

How to Become

FLUENT IN DATA SCIENCE

by

SHARP SIGHT

TABLE OF CONTENTS

| | |
|--|----|
| Table of Contents | i |
| Data science is one of the most valuable skills of the 21st century | 1 |
| Why you need to become “fluent” in data science | 3 |
| How to master data science in less than 3 months | 12 |
| The secret to memorizing data science syntax | 29 |
| Discover our step-by-step system for memorizing syntax and becoming “fluent” in data science | 40 |

DATA SCIENCE IS ONE OF THE MOST VALUABLE SKILLS OF THE 21ST CENTURY

So you want to learn data science. That's awesome.

Right now, the world has more data than we know what to do with. Some writers have called it “the data deluge,” which sounds hyperbolic, but it's fairly accurate. Corporations, non-profits, scientists, even individuals now have the ability to capture massive amounts of data.

Meanwhile, the world desperately needs insight. The world is overrun with problems waiting to be solved, we just need *insight into how to fix those problems*.

So we have plenty of data. And lots of problems waiting to be solved.

What we don't have is enough people to analyze that data. We need insight, but don't have enough people to produce it.

That's where you come in. Data science is one of the biggest opportunities of this century. If you can master the skills of data, you'll not only be able to add massive value to the world by understanding and fixing complex problems, but also capture some of the value you create (in the form of profit).

Want to change the world? Want to create real value and build wealth?

Learn data science.

WHY YOU NEED TO BECOME “FLUENT” IN DATA SCIENCE

Here at Sharp Sight, I frequently use the word “fluent” to describe an ideal skill level with data science.

When you join our courses, our goal is to help you become “fluent” in data science, or “fluent” in a particular skill.

We want to help you become fluent in data science as fast as possible.

I think most people intuitively understand this idea, but I want to explain a little bit about what I mean when I use that word in the context of programming and data science.

FLUENT: A TRADITIONAL DEFINITION

Let’s start by taking a look at a traditional definition of the word.

Here is a definition taken from Wiktionary.org:

Adjective [[edit](#)]

fluent (*comparative* **more fluent**, *superlative* **most fluent**)

1. That **flows**; flowing, **liquid**. [[quotations ▼](#)]

fluent handwriting

2. (*linguistics*) Able to use a language **accurately**, rapidly, and **confidently** – in a *flowing* way. [[quotations ▼](#)]

She's **fluent** in French.

Usage notes [[edit](#)]

In casual use, “fluency” refers to language *proficiency* broadly, while in narrow use it refers to using a language *flowingly*, rather than haltingly.

Take a look at that second entry.

Fluency indicates an ability to perform “accurately, rapidly, and confidently.”

That is *exactly* what you’re looking for as someone using data science.

You want to be able to use data science techniques “accurately, rapidly, and confidently.”

Moreover, you want to be able to perform “in a flowing way.”

This is in contrast to many people who *lack* fluency. Many wannabe programmers and data scientists write code in a way that’s slow, strained, and full of breaks to look up syntax.

Having said that, two of the concepts in the definition above are important, and I want to drill in to them a little bit:

- accuracy
- speed

If you want to do data-related work, you need accuracy and speed. Let's break this down.

FLUENCY IMPLIES ACCURACY

Importantly, fluency implies *accuracy*.

Just like a spoken language, when you're fluent in a programming language, you have the ability to recall and use the correct commands and syntax.

Again, a definition from Wiktionary is instructive:

accuracy (*countable and uncountable, plural accuracies*)

1. The state of being **accurate**; being free from **mistakes**, this **exemption** arising from **carefulness**; **exactness**; **correctness** [quotations ▼]

*The value of testimony depends on its **accuracy**.*

2. Exact **conformity** to **truth**, or to a **rule** or **model**; degree of conformity of a measure to a **true** or standard **value**.

One particular part of this is helpful.

Accuracy is the state of being “free from mistakes.”

Do you want to use data science techniques? Do you want a data science job or an analytics job? Do you want to get paid money for these skills?

Then you had better be able to write the code “accurately” which is to say, relatively free from mistakes.

To be clear: I'm not talking about absolute perfection. Everyone makes a mistake from time to time. But if you want to get paid to do data work, you need to actually be able to write the code.

You'd think this would be obvious, but many wannabe data scientists actually can't write data science code.

For example, an acquaintance recently told me that he had “taught himself” Pandas and machine learning in Python. But he added that he couldn't remember the difference between the `.loc[]` and the `.iloc[]` functions. So instead, he just found workarounds for everything using for-loops and other techniques from Base Python.

This guy wanted to use data science techniques (in fact, he was the founder of a defunct AI startup). But instead of learning, memorizing, and mastering the essential syntax, he resorted to work-arounds and cobbled-together code.

This is sort of like a person who wants to learn Spanish for a trip to Mexico, but never learns essential Spanish vocabulary and grammar.

If you want to be a data scientist, or you want to use data science tools in your work, you cannot be like this.

You need to commit to learning the tools and syntax, so that you can recall the syntax accurately, and ultimately apply it.

FLUENCY IMPLIES SPEED

The other main part of fluency is speed.

Let's go back to the example of a spoken language, like Spanish.

If you want to have a conversation in Spanish, you need to be able to speak the language with some amount of speed. The words need to flow out quickly, one after another. If you need to take a minute to look up every other word in a dictionary, any conversation will be slow, strained, and difficult.

It's very similar with data science.

If you have to look up syntax every 4 or 5 minutes, progress on any project you work on will be extremely slow.

That won't cut it in a real job.

I used to work at Apple. Before Apple, I worked at a large American bank.

I can tell you with some confidence that doing data science work in a real job often has serious time constraints.

On more than one occasion, I was asked to produce a polished, informative, accurate deliverable with only a few hours notice (a deliverable going to high-level executive teams).

Within only a few hours, I needed to get the correct data, clean it, draft the deliverable, iterate a few times to make sure that it met the needs of my team, and then polish it up for presentation.

In situations like these, the deadlines were extremely tight.

I didn't have time to look up syntax on Google. I didn't have time to try to remember how to use a particular technique. There was only time to execute, and I had to execute fast.

I got the work done ... beautiful deliverables on deadline.

I was able to do the work, because I knew the techniques backwards and forwards, and I was able to use those techniques quickly and accurately.

That's why they paid me 6 figures.

If you want to get a similar, highly paid data job, that's what you need.

You need to be able to do the work.

Most of the time, you need to be able to do it fast.

You need to be "fluent" with the important techniques.

FLUENCY ENABLES HIGH PRODUCTIVITY

In the end though, “fluency” in a programming language is about *productivity*.

If you’re working in a data-related job, you’ll be given projects and tasks that will require you to use data science syntax to get things done. The expectation is that you can actually produce deliverables that create value for the business.

You need to be *productive*.

As I already mentioned, this means being able to produce valuable deliverables, created with accurate, well structured code, within reasonable timeframes.

To do this, you have to know the skillset well enough to get things done, which is to say, a sort of “fluency” with the required skills.

CAN YOU ACTUALLY DO THE WORK, QUICKLY AND ACCURATELY?

You have to ask yourself a question:

Can you actually do the work? Are you actually productive?

If you're still looking up code snippets every 5 minutes, you have to ask yourself if you could actually cut it in a real job.

To be clear: it's normal to look something up every now and again. Here, the analogy with writing is instructive ... great writers still use dictionaries and thesauruses. But a great writer will not need to look up a word every 5 minutes. Great writers know the essential vocabulary of a language. They know most of the grammar. And they can produce good, clear sentences in the language in which they are writing.

It's the same thing in programming and data science.

You need to be able to actually do the work.

You need to be productive.

You need to know the essential syntax, and you need to know how to apply that syntax to produce deliverables.

You need to be "fluent" in your programming language of choice.

HOW TO BECOME "FLUENT" IN PYTHON

The subject of becoming "fluent" is somewhat more complicated, and I'll leave a full explanation for a different blog post.

But, at a high level, becoming fluent in Python can be divided into a few major steps:

- master the essential “vocabulary”
- learn how to put the “vocabulary” together
- work on projects to integrate it all into a coherent whole

When I say “essential vocabulary,” I mean foundational data techniques like adding new variables, **subsetting rows**, **retrieving columns**, as well as creating essential charts and graphs like **scatterplots** and **bar charts**.

Ultimately, if you want to master these techniques fast, you need a system.

You need a system for practicing individual commands and functions.

And you need a system for learning how to put individual pieces together.

And finally, you need to know how to work on projects (there’s a bit of an art to selecting good projects when you’re trying to learn).

Here at Sharp Sight, we have courses that cover all of these things. We can show you how to memorize syntax within only a few weeks. We can show you how to put syntax together to do data science work. And we can instruct you on how to select and work on good projects.

HOW TO MASTER DATA SCIENCE IN LESS THAN 3 MONTHS

For some people, learning the basics of data science takes years.

Now, I'll admit that if you eventually want to be one of the 'best of the best,' you *will* have to put in years of effort.

But to master the basics? Years? Just to get the foundations?

It should **not** take that long.

Any smart, dedicated person should be able to master the basics of data science within only a few months.

... And I mean *master*. I firmly think that it's possible for a motivated data science student to learn the basics so well that he or she can write the basic syntax 'in their sleep.' A good student should be '**fluent**' in the basics within *months* not years.

There's a good analogy with spoken languages. A smart, dedicated student with a good learning plan should be conversational in a language like Spanish within 8 to 12 weeks.

Similarly, a good data science student studying R or Python should be 'conversational' (i.e., fluent in the basics) within 8 to 12 weeks.

Yet, as I've already said, it takes most people years to learn the basics.

What's going on here?

To put it bluntly, *most people are terrible at learning*. They don't know how to learn new skills quickly, efficiently, and effectively, so they take much longer than they need to.

On the other hand, if you really know "how to learn," it's possible to learn the foundations of data science, very, very quickly.

Here's how.

The 5 Step Plan to Master Data Science *FAST*

To rapidly master data science, you need to follow this 5 step plan:

1. Break it down
2. Figure out what to do, and what not to do
3. Design a plan
4. Learn
5. Practice

Let's dive into each of these.

1. BREAK IT DOWN

To learn anything very quickly, you need to break it down into small components.

For example, to rapidly learn a spoken language (like Spanish), you'll want to break it down into very small units first: words. To be clear, *which* words you learn is important (we'll talk about that in a moment), but this is still a critical step. You might even go a step further and break words down into sounds (i.e., phonemes) so that you first learn the unique sounds of the particular language.

A very similar process applies to data science. To rapidly master data science, you need to break data science down into smaller sub-disciplines. Furthermore, those sub-disciplines can be broken down into a set of skills and techniques. Going a step further, all of those techniques can be broken down into small learnable units that you can practice (we'll talk about practice later).

At a very high level, you can break down basic data science into the following sub-disciplines:

- Data visualization
- Data manipulation
- Data analysis

I'll add that there are many 'special topics' in data science that aren't exactly covered by these categories. However, speaking at a very high level, these categories account for 'the basics' that you need to know.

2. FIND OUT WHAT TO DO, AND WHAT NOT TO DO

After you break things down, you need to figure out what to do, but also what not to do. To rapidly learn data science, it's critical to select the right material and distinguish between what to do and what not to do; you need to distinguish between what's really important and what is unimportant. Figuring out what *not* to do is perhaps the more important of the two. When most people begin learning, they try to learn way too much. More often than not, this leaves people feeling overwhelmed, and it often causes them to spend time on topics that are not necessary.

Let me give you an example. As of 2017, there are over 10,000 R packages. You read that right. 10,000. Realistically, you will never learn all 10,000 packages, let alone *master* all 10,000 packages.

I should also point out that there is a lot of redundancy among these packages. In R, there's often more than one way to do things. For example, to perform data visualization you can use the `plot()` function from base R, but there's also several other packages and **tools for visualizing and plotting data**. Do you know which tools are the 'best'? Do you know which packages you need to learn, and which ones you should skip?

To **master R** quickly and efficiently, you need to be able to select a very small number of packages among these 10,000. You need to choose what to learn and what to ignore.

Additionally, once you select the best packages to learn, you need to choose what to learn *within* those packages. Even if you select the best R packages, within those packages there are tools that you absolutely need to know, but other things that you probably don't need to learn right now. Some tools and techniques are things that you don't really need, and it

would be better to wait a few months or years to learn them. Again, you need to know what to learn and what not to learn.

2b. FOCUS ON FOUNDATIONS

In the context of ‘selecting’ the right topics to learn, I should mention that it is very useful to focus on foundational skills.

If you study top performers of all kinds, you’ll find that essentially all of them place a strong emphasis on ‘the basics.’ Top performers focus on foundations.

To master data science, you need to do this. You need to master foundational techniques before you move on to advanced topics.

So as you’re learning data science, this means that you need to master 3 critical areas:

1. Data visualization
2. Data manipulation
3. Data analysis (AKA, exploratory data analysis)

A big mistake that beginners make is that they jump into advanced topics too soon, before they’ve mastered these foundations. For example, new

data science students **get excited about machine learning** and want to start by learning machine learning. These students would be much better served by mastering the critical, foundational tools first like data visualization and data manipulation.

To put this another way, by focusing on the foundations, you put yourself in a position to rapidly master more advanced topics later. If you master the critical foundations first, you will be better equipped to learn more advanced topics later, and you will do so at a much faster rate.

Do you want to be a top-performing data scientist? Master the foundations first.

That invites the question: What are the foundational data science skills in R?

The following is a quick list:

- Basic visualizations: bar charts, line charts, histograms
- Manipulating colors in charts
- Visualization formatting (I.e., how to format your charts to make them look good)
- String manipulation

- Date manipulation
- Data reshaping (i.e., transforming from 'wide' format to 'long' format and visa-versa)
- Adding variables
- Removing variables
- Aggregating data
- Reading in data (from external sources)
- Working with factor variables (e.g., ordering factors, re-naming factor levels, re-categorizing factor variables, etc)

This is a fairly high-level, list but it is a pretty good list of the things that you absolutely need to know. If you can't do these, do not move on to more advanced topics. Don't get shiny object syndrome. If you can't do all of these skills, you need to go back and learn them right now.

Moreover, if you want to be a top performer, you should be able to do these things without even thinking about them. Top-level data scientists can do these things 'in their sleep.'

3. DESIGN A LEARNING PLAN

Once you select the right topics, you need a learning plan. Specifically, you need to sequence the topics in the best order.

One reason for this is that some topics are dependent on others. For example, in math, you need to know arithmetic before you learn algebra, and you should know algebra before you learn calculus.

Data science is similar. There are some topics that are dependent on other topics. For example, I have routinely said that the prerequisites for machine learning are data visualization, data manipulation, and data analysis. To effectively learn ML, you need to be able to wrangle a data set, clean it, and visualize it. So, if you want to eventually learn ML, you need to start with visualization and manipulation first.

There are also more subtle sequencing considerations. I personally believe that it's best not to start with data manipulation, because by definition, data manipulation is required for more complex and messy data. There are much better data science topics to start with.

So in order to rapidly master data science, you need to be able to sequence the material in the optimal order, so you learn the right things at the right time.

4. LEARN

Once you have a plan, you can start learning.

However, learning data science topics can be challenging. Many topics can be very confusing. For example, when I try to learn something new, I typically buy 5 or 10 books on the topic, but I often find that many of the books don't explain the topic in a clear way. Frequently, out of 5 or 10 books on a subject, 1 or 2 books are dramatically better at explaining that subject.

How quickly you learn can depend critically on the quality of your learning materials.

5. PRACTICE

Having said that, learning is not the final step.

Once you learn the basic concepts and techniques, you need to **practice**.

You need to practice techniques and review concepts until they are 'second nature.'

This is an extremely important point. There is absolutely a difference between learning something once, and remembering it in the long run.

Let me give you an example. If I show you a video right now that explains the `ggplot()` function, you'll probably understand how it works. The syntax is fairly easy to understand once someone breaks it down and explains each piece of the syntax.

Next, if I ask you to write some simple `ggplot()` code, you'll probably be able to do that too. For example, let's say I ask you to create a simple scatter plot:

```
ggplot(data = diamonds, aes(x = carat, y = price)) +  
  geom_point()
```

If I ask you to do something simple, like typing the code into R studio, you'll probably be able to do it.

But what if I ask you to do it again 3 hours later? If I ask you 3 hours later to write that code from memory, there's a good chance that you won't be able to do it.

Why?

Because we forget. The human brain *naturally forgets*.

However, there's a way to fight this. You can halt this forgetting process by practicing. Specifically, you need to repeat and review what you've learned.

Practicing techniques and repeating what you learn will enable you to remember those things in the long run. Moreover, as you practice, you will become more 'fluent' in those techniques. You will execute those techniques more quickly and with less hesitation the more you practice.

An added benefit of practice, is that effective practice methods help you become a "top performer." In fact, research has shown that elite levels of performance are strongly tied to deliberate practice. If you want to be a top performer, practice is critical.

I've said several times that to be a top-performing data scientist, you need to be able to execute the basic techniques 'in your sleep.' You should be able to do essential data visualization and data manipulation 'with your eyes closed.'

You can achieve this level of mastery by practicing the right way, using good practice systems.

EFFECTIVE LEARNING HAS LARGE BENEFITS

Learning data science quickly can be a massive benefit.

Let's put some numbers around it.

Let's say that two people are learning data science: you and someone else. The other person learns extremely inefficiently, and takes 1000 hours to master the basics. But you learn the basics much faster, in about 200 hours.

The difference, 800 hours, is a really big difference. Again, we can put some numbers around this.

If your free time is worth only \$20 an hour, that time savings of 800 hours translates to \$16,000.

But let's say that you **really** value your time. (You *should* value your time.

Time is the only resource you can't get back.) If you value your time at \$50 an hour, the time you save by learning more efficiently amounts to a staggering \$40,000.

Now these are just example numbers for illustration, but you get the idea. Being highly effective and efficient in learning data science has massive benefits.

There's actually another benefit of being an effective learner. If you really know how to learn, you'll not only learn more quickly, but you'll attain higher levels of proficiency and mastery.

If you are highly effective at learning data science, it becomes much easier to become a 'top performer.'

It really pays to be a top performer. The reason for this is that the best people in tech often receive outsized gains. The best people disproportionately get the best jobs, highest salaries, and best perks.

You've probably heard about the mythical '10X developer' ... people who are 10 times more productive. These top performers often get the lion's share of rewards in the tech industry.

In this regard, the tech world is sort of like sports. Think about basketball: you have a few guys like Kobe, and Michael, and LeBron who made millions of dollars per year. Then you have a larger set of guys in the NBA

who have less skill and make dramatically less money. Even worse: for every guy who made it into the NBA, there are hundreds who didn't.

The tech world is similar. Top performers get the lion's share of rewards, while less skilled performers make dramatically less (and many people struggle to break into the industry at all).

GET EXPERT HELP

It definitely pays to rapidly master data science. It pays to master the essentials as quickly as possible.

Let's review how you can do that:

1. Break the subject down
2. Sequence the material in the right order
3. Study the material
4. Practice

At a high level, that's really it (although, the key to getting it right is in the details).

If you can apply this learning process to data science, you'll accelerate your learning and increase your chances of success.

But if you really want to accelerate your progress and learn as quickly as possible, there's one more thing you can do.

You can get guidance from an expert.

Top performers understand that they can save massive amounts of time by getting advice from people who have already mastered the topic.

Learning a new subject is time consuming, because you need to figure out what you need to learn, design a learning plan, sequence the material, and all of the things I've already talked about. But you need to do these things *without a clear understanding of the subject*. It's like trying to find your way through a jungle, alone, without knowledge of the terrain. You'd be well served by getting a guide ... someone to safely and quickly get you to your destination.

A data science mentor can tell you exactly what to do: "learn this first, learn this second, focus on x-y-z, don't bother learning that topic, etc." A good teacher can dramatically accelerate your learning, because they remove the burden of having to find the path on your own.

For example, when ‘superlearner’ Tim Ferriss wants to learn something, he finds a world class expert on the subject and gets help. Ferriss knows that he can dramatically accelerate his learning by getting expert advice.

If you want to rapidly master data science, you need to do the same. While it is possible to learn data science on your own, you can learn much, much faster with expert guidance. That might include finding a data science mentor, but it could also mean a good data science course.

THE SECRET TO MEMORIZING DATA SCIENCE SYNTAX

A while ago, I received an email from a Sharp Sight reader. The author of the email is having trouble learning data science in R. He's taken several data science courses, but still has trouble with critical data science skills. Here's an excerpt from his email:

Miguel

1:24 PM (1 hour ago) ☆

I began doing [REDACTED] and [REDACTED] courses but found that I could take one of them, do all the excercises, and then do nothing for one month: when I came back to the next module I had almost forgotten all the R programming skills.

I've redacted the company names, but I'll tell you that he took courses by two very well known data science training companies.

Does this sound familiar?

What Miguel is experiencing is unfortunate, but all too common.

I regularly receive emails from people telling me that they've taken courses about data science, but still can't write code. They initially learned the material, but eventually forgot everything.

This is a critical problem on the path to mastering data science. If you don't practice what you learn, you will forget.

WHY YOU'RE FAILING TO MASTER DATA SCIENCE SKILLS

The critical problem for students like this isn't necessarily learning.

Many data science skills are easy enough to understand. **dplyr** is fairly

easy to understand. Although **ggplot2** has a few quirks, it is also

relatively easy to understand. You can watch a few videos and learn how

these tools work. It's pretty easy to type in some code a few times and get

these tools to work on your own computer. These tools are easy to learn.

Moreover, there's plenty of material out there to help you learn. Although

the quality definitely varies, you can find information about almost anything

on the internet. Even here at Sharp Sight, we give away tons of free

tutorials to help you learn data science.

There's plenty of tutorials to help you learn, and many techniques

(especially the "basics") are easy enough to *learn*.

But they aren't necessarily easy to *remember*. That's because the human

mind naturally forgets.

Have you ever had the experience of asking someone their name, only to forget it 5 minutes later? Have you ever learned a fact from a book, but then forgotten it the next day?

This is natural. It's completely normal. The human brain naturally forgets.

So on the path to mastering data science skills, the roadblock typically isn't learning so much as the problem is forgetting.

Forgetting what you learn is stopping you from mastering data science.

YOU NEED TO MASTER DATA SCIENCE, NOT JUST “LEARN” IT

At this point, I want to re-emphasize the importance of mastery.

Here at Sharp Sight, we routinely stress the importance of mastery. You **need to master data science skills**. It's not enough just to “sort of know them.”

This is particularly true of basic syntax. Like it or not, R syntax (or the syntax of another data science language like Python) gives you the tools to “get things done.” You need syntax to create deliverables and ultimately create business value for your clients.

If you get hired, it's not good enough to just have a vague memory of the syntax. It's not good enough to be a cut-and-paste coder who goes to Google every 5 minutes to look up how to write a simple piece of syntax. To work as a data scientist, you need to be relatively "fluent" in writing data science syntax. If you work in R – like we often do here at Sharp Sight – that means that you need to be relatively fluent in writing R code. You need to have a strong working command of the syntax. This means that you should be able to write the code for basic tools rapidly and from memory. Mastery isn't only important for "getting things done." Mastery of basic syntax is also important for getting hired. To get hired as a data scientist, you need to be somewhat fluent. At the very least, you'll need to be fluent in the "core" skills.

What do you think will happen if you walk into a data science interview and someone asks you to write the code for a scatterplot ...

Do you think that it's enough to tell them, "yeah, I watched a video on that once and typed the code a few times, but I don't really remember how to do it"?

Uh ... you won't get the job.

To get a job and to be effective in a job, you will need to master “core” data science skills. You will need to have a strong fluency in the syntax for data manipulation, data visualization, and data analysis.

If you want a data science job, and perform well in that job, mastering the skill set is critical.

TO MASTER DATA SCIENCE, YOU NEED TO SOLVE THE PROBLEM OF “FORGETTING”

To achieve this level of mastery – to remember the syntax and write code from memory – you need to *stop forgetting*.

Ask yourself, how much time have you spent trying to learn and master data science? Do you remember what you’ve learned?

My guess is that out of all of the syntax you’ve “learned” by reading a blog post or cutting-and-pasting code, you remember only a small fraction.

And I bet it’s frustrating. You’re wasting time! If you weren’t forgetting so much of what you learn, you’d probably have mastered R and Python by now!

This is the problem.

And to be clear, I'm not trying to berate you. We *all* forget. This is just how the human mind works. But like it or not, it's a problem.

To master data science, you need to stop forgetting what you learn.

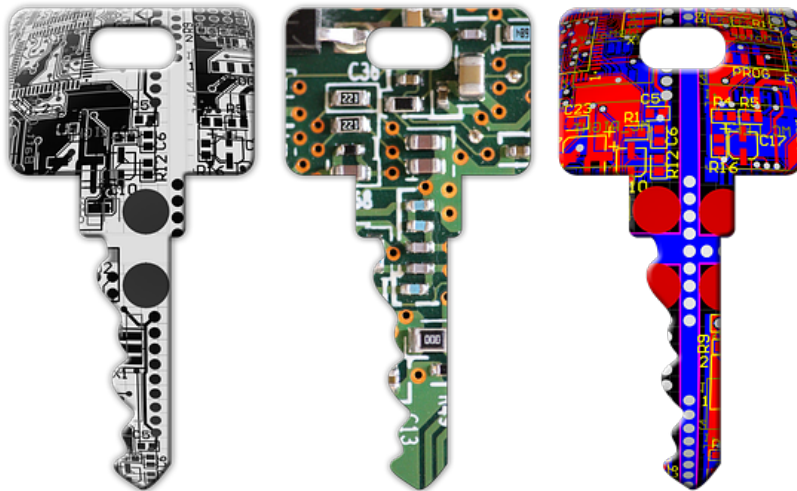
REPETITION IS THE KEY TO MEMORIZING SYNTAX

I'll simply tell you the secret to stop forgetting.

It's the secret to remembering what you've learned. It's the secret to mastering data science syntax, or any skill for that matter.

The secret to remembering what you learn is *repetition*.

Repetition is the key. You need to repeat your practice.



TOP PERFORMERS REPEAT THEIR PRACTICE ACTIVITIES

If you look at top performers of all kinds, they repeat their practice

activities. Top performers know that the **secret to mastery is repetition**.

Relentless repetition. Repetition until the skill becomes second nature.

Repetition until you've achieved "unconscious competence." Then a little more repetition, just to be sure.

ATHLETES REPEAT THEIR PRACTICE

In his book *Relentless*, elite basketball trainer Tim Grover – the famous trainer of Kobe Bryant and Michael Jordan – explained that guys like Michael and Kobe practiced basketball shots over and over again. He notes that on some days, an elite player might practice a single shot thousands of times in a single day. Thousands of repetitions. Michael Jordan wasn't born the best, he earned it with millions of repetitions over years of training.

MUSICIANS REPEAT THEIR PRACTICE

We see something similar among musicians. Elite violinists and musicians are known to practice a single “phrase” of a song over and over. At first, they might just practice to learn and memorize the phrase. At first, they’re just trying to execute the musical phrase correctly. But as they move on, they try to refine it and add a bit of finesse. For musicians, the process of memorization and refinement comes from careful repetition.

From personal experience, I can also tell you that learning a musical instrument is strongly dependent on repetition. You have to repeat scales and techniques until you can execute them fluidly. You repeat them until they are second nature. Eventually, you repeat them until you never forget them (I can still play songs that I learned years ago).

NAVY SEALS REPEAT THEIR PRACTICE

Navy SEALs – who are renowned for their ability to succeed in tough battle conditions – are also relentless trainers. They succeed because they

prepare. SEALs make a point to train their skills until they have them completely memorized. Then they train some more.

Here's a quote by retired Navy SEAL, Rob Roy:

“Repetition leads to memorization and memorization leads to instinct.

Therefore, one must train and train their skills until they know a procedure cold. And then they must train some more.”

I love this quote. It's nominally about training combat techniques, but it applies to almost any skill. The secret to becoming great at anything is *repetition*. It doesn't matter if you want to be a soldier or a data scientist.

If you want to walk into a data science interview and astound them with how well you know the code, you need to practice syntax. You need to repeat the syntax you learn. If you want to be fluent in writing data science code, and if you eventually want to be one of the best, you need repetition.

TO MASTER DATA SCIENCE, YOU NEED TO REPEAT YOUR PRACTICE

To be a great data scientist, you need to have some core syntax memorized. You need to essentially memorize the code to perform basic techniques: data manipulation, data visualization, data analysis.

Will you occasionally forget a few things. Yes. Sometimes. But there are some things that you should know cold. For example, you should be able to write the code to filter a dataset without hesitation. You should be able to write the code to create a scatterplot without flinching. You should have these memorized.

Being a great data scientist means that you need to have some things memorized. That's what it takes to get things done.

To memorize the syntax, you need to repeat it. Over and over. And then train a bit more ...

REPETITION REQUIRES A PRACTICE SYSTEM

Repeating your practice activities requires you to be systematic. You can't be haphazard in how you train. If you don't train in a systematic way, you're greatly increasing your chances of failure.

This shouldn't surprise you. Many skills have practice drills that are used to help people learn and master that skill.

For example, musicians often have **drills to practice** scales, musical phrases, chords, and arpeggios.

Basketball players have passing drills or drills to practice a particular shot over and over.

The best performers are systematic in how they train.

If you want to be great data scientist, *you* need to be systematic as well.

The problem here is that almost no one trains this way. As I've noted before, many new data science students use the "jump in and build something" method of data science training. They just **jump in and start a big project**. This would be like trying to learn basketball by just playing a few games every now and again. Yes, you can learn a little, but you'll never be as good as someone who drills and trains the details to mastery. To be clear, data science projects can be good later in your data science training. Projects are a great way to integrate what you've learned, after you've mastered the basics. But projects are completely inefficient for mastering basic syntax. The reason is that when you work on a project, there's not enough repetition. In a project, there are many pieces of that you will only use once or twice. That's not enough repetition to memorize the code snippet.

No. Projects aren't the best way to master data science because they don't give you enough repetition. You need a structured practice system.

DISCOVER OUR STEP-BY-STEP SYSTEM FOR MEMORIZING SYNTAX AND BECOMING “FLUENT” IN DATA SCIENCE

In our paid training courses at Sharp Sight, we will show you just such a practice system. Our courses teach you a specialized training methodology for repeating data science syntax until you have it memorized.

Many of our students report becoming “fluent” in writing data science code. They write code from memory.

If you want to learn how to practice data science ... how to repeat your data science practice so you memorize the syntax, sign up for our email newsletter and enroll in our course when it opens.

Our training courses only open up a few times per year, and the enrollment details are sent exclusively to people on our email list.