**The 5 Most Critical Insights You’ll Gain in Your First 5 Years as a Data Scientist**

**Keep these five insights in mind to advance your career from day one**



Photo by [Faye Cornish](https://unsplash.com/@fcornish?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

Your first five years as a data scientist are going to feel a lot like drinking from a firehose.

New information is going to be coming from left and right, you’re going to have to…

By the end of your first five years, you may feel as though you’ve been hit by a bus of information and learning experiences.

However, when you look back on those first five years, you’ll probably notice something pretty cool: you’ve gained several valuable insights that you can use to leverage your career in the next five years.

Be that as it may, some of us want to fast-track our learning and gather as many insights as we can before we hit that five-year mark. Who doesn’t want to begin leveraging their career right now?

The following five insights are those that you can expect to gain in your first five years as a data scientist.

**1. Automation is (almost) always worth it**

One of the biggest complaints of data scientists is that the work can be monotonous and repetitive.

Therefore, why not skip to the good part and automate everything you could do in your sleep?

If you love 75% of your job, don’t quit because the other 25% is boring. Automate the boring stuff so you can use your time more effectively to tackle the exciting stuff.

According to one of the most quoted productivity conventions, roughly [20% of the work you do yields 80% of the results](https://www.aafp.org/pubs/fpm/issues/2000/0900/p76.html#:~:text=Simply%20put%2C%20the%2080%2F20,80%20percent%20of%20the%20results.). This means that the other 80% of your time is spent doing tasks that yield little to no results.

Why waste your time (and your company’s time, for that matter) on tasks that yield little impact?

Automating tasks can be as simple as setting up automatic reminders for yourself, uploading data, converting file types, backing up files, and much more.

Your ability to automate is only as limited as your creativity. Just make sure that you’re able to admit when the automation goes too far.

**2. Documentation goes far beyond comments**

Documentation can start or end your career as a data scientist.

Without good code documentation, code cannot be sent to engineers to be integrated into an existing process. Nor can it be reworked by an intern ten years down the road who’s told to modernize it.

[Good code documentation](https://towardsdatascience.com/how-to-write-good-code-documentation-for-data-scientists-c9940aebb4f0) not only tells those using it what it does and how it works, but it also provides a look into the original data scientist’s thought process about why the code had to be written a certain way.

Code documentation should [go far beyond comments](https://towardsdatascience.com/how-to-write-good-code-documentation-for-data-scientists-c9940aebb4f0) and should seek to include flow charts, full README files with step-by-step descriptions, and the provision of contact information if none of your documentation can help.

Many of the other data scientists you come across in your first five years won’t have entered the field with a software engineering or computer science degree. Therefore, you all must work together to produce code documentation to help others understand your work.

Writing solid documentation is often an afterthought once the major business problem has been solved. However, by looking to the future, data scientists can ensure that their company benefits from the impact first delivered from their analysis for years to come by simply writing good code documentation.

**[How to Write Good Code Documentation for Data Scientists](https://towardsdatascience.com/how-to-write-good-code-documentation-for-data-scientists-c9940aebb4f0" \t "_blank)**

**[A crash course on the best practices you need to ensure that everyone understands the code you write.](https://towardsdatascience.com/how-to-write-good-code-documentation-for-data-scientists-c9940aebb4f0" \t "_blank)**

[towardsdatascience.com](https://towardsdatascience.com/how-to-write-good-code-documentation-for-data-scientists-c9940aebb4f0" \t "_blank)

**3. Good coding practices make you a true professional**

Most data scientists don’t enter the field with software engineering or computer science backgrounds. If anything, most data scientists you see today have majors in everything from physics to nursing to industrial engineering to political science.

Data scientists are infamous for writing code that has a severe [absence of functions, unclear variable names, spaghetti code, not a single hint of a unit test, and a severe lack of style](https://towardsdatascience.com/software-engineering-best-practices-for-data-scientists-4c199ede6e03).

Therefore, the value of a data scientist is in their ability to write clean code that can be [updated, debugged, and moved into a production environment with few swear words coming from the engineering department](https://towardsdatascience.com/software-engineering-best-practices-for-data-scientists-4c199ede6e03). It can’t be stressed enough how clean code can help a company run efficiently. The ability to assure stakeholders that you can get results rests on your ability to write clean code that runs correctly the first time. Not only that but writing clean code will make you a favorite with everyone you work with throughout the rest of your career.

Think I’m kidding? Talk to any software engineer about a time they had to work with a data scientist’s code and you’ll quickly see that the horror stories outnumber the fairytales.

Good coding practices are what separate professionals from amateurs.

**[Software Engineering Best Practices for Data Scientists](https://towardsdatascience.com/software-engineering-best-practices-for-data-scientists-4c199ede6e03" \t "_blank)**

**[A crash course on how to bridge the gap between data science and software engineering.](https://towardsdatascience.com/software-engineering-best-practices-for-data-scientists-4c199ede6e03" \t "_blank)**

[towardsdatascience.com](https://towardsdatascience.com/software-engineering-best-practices-for-data-scientists-4c199ede6e03" \t "_blank)

**4. Being a great storyteller makes you indispensable**

Everyone can write code and conduct data analyses.

Everyone can conclude on what the data is telling them.

Everyone can produce beautiful visualizations.

But can everyone tell a story that moves stakeholders to action? Can everyone develop stories that will tell everyone at the table how we’ve gotten to where we are, what we’re currently doing, and how we will progress forward in the future?

Data scientists who can tell great stories are indispensable. Telling stories goes further than that though, and encompasses all forms of communication that are vital to data scientists.

The ability to communicate is not a skill held by all data scientists and is one that will help you advance rapidly in your career. Therefore, you must practice communicating at every chance you get. Develop stories for your stakeholder presentations that have a clear beginning, middle, and end. Shape your data analyses to provide a clear story of what that data is currently telling you. Practice producing stories of what future projections could hold for your company that gives stakeholders a clear glimpse into what the future could look like.

Becoming a great communicator isn’t something you can achieve overnight, but is something that you’ll be able to do if you practice it every day.

**5. Keep it simple, stupid**

Your fancy model doesn’t mean anything if it draws the wrong conclusion.

Your complex algorithm doesn’t prove that you’re better than everyone else if it makes mincemeat of already coherent data.

Your two lines of code that replace ten lines of code don’t make you a data science god if no one else can understand what it does.

For some reason, this adage seems to go out the window at one point or another in everyone’s career as a data scientist. Whether it’s because the work is boring or you’ve learned something new, there will come a time when you no longer want to keep it simple. And that’s fine.

As long as you find your way back to keeping it simple.

Keeping it simple is what will make you a great mentor at your company, what will allow you to complete work quickly and efficiently, and what will eventually lead you to manage teams of your own.

Keeping it simple means looking for the shortest logical path between A and B. It also means being direct in your work — if something makes more sense written as ten lines of code, then write it as ten lines of code. Furthermore, it also means you’re avoiding looking for overcomplicated ways of doing otherwise simple things.

This statement should be one you learn to live by, not only in the hopes of becoming a better data scientist, but as becoming a better teammate, manager, and boss. This statement will help you stand out after only five years as a data scientist.

**Key takeaways**

* Much of the monotonous, repetitive work of a data scientist can be automated to ensure that you’re using more of your time to deliver impactful results.
* The ability to write good code documentation will ensure that your company benefits from the impact first delivered by your analysis for years to come.
* Good coding practices will make you a great teammate and a valuable asset in whichever data science position you hold.
* Everyone can write code, do the math, and complete an analysis — what will set you apart from other data scientists is your ability to tell stories and communicate effectively.
* Seeking the simple path towards completing a task will make you a great mentor at your company, will show your superiors that you’re the person to trust to get work done efficiently and effectively, and will lead you to manage a team of your own one day.