people who had that title. I made tools at Bitly for studying very large numbers of people's behaviour and trying to build interesting, potentially profitable streams.

Bitly ran its course. I was there for about a year and a half. We did lots of interesting things, but it became time to move on. About that time, a position at The New York Times R&D Lab showed up, which was somewhere I'd wanted to work for years, so I moved over to the lab where I've been now for about two years doing all sorts of interesting things.

Essentially, leaving academia is a moment where you have to decide if you want to be a professor or not, and I think I'd already decided that was not quite what I wanted to do. I like coding and making things, but I don't enjoy talking all day, so that was the decision I made.

We've been talking to a lot of people who decided to jump ship from academia. It seems like a lot of them have been citing reasons such as the lack of dynamism. They felt that data science was much more interesting and fast paced. Did you feel that as well?

No, not really. Academia was very fast paced and very intense, with cutting edge research. The stuff I got to work on was amazing. Watching the very modern imaging of T-cells changing and learning about viruses was overwhelmingly fascinating. When the practical wasn't going quickly, the theoretical was going quickly, and there were always ten different things to do.

I had a very interesting time in academia. Postdoctoral positions are great fun. Lecturing, however, didn't look like so much fun. I wanted to hold onto the fun bits of academia and get paid, which is no small thing when you are starting a family. When staying in New York, which is a bizarrely expensive place, a certain set of constraints comes your way. In short, academia was amazing.

It seems like from your academic background you learned a lot from looking at a complex system, a mass of data, and extracting stories and hypotheses from that. You talk about how the unifying theme in data science is actually just identifying massive behavioural phenomena. What is your advice for identifying the questions, telling the stories, and identifying the hypotheses in the data set; especially since you say data science is all about abductive reasoning, finding stories, and learning from data? What is your advice for learning which story to tell with data and what to look at?

The key piece of advice is always to draw lots of pictures and draw them very quickly. Draw pictures of how things work, even just flow diagrams or engineering block diagrams. Make very rough, quick visualizations of what's in the data, starting with time series and histograms. Thinking hard about graphical modelling and really trying to get to grips with the system and data set that's in front of you helps you think about how the probabilities fit together.

Make very rough, quick visualizations of what's in the data, starting with time series and histograms.

The danger that I see people getting into is that the drawing of the picture becomes the last thing you do, like when you're reading an academic paper. The results and pictures are always at the end of an academic paper, which is a terrible shame. I think the paper should start with pictures of time series and distributions, and go from there into the theory. That's often how we work.

That would be my very general advice: to fail early and to fail often. It's okay to draw lots and lots of pictures that might all be rubbish, but if you draw pictures quickly and really start to understand what's actually going on, you begin to get much deeper ideas of what the right questions are, than if you just start with a classifier.

Can you elaborate on drawing pictures a bit more?

I learned a lot at Edinburgh about graphical modelling, which is a very simple technique for exploring conditional probabilities and trying to explore how random variables in a system affect one another. The beautiful thing about graphical models is that if you start drawing them, you are, at the very same time, beginning to explain your assumptions about the system. Also, you're starting to do quite a mathematical task of imposing some structure that you can then test. I really enjoy quickly trying to show whoever I'm working with a graphical model of how I think things work. The conversation gets going very quickly and it leads to testable hypotheses, which is great.

The other interpretation of drawing things quickly is to get immediately into the data set. As soon as someone hands you a data set or gives you access to a stream, the very first thing to do is to find an interesting variable in the data set and plot it. If it's over time, plot a time series. If you've got lots of samples of that variable then plot a distribution. If it's both then plot both. You can do that using Python, or R, or Tableau, or Excel. Do that first and don't waste time. It takes five minutes to make some plots.

The reason is that it gets you thinking about your assumptions in the same way that graphical models do. The distributions and the time series get you thinking about the

data. Both of those together are the beginning of a modelling process that will see you in good stead. It's quite an iterative process. If all you've got is a Bash terminal, then I would sort my data and then pipe that to "uniq –c" to get a really cheap histogram.

You say that visualization and communicating data is very important because it helps other people generate hypotheses and trust the data. What advice do you have for the best way to approach making visualizations to an internal company audience?

One thing we've been doing lately is trying hard to show all the data. I would normally start by thinking about how a system is working and what I'm trying to get out of a data set. Then I would draw some aggregate visualisations, for example, a histogram if I'm interested in how things are distributed, or a line plot if I'm interested in a time series.

As soon as someone hands you a data set or gives you access to a stream, the very first thing to do is to find an interesting variable in the data set and plot it.

One thing we've tried to do more recently is to draw every single data point in a visualisation, rather than aggregating, just like in a scatter plot. This is something made much easier as I now get to work regularly with Nik Hanselmann, who is a creative technologist in the lab and is extremely adept at this sort of thing. If you can make a scatter plot of a large data set interpretable, then that act of showing all of the data points allows people to see a zoomed out view of the whole thing and allows them to pick on individual data points. They see the outliers and wonder why there's an outlier there.

Clusters are another good example. If you've done your scatter plot well and people can start to pick out different features of the scatter plot by looking directly at the bit of data that you want to show them, then they start to ask questions and start to wonder. That helps you as the analyst or the data scientist. It helps you in trying to understand what your audience is actually interested in and how you might help them make decisions. It's an incredibly difficult thing to do without some sort of interaction like that. Trying to show all the data points is quite challenging sometimes, but that's been oddly effective over the last year or so.

Other than that, axis labels. I feel old saying it but lots of people don't put axis labels on things. You read lots of blog posts about lying with statistics and all that sort of stuff and all the tricks people play, which is fine, but it's very difficult to get away with those tricks if you label your axes properly. You shouldn't trust any graph that doesn't have their axis labelled properly.

How do you think this whole explosion of data, as well as computational power and analysis on top of that, is going to affect the nature of journalism?

The reason that I ask this question is that where I went to school at UC Berkeley, there were actually quite a few workshops held for students who wanted to go into journalism, but these workshops weren't at all about journalism. They were all about D3, Javascript, Python, and R. To somebody who doesn't have as much background knowledge, how would you describe what's going on regarding big data and journalism?

There are a few parts to your question. There have been computer-assisted reporting (CAR) journalists for a long time now. Our computer-assisted reporting desk has been around for many years so the idea that data has been affecting journalism is not a new one.

This is how I came to grips with the term "big data." A friend of mine pointed out that we should think about big data like we think about punk — a cultural moment that was meaningfully hyped for a period, which then led to a lasting change in society.

I like this idea of "big data" as a cultural moment because there's been a definite change in the amount of data we can collect and the expectation of collecting data in the first place. The standard costs of storage, processing, and transmission have all gone down. There has been, over the last few years, a dramatic cultural shift in and around data storage. That hasn't gone by journalists — it's quite the opposite. What they're faced with are huge data sets that they think might contain stories. Or it'll be the other way round where they believe that there is a story and will use the FOIA (Freedom of Information Act) to get data sets.

A journalist will often assume that there is data associated with a story and will demand to see it.

When they're telling a story, or they believe that there's data associated with the story, they will search government organizations or FOIA organizations that have worked with the government and are subject to the Freedom of Information Act. I think the rise of the FOIA is an interesting response to big data in the sense that a journalist will often assume that there is data associated with a story and will demand to see it.

Rather than the WikiLeaks style, where there is a huge data dump for one reason or another, journalists will believe that there is data associated with their story and will use FOIA data in order to support the story. This is a lot more work, and that work is a lot more laborious. The impact of big data is a culture where we expect there to be data.

The other side of that, which I think is probably a bit sexier, is the WikiLeaks side of things, where there is a huge pile of data that is being made available - like the Medicare data. There's a huge data set that's been released around how Medicare dollars are spent with personal information about doctors that receive Medicare funding. There have been a lot of stories out of that data set that are very interesting. That's the other mode that journalism works in. That's when people want to use R, or Python, to clean and analyse the data and d3, ggplot, or matplotlib to build visualizations of that data set. D3 is especially interesting because it's used to make web and print graphics, which is why you see it a lot.

What can you share with us about what you do at the R&D Lab, especially since most of the people we've been talking to work at technology companies, instead of a journalism company with a very strong technology component?

The R&D Lab was set up in 2006 to fulfill a number of roles. Specifically, it tries to think three to five years into the future, tracking social, cultural and technological trends relevant to The NYT. That gives us quite a range of possible projects.

We're thinking a lot lately about how to extract information from article data.

The other function of the R&D Lab is essentially to listen. That takes two forms. One is a futurist approach where we try and watch what's going on in the blogosphere and watch what's going on in new technologies. We try and keep an ear out for anything that looks like it might have something to do with the future.

We also act as a gateway. If someone is developing a new, interesting business software that they think The New York Times might be interested in, but there's not an immediately clear use case, often we'll speak to them. We'll ask them some questions and try and understand what they think the future looks like. We can think about how that fits in with how The New York Times thinks about the future.

In terms of projects, it's quite varied. We're thinking a lot lately about how to extract information from article data. Given an article, can you extract all of the statistics, the quotes, facts and events? This is a tired old problem, so we're trying to think about other ways we might accomplish that. Can we capture information like that during the writing process or the editing process or the production process, rather than approaching the articles with an extractive, natural language processing view? It would be much more interesting to see what metadata we could generate in the first place.

That's an example of journalistic stuff. Then, we think about how the news might be

presented in the future. One example is a good idea that the lab had regarding the future of tablets. The New York Times R&D Lab had thought about what a tablet reader application would be like well before the iPad came out. When the iPad did come out, The New York Times had a head start in understanding how people might interact with their tablet and what would be interesting to show on it.

What advice do you have for other PhD students and people in academia transitioning to data science, especially since you've been through this already? What advice would you give someone interested in transitioning to data science?

Code in public.

Code in public, that's number one. If you're going to be a data scientist, you're probably going to have to be able to program in one way or another. There are lots of different options, but you're probably going to have to be quantitative and be able to write non-trivial programs on the computer. As you code, as you practice, as you go to hackathons, as you code for your post doctorate or for your PhD or for your graduate degree, make sure you do it in public. Put it on Github. To a certain extent I'm on the other side of it now where I put every thing I think of on Github, so it's a bit of a mess.

Especially with PhDs, one of the problems we see is that although they come from impressive universities, they have impressive resumes, and they've written these nice papers, but we still have no idea if they can actually write code. That makes them more difficult to hire.

Coding in public also encourages you to engage with communities that you work with. There are programming communities that share your languages; academic communities that might want to use your code to test out your claims; and companies that want to evaluate you and reduce the risk in hiring you.

The other thing is networking. It's more or less the same thing, but it's important. In major cities it's very easy for you to get out of your office or house and visit meetups and user groups to give a talk. Giving talks about your academic work to lay people is an incredibly interesting and enlightening experience, one that you should go through. It also exposes you to the business communities and the various kinds of people that you might want to get jobs from in the future. It also shows you what other people are up to; it knocks your academic naivety very quickly, which is great.

Other than coding in public and networking, try to apply all your work to something. I wrote three papers for my PhD, and they were all about the EM algorithm. There's a load of spatial-temporal models that I put a lot of work in to. Over 3-4 years, I wrote

some papers, and nobody cared, nobody at all. However, when we applied this theory to modelling troop movements in Afghanistan, lots of people cared. We won awards. We wrote a book. We were in the news. The idea of taking the advanced things that you learn at school and applying them to something important and meaningful exposes you to a world that's difficult to see from the incremental science of being a good student.

RILEY NEWMAN Head of Data at Airbnb

Data is the Voice of Your Customer

Riley Newman paid his way through college at the University of Washington by being part of the US Coast Guard. After graduating with degrees in economics and international studies, Riley pursued graduate studies in the UK at the University of Cambridge, before he was called back to the US by the Coast Guard.

After working for a few years in economics consulting, Riley met the founders of Airbnb, and was drawn to their vision and focus on culture. He ended up joining Airbnb as one of the early employees.

Now, Riley is the Head of Data Science for Airbnb where he data science teams using data data to listen to customers' voices and desires.

Can you explain a bit about your background and how you came to Airbnb?

I went to college in Seattle where I majored in international politics and economics. Halfway through my time there, I realized the value of statistics for understanding social trends better. This was something I deepened in grad school, and wanted to pursue in a PhD, but I had joined the Coast Guard in undergrad to pay for school and they called me back from the UK after my master's. So I came home to the Bay Area with the plan to get some experience working with data while completing my Coast Guard obligation and then planned to head back to the UK for the PhD.

I spent three intensive years working with a group of economists that were modeling the 2008 recession. One of them had a degree in computer science and taught me the value of automating analytical processes, which I found intriguing. When the time came to leave for the PhD, I was torn — I only wanted to do it for the technical training; my heart wasn't in academia, and I was a bit tired of consulting. Serendipitously, I met the founders of Airbnb through a mutual friend, right as I was struggling with this.

There are a couple of things about Airbnb that resonated with me. First and foremost, the concept behind the company. In undergrad, I read a lot about globalization and the growing interconnectedness of the world; also about the fundamental sustainability issues associated with this trend. Airbnb struck me as a solution — it would facilitate more travel and bring international communities together without requiring the construction of additional structures.

I was also attracted to the founders' focus on culture. This is something I hadn't experienced in previous roles. They placed so much value on the sense of camaraderie on the team — more so than high school and college lacrosse teams, Coast Guard units, or the consulting firm — and the impact that brings to our work. Looking back, I think this is the "secret sauce" of Airbnb's success.

Finally, I was excited about helping to build something. As a consultant, I had exposure to a wide variety of problems but, at best, we could convince the client that our work was actionable. At Airbnb, I would be able to follow the analysis all the way through to impact. And startups are fast-paced environments where you can see the impact of your work on a daily basis. That was really exciting to me.

They placed so much value on the sense of camaraderie on the team. Looking back, I think this is the "secret sauce" of Airbnb's success.

How does this tie into the industry buzz around "Big Data"?

"Big Data" is such a common term these days. I heard a joke recently, "What do big data and teenage sex have in common? — Everybody is talking about it. Nobody knows what it is. All their friends say that they do it, so they say that they do it too."

Like all buzzwords, Big Data is getting tiring. But I met with a more seasoned data scientist recently who described the field in the '80s and '90s — there was much less data so they needed to use advanced statistical methods to identify simple trends. These days, with the volumes of activity web companies are able to generate, and the depth of storage facilitated by technologies like Hadoop, we're able to gather and make use of much more data. So it's more of a question about how to sift through it all. I think this has made computer science degrees that much more valuable.

I have a data scientist friend whose resume begins with three things he firmly believes: more data beats better models; better data beats more data; and the 80/20 rule. I couldn't agree more.

I think that is a really good introduction as to what you think data science is. I want to go back to something that you mentioned earlier; your Master's degree was in Economics, is that right?

Yes, I was in the applied economics department at Cambridge. My research was in the field of economic geography/spatial econometrics.

Many people we've interviewed have their PhD or Masters in physics, statistics, math, or computer science. You're one of the few data scientists we've talked to with an economics background. Do most people on your team tend to come from backgrounds in the hard sciences or are the social sciences represented as well?

More data beats better models; better data beats more data; and the 80/20 rule.

Everyone on the team has some degree of quantitative training but I like having a wide variety of backgrounds because this brings different skill sets and approaches to solving problems. For example, computer scientists are great at scripting automated solutions and productionizing models; statisticians ensure our models are rigorous; physicists are very detail-oriented; and economists can build frameworks for understanding problems. Airbnb is particularly interesting to economists because of our two-sided marketplace, which lends itself to modeling supply and demand, and looking for ways to make our markets more efficient.

But the key thing is that everyone on the team is able to drive impact in the company through the cultivation of insights drawn from data. I'm less interested in what people studied in undergrad than in their ability to do this successfully. However, this requires a solid grasp of statistics, experience in coding, and great communication and problem-solving. Our interview process exposes these skills very well, so we're able to consider people from less traditional backgrounds.

I agree with you that there isn't necessarily one particular field that data scientists come from. However, it does seem like most people come from certain fields that tend to teach some of the most relevant skills. Building on that, what would you say are some of the most valuable or relevant skills that someone in academia should build right now?

Many people coming into data science from academia have honed their ability to think mathematically or statistically and, to some extent, work with data. The big division that I see is the ability to lend those skills towards problems that will result in an actionable solution. In other words, the types of questions they ask are as important, or more, than the methodology behind solving them. In their research, they focus on why something is the way it is or how it works; in industry we're more interested in what we should do. If the how or why lends itself to answering this, great. But if nothing changes as a result of your work, then it wasn't that valuable.

When we ask other data scientists that question, we hear about technical skills like Python and programming. We don't hear as much about extracting actionable insights.

I'm not saying that those aren't relevant; I'm presupposing that anyone hoping to generate actionable insights from data has the ability to work with the tools of the trade. At Airbnb, we mostly use Hive, R, Python, and Excel.

When we ask other data scientists that question, we hear about technical skills like Python and programming. We don't hear as much about extracting actionable insights.

When we interview people, our process is very transparent (see Quora post on this, here). We give candidates a day to solve a problem similar to something we've faced, using real (but anonymized) data. They spend the day seated with the team and are treated like anyone else, meaning they can collaborate with anyone. At the end of the day, we have them walk us through what they found and tell us what we should be doing differently as a result. This is too tight of a timeframe for someone to learn a tool while trying to use it to solve the problem. Their time needs to be completely focused on getting to that actionable insight.

Continuing along that vein, you've been talking a lot about data scientists coming from a Master's or a PhD. I'm also wondering about your opinion of people coming in with a Bachelor's in a quantitative field or similar?

People can absolutely break in with just a Bachelor's. We shifted to the interview model I described earlier because we realized our image of a data scientist was yielding false negatives. If you have the right mindset, a decent understanding of statistics, and can use SQL and R, you'll be able to get a job.

This is particularly true in younger startups. When I think back to the early days of Airbnb, we were able to squeeze a lot of growth out of a simple ratio. If I spent a month building a perfect model, I would have wasted 29 days. As a company matures, so does (hopefully) its understanding of its ecosystem. So there's a need for more sophisticated approaches.

You started at Airbnb when it was in a really early stage. Now you're at the point where it's growing very fast — it's become a large company. What are some of the ways you've seen that transitioning into the work that you actually do?

I see this transition shaping our work in two ways. First, the team is big enough now that we're able to go much deeper into problems. In the past, we were jumping from one fire to the next, so we weren't able to invest large amounts of time into a single problem. And that's natural for a startup. But as the team has grown, we've been able to focus on some of the key topics for the business and understand them more deeply. We also now

have people on the team building data products, which is exciting.

Second is the democratization of information. We're not the only team that has grown over the last few years and everyone is hungry for data to guide their work. So we have to find ways to remove ourselves from the process of answering basic questions. The last thing you want to be is the gatekeeper of information because you'll spend all of your time responding to ad hoc requests. So we've invested a lot in the structure of our data warehouse and the tools used for accessing it so that it's intuitive to people with less experience working with data.

What do you think are some of the most fundamental ways in which data science can add value to the company?

I think data can add value everywhere. It's the voice of your customer — data is effectively a record of an action someone in your community performed, which represents a decision they made about what to do (or not do) with your product. Data scientists can translate those decisions to stories that others can understand.

When I think back to the early days of Airbnb, we were able to squeeze a lot of growth out of a simple ratio. If I spent a month building a perfect model, I would have wasted 29 days.

We spend a lot of time with our product team, which is the most traditional place for a data scientist. There's a wide range of work happening here. For example, our trust and safety team builds machine learning models to predict risk and fraud before it takes place. They also have to think about ways to measure intangible things, like the strength of trust between people in our community so we can identify ways to improve this.

We have other people working on matching guests and hosts, improving the model behind our search algorithm and uncovering new features to improve the match. A while back we published a blog post about this here.

With our mobile team, we try to uncover opportunities for improving the app. One guy on the team looked at the probability of performing an action on the app relative to how far away that feature is from the homepage. This obviously showed that the more buried something is, the less likely it is to happen — but it's a framework the mobile team can now use to think through the structure of the app.

But we don't just work with product. We think about user lifetime value and growth opportunities with our marketing team, operational efficiency with our customer support team, and we've even been chatting with our HR team about how they can leverage data

to better understand recruiting and career growth.

I try not to segment our work by stakeholder; rather I look at the key drivers of the business and try to figure out what problems need to be solved in order for Airbnb to be better, and then figure out who is in the best position to use that information.

So we've invested a lot in the structure of our data warehouse and the tools used for accessing it so that it's intuitive to people with less experience working with data.

Airbnb is a transactional business so there's a funnel we can break apart and analyze. And getting back to the concept of data being the voice of the customer, we always start by looking to our community for advice on what to do next. At the top of the funnel we try to understand how people are hearing about Airbnb. We can use online and offline marketing to drive this, emphasizing growth where we think there's a strategic opportunity or where we're seeing positive ROI (return on investment). For this, we begin by looking to our community for ideas; for example, where many people are searching for places to stay but we don't have enough supply to accommodate them. If this isn't an anomaly (e.g. one-off event), it represents an opportunity for growth.

Next is the experience people have when they come to our site. There's a lot of A/B testing here, looking for ways to make it more intuitive and satisfying to a person of any demographic, anywhere in the world.

After that is the offline experience, which is tricky because the data behind this isn't as rich as site usage. But we can get a lot from the reviews people leave each other - a combination of quantifiable ratings and NLP we can perform on the text of the review.

Finally, we look at what we can do to get people to come back and try it again. Mostly, this means improving each of the steps above, but we think about experiences people have with customer support or community groups as a way of staying connected.

The final thing I would love to get your perspective on is just looking towards the future. Where do you think the future of data science is and where do you think we are relative to what data science could be in the future?

I think we'll see a lot of growth on the tools front. It's amazing how quickly Hadoop and Hive have matured just over the last few years and there are new and exciting technologies emerging almost daily. So I'm hopeful that we'll eventually have lightning-fast tools that can work with data of any size.

I also think data logging will develop a lot, because people are aware that you only focus

on what you can measure, and you only measure what you can log. So questions like we have about the offline experience will hopefully get easier to answer as data becomes more ubiquitous.

Good data science is more about the questions you pose of the data rather than data munging and analysis.

Every now and then I see an article about the field of data science disappearing to automation. In effect, the tools get so good that you don't even need to analyze data; the insights are just there waiting for you. While this may be partially true with the growth of machine learning, I don't think it will ever fully be the case. Good data science is more about the questions you pose of the data rather than data munging and analysis.

But with that in mind, I can imagine the field of data science opening up to people that are less technical. As tools get more sophisticated and easy to use, we'll see more people getting excited to work with data. We've already observed this at Airbnb, where we train everyone in the company to use SQL. As I mentioned earlier, you don't want your data science team to be the gatekeepers of all information. We want everyone to be able to interact. I love watching people with no background in statistics or CS wrapping their minds around the basics of working with data. They get so excited, then they get curious. And that frees us up to focus on interesting problems that will impact the business.

This sounds like the democratization of data science.

Exactly. It's happening today at Airbnb, and I bet we see a lot more of it in the future.

CLARE CORTHELL Data Scientist at Mattermark

Creating Your Own Data Science Curriculum

After graduating from Stanford, Clare Corthell embarked on a self-crafted journey to acquire the knowledge and skills to understand and analyze macro-behavioral trends. One thing led to another, and her collection of resources turned into the Open Source Data Science Masters - a curriculum of online courses, books and other resources that one could use to learn the mathematical and programming foundation crucial to a data scientist.

Clare took a risky move by crafting her own degree program, outside of traditional educational institutions. She faced skepticism of a self-taught individual in a job that is typically inhabited by PhDs, but also found a community of supportive colleagues.

Overcoming these challenges, Clare completed her Open Source Data Science Masters and found herself as a data scientist at Mattermark, a venture-backed data startup working with large datasets to help professional investors quantify and discover signals of potentially high-growth companies.

What was your background, before you began the Open Source Data Science Masters and before your role at Mattermark?

I'm a product person and an entrepreneur. I fell in love with startups long before I attended Stanford, where I designed a degree in a then-obscure program called *Science, Technology & Society*. You get to marry two engineering tracks, so I ended up designing a degree in product design and digital development, which then got me started working on product with early stage companies.

Before the OSDSM (Open Source Data Science Masters), I was designing and prototyping products for an early stage education technology company in Germany. Designing from user anecdotes alone became difficult when you only pull from anecdotes, so I started digging deeper into analytics and customer profiling. I started thinking about observing meta-trends among users instead of studying their behavior with a clipboard from behind a one-way window. What if I just ran several tests on two different prototypes? Then we would have data to tell us which one to develop! But as with many European startups, the company didn't get funded, so I had a few weeks to think about how this new perspective fit in. On a long layover in Barcelona, I ordered an espresso and wrote

down the technical skills I would need to dissect meta-trends and understand user data. That list laid out 6 months of full-time work, after which I'd really be able to do some damage. This became the Open Source Data Science Masters.

I started thinking about observing meta-trends among users instead of studying their behavior with a clipboard from behind a one-way window. What if I just ran several tests on two different prototypes? Then we would have data to tell us which one to develop!

As with any story, it is now retrospectively clear that I would secretly fall in love with an applied statistics class I cheekily called "Exceltastic." We worked with Bayes' Theorem and Markov Chains in the business context, figuring out things like how many cars can pass through two toll booths per hour. Everyone else sulked and moaned through munging spreadsheets while I harbored a dirty secret: I loved Excel models! Even so, I didn't know when my toll booth throughput calculations would be demanded of me, nor what class logically comes next. It took getting into industry to shed light on the value of keeping metrics. Things like my Exceltastic class don't seem to fit into an overarching puzzle, but we believe they shape our path. That's the power of confirmation bias. One of my favorite designers has this phrase that he prints in various media: "Everything I do always comes back to me." I've always found that fitting.

What is the Open Source Data Science Masters? What does its curriculum look like?

It's a collection of open-source resources that help a programmer acquire the skills necessary to be a competent entry-level data scientist. The first version included introductory linear algebra, statistics, databases, algorithms, graph analysis, data mining, natural language processing, and machine learning. I wrote the curriculum for myself, then I realized that people all over the internet were asking for it, so I published it on GitHub.

In August, I opened the curriculum for pull requests on GitHub. Without feedback it's difficult to know whether you've covered the right things. Further, it was an effort to get feedback on the idea of an institution-free degree, a kind of home-school for advanced degrees. The internet was astonishingly supportive and excited — and that excitement is addictive. It makes you want to be more transparent, and to become part of other peoples' wonder in learning new things.

How did you get started with the Open Source Data Science Masters?

I knew that a traditional Masters program would take at least the next three years of

my life, but even more importantly it wouldn't focus on what is core to the profession I wanted to enter. I knew what I wanted and I was willing to take the risk of a non-institution education.

It took getting into industry to shed light on the value of keeping metrics. Things like my Exceltastic class don't seem to fit into an overarching puzzle, but we believe they shape our path.

I set out for the curriculum to take 6 months to complete (March - August 2013), with a small project at the end and various programming mini-projects focusing on scraping, modeling, and analysis. It was amazing how difficult it was to manage myself. School gives you this structure that you don't have to question or design, which you don't really see until you have to manage your own curriculum and deadlines. There's a lot of product management that goes into an educational track like the OSDSM. I'm grateful to all the people who supported me and helped me throughout, even if they didn't quite understand the strange and uncharted waters I was braving to get there.

How did you find the resources?

I reverse-engineered most of it from job descriptions that interested me. This meant companies I believed would grow quickly and provide the most opportunity: mid-stage startups, 100-200 people, existing data science teams and reverence for the methodology. I didn't want to be the lone wolf and knew I needed mentorship.

People tend to frown on centering the goals of the classroom on applicability in the real world, but a classic liberal educational approach in a technical career pivot won't serve you. This is a technical vocational degree, so the goal was very concrete. I should be employable and employed on a data science (or Analytics Engineering) team after completing the curriculum.

There was another realization that coalesced very quickly: the act of designing from insights of single users does not scale. I was also hankering for something more technically and algorithmically challenging. I'd bought this book before I moved to Germany, *Programming Collective Intelligence*. I just bought it, I really had no reason to. When I first opened it up, I understood next to nothing. But I carried it with me in Germany, and every time I opened it, something new jumped out and I understood more about scaling user insight. The book became my cornerstone, how I measured my progress. It's a bible for Data Scientists.

I also used the following resources/websites:

- **Quora**: This is a great resource for the Valley — it's truly navel-gazing, but if that's what you're doing, it's useful. People like DJ have answered questions about what a Data Scientist does on a daily basis. You can start to discern the technical capacities that are required of you, mathematical foundations that are necessary, and so forth.
- **Blogs:** Zipfian Academy, a data science bootcamp, had a blog. They had a great post on the resources they saw as core to becoming a data scientist: A Practical Intro to Data Science
- **Coursera:** I'm Coursera's biggest fanboy. They're part of this quietly-brewing educational revolution, which will soon be less quiet. My story is a tremor before the earthquake, I'm just waiting for the ground to start shaking.

How much math (probability, statistics, ML) did you try to learn? How much math do you think a data scientist needs to know?

You don't have to know everything. That's why I've tried to keep the curriculum so tightly focused on its goal. Programmers are great at "just-in-time" learning because it's impossible to know everything. That's a great trait. If you have a core set of competencies and understand how to "debug" problems and learn what you need to solve them, you can do damage. And naturally, you improve over time by recognizing new problems as chunks of old problems you've already seen and solved.

The internet was astonishingly supportive and excited — and that excitement is addictive. It makes you want to be more transparent, and to become part of other peoples' wonder in learning new things.

So much of this curriculum is abstract, and that's where people get scared. People are scared of math because it's not applied in our education system. But those scary elements of math and abstraction diminish with concrete examples and conversations with others. I had a few phone-a-friend lifelines, and I ate up Khan Academy and Coursera videos. There's something magical about how much more communicative spoken English can be, especially when you can rewind and digest a concept for the second, third, or even fourth time. You can always talk through a problem with someone else, even if they're not an expert. Talking through things is synonymous with debugging. One of my mentors calls this "the rubber ducky method," because if you talk a problem through to a plastic duck, sometimes you start to find the holes in your assumptions. Then you can plug them up.

If you think about people as having different levels of competency in these different realms, it doesn't take long to understand that working as a team allows you to stack

your respective skills on top of one another. Having specialties among the team is really essential to getting things done in a small organization. I was lucky enough to join a company where I get mentorship in verticals where I'm middling or even an amateur. It's amazing to learn with other people. Finding a job where you have mentors and training is essential to continue to grow and improve. And if you're not improving and growing, you're dead in the water. So that's a long-winded way of saying: Working with other people is essential to working with more complex concepts and systems. Rome wasn't built by some guy, and probably not at a weekend hackathon.

What would you do differently if you could redo the Masters?

You don't have to know everything. That's why I've tried to keep the curriculum so tightly focused on its goal. Programmers are great at "just-in-time" learning because it's impossible to know everything.

As Patient Zero of a new type of internet-based institution-free education, I didn't know what to expect. It was impossible to know how I would be judged and whether I would benefit from my experiment. This type of ambiguity usually makes people extremely uncomfortable. It's like leaving a six-year-old in the library by herself instead of putting her in class with a teacher. What is she going to do? Pull a bunch of books onto the floor and see how high she can stack them? Watch birds at the window and think about how wings work? Or is she going to find something interesting and gather books that will help her form her own ideas about the world?

I knew that it would be a risk, but I took a leap of faith and left myself alone in the library. In the end, the greatest reward didn't come from the curriculum, it came from what taking a risk demonstrated about me. It led me to a tribe that respected the risk I had taken, and valued the grit that it required to follow through. Many people were displeased that I let myself into the library without an adult. But I'm not interested in taking the recommended path and clinging to a librarian. I have no interest in small ambition.

What's the difference between data science job descriptions & day-to-day role at Mattermark?

Our CEO Danielle was once asked how many data scientists we have at Mattermark. We're all data scientists, she thought — we all use, manipulate, and analyze data on a daily basis to make our customers happier and more profitable. We even all write SQL! That's not something you see every day at a company, but it's essential when you're building and selling a data product. I build products as an engineer, anything from fitting

clustering algorithms, building automated analyses, designing UIs, acquiring new data — it's a startup. It's all hands on deck.

It's not clear that data science is a job title to stay yet. For example, do we know if growth hacking is a subset of data science? We don't. There will always be a top-level salary for a person who can turn chaos into insights. That won't change. Data Scientist is a title we'll continue to use while we figure it out.

What could someone in school, or otherwise without too much background in industry learn from your experience?

The ability to evolve my own career with a self-designed curriculum begins to outline the immense cracks in the foundation of higher education*. The deconstruction of this system was very long in coming, but it's happening now. The lesson is the following: if you take initiative and acquire skills that increment your value, the market is able and willing to reward you.

The ability to evolve my own career with a self-designed curriculum begins to outline the immense cracks in the foundation of higher education

Though people continue to believe and espouse old patterns of education and success, these patterns do not represent requirements or insurance. The lack of any stamp of approval is a false barrier. There are no rules.

It's important to understand the behavior of the market and institutions with regard to your career. When breaking out of the patterns of success, know that people will judge you differently than others who have followed the rules.

There are two very discrete things that I learned: The market is requiring people to perform tryouts for jobs instead of interviews, and most companies don't hire for your potential future value.

Tryouts as Interviews: The economy has set a very high bar for people coming into a new profession. Job descriptions always describe a requirement for previous experience, which is paradoxical because you need experience to get it. Don't let that scare you, not for a minute. Pull on your bootstraps and get in the door by giving yourself that experience — design and execute on a project that demonstrates your ability to self-lead. Demonstrate that you can take an undefined problem and design a solution. It will give you the confidence, the skills, and the background to merit everything from the first interview to the salary you negotiate.

Even more concretely, work with a non-profit organization (or another organization that doesn't have the economic power to hire programmers or data scientists) to create a project that is meaningful for the organization and also shows off your skills. It's a great way to do demonstrative and meaningful work while also aiding an organization that could use your help, and likely has problems people are paying attention to solving. Win-win.

Current Value vs Potential: Look for companies that will hire you for your potential. It's important to be upfront about your grit, self-sufficiency, and ability to hit the ground running. Luckily, with disciplines like data science, the market is on your side. Sometimes companies can spring for a Junior Data Scientist and invest in your growth, which is really what you wanted from the beginning.

Talk with people who can recognize hustle and grit, and not necessarily those who are looking to match a pattern drawn from your previous experience.

Everyone will tell you this, but I work on product so I'll underline it even more strongly: Learn to write production-level code. The more technical you are, the more valuable you are. Being able to write production code makes you imminently hirable and trainable.

*[*NB: Don't think for a minute that I don't believe in the tenets of a true liberal education - quite the contrary. I continue to read philosophy and history, in part because we cannot draw fully upon the knowledge of man without doing so. These are essential elements to being a purposed, ethical, and effective person - but they don't directly accelerate a career. The true liberal education has nothing to do with market forces, and never should. Higher Education as it exists today and Liberal Education should be held as wholly uniquely-motivated institutions.]*

How was your self-taught path to becoming a data scientist received by company recruiters? What advice would you share with entrepreneurial individuals who are interested in the field?

Talk with people who can recognize hustle and grit, and not necessarily those who are looking to match a pattern drawn from your previous experience. Often, these kinds of people run startups.

Recruiters gave me a very real response: They didn't see my course of self-study as legitimate. It's hard to give yourself a stamp of approval and be taken seriously. I wouldn't recommend that just anyone do what I did — it will take a while for autodidactism to

become more accepted, and maybe it will never be a primary pattern. But maybe people like me can help expose this as a viable way to advance professionally. I know that great companies like Coursera will continue to innovate on these new forms of education, keep quality high, and democratize access.

tl;dr

If you want to get to the next level, wherever your next level may be, it's possible to pave your own road that leads you there. It's a monstrously tough road, but it's your road.

DREW CONWAY Head of Data at Project Florida

Human Problems Won't Be Solved by Root Mean-Squared Error

After graduating with degrees in both computer science and political science, Drew found himself working at the intersection of both fields as an analyst in the U.S. intelligence community, where he tried to mathematically model the networks of terrorist organizations.

After spending a few years in DC, Drew enrolled in a political science PhD at New York University. It was here that he drew up his famous Data Science Venn Diagram. It was also during this time that he co-founded Data Kind, a nonprofit organization which connects data experts with those who need help. After a stint at IA Ventures as their Data Scientist in Residence, Drew joined Project Florida as Head of Data, where he uses data science to give individuals better insights into their health.

Drew is also the co-author of the O'Reilly book, Machine Learning for Hackers.

Your data science Venn diagram has been widely shared and has really helped many people get an initial sense of what data science is. You created it a long time ago, back in 2010. If you had the chance to create it again today, would you change any part of it?

Quite a lot. I can speak a little bit about the history of it which I think is probably less glorious than people know.

I was a graduate student at NYU and was a teaching assistant for an undergraduate class in Comparative Politics. As a teaching assistant in those classes, your mind wanders because you already know the material.

It was 2010, and the idea of data science was much more primordial. People had less of a sense of what data science was. At that time I was thinking about the definition of data science. I had been speaking to people like Mike Dewar, Hilary Mason and some other people in New York and was influenced by their ideas and some of my own and came up with the definition while sitting there in class.

The original Venn diagram I made on data science, which ended up becoming quite well-known, was drawn using GIMP as the editor — the simplest, cheapest program in the world. But I'm very happy that it seems people have attached themselves to it and it make sense to them.

What has become more apparent to me as the years have passed is that the thing missing from it is the ability to convey a finding, or relevant information once an analysis is complete, to a non-technical audience. A large amount of the hard work that most data scientists do is not necessarily all data wrangling and modeling and coding. Instead, once you have a result, it's about figuring out how to explain that result to people who are not necessarily technical or who are either making business decisions or making engineering decisions.

Really, it's all about conveying a finding. You can use words to do that, you can use visualization to do that, or you can develop a presentation to do it. A well-rounded data science team will have someone who is very competent at this. If your organization is making decisions based on your analysis, you need to be sure they understand why.

A large amount of the hard work that most data scientists do is not necessarily all data wrangling and modeling and coding. Instead, once you have a result, it's about figuring out how to explain that result to people.

This echoes parts of what we've heard when we talked with Hilary Mason and Mike Dewar. Both of them emphasized the storytelling part and how to carefully communicate the analysis part.

It's something that receives the least amount of thought, but turns out to be one of the most important things once you're doing this in the wild. Even the people who have had success in data science up to this point have just been naturally good at it, whether they were blogging about it or giving good presentations. Both Mike and Hilary are examples of people who are good at doing that. They are naturally good at it. People who are not naturally good at it can learn about it through coaching, and mentorship.

In just the same way, if you're not a good coder you can become a better coder through coaching and mentorship.

You said on a Strata panel: "Human problems won't be solved by root mean square error." What did you mean by that?

I think when people think about data science, or even machine learning applied to data science, people think that we have a well-defined problem, and we have our data set. We need to find a way of taking that problem and that data set and producing an answer that is better than the one that we currently have.