

Fresh code and chips, please

An impromptu decision reveals new angles in real estate research

Stephen Ryan



THE ANALYST



A few years ago, I heard of a survey in which people were asked what language they would most like to learn. I expected the result to be Mandarin or Spanish or similar, but in fact the younger cohort responded Python. That a programming language would even enter someone's head as an answer amazed me. I decided on the spot to learn to code.

With no background in computing, I flailed around at the start, until a friend gave me a book called *The First 20 Hours – How to Learn Anything Fast*. This self-help book explains how to quickly pick up various skills, including yoga, windsurfing, playing the ukulele, touch typing and, towards the end of the book, programming. My friend was enthusiastic and urged me to try it, but I was sceptical.

The author, Josh Kaufman, has a distinctive approach to learning. He begins by examining his own limited knowledge of whatever skill he desires to learn. He then identifies 'a loveable project' to tackle. These steps lead him from a vague idea of what is required to a clear list of key sub-skills. For coding, he knew that programmers write programmes, which suggested to him that it's a creative exercise that can be done in many ways. Using this framework, and relying heavily on Google searches,

he was sufficiently competent to 'beat some code into submission' after 20 hours.

"Within a few weeks (though not 20 hours), I was hooked on coding"

Since Kaufman's book seemed to be pitched at total novices like me, I started to grind my way through its 60-page chapter on programming. Within a few weeks (though not 20 hours), I was hooked on coding for three reasons:

1. Community spirit. Experienced developers and those learning how to code share ideas with each other via websites like Stack Overflow. The openness and unreserved willingness to collaborate is something I had never experienced before in a learning environment and I found it exhilarating.
2. Speed. Once written and working smoothly, a piece of code (that is, a programme) will churn through vast volumes of data, performing mathematical or logical or linguistic operations at lightning speed.
3. The abstract side of coding. There are intriguing concepts to be pondered (for example, the difference between true and 'truthy'). The concept of nil, a contraction of



the Latin nihil, is particularly elusive. Nil can denote a list with no entries, which is odd enough, or the object which means there is no object, which is even odder, but interesting. In the words of Sergeant Pluck in Flann O'Brien's novel *The Third Policeman*, 'That is a great curiosity, a very difficult piece of puzzledom, a snorter'.

Greatly encouraged by my speedy progress and the success of my 'loveable project' (involving recipes), I finished *The First 20 Hours - How to Learn Anything Fast* and then took an online course, followed by another one. The dots were joining up in my head, just as they do when learning a foreign (human) language. Becoming reasonably proficient in a coding language is probably harder than learning Spanish, but easier than learning German.

As I became more confident, I experimented with work tasks. In 2019, when the Didobi team was researching global capital flows for the Investment Property Forum, we had detailed data on 17,500 real estate transactions spread over 450 cities worldwide.

Our spreadsheets were struggling to handle this volume of data.

Knock, knock.
"Who's there?"
Long interval.
"Excel."

So we analysed the data using code and run time was reduced from two hours to under two minutes. Problem solved.

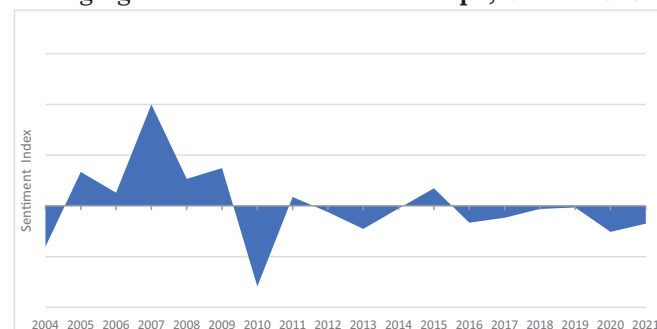
A basic knowledge of coding opens up all the possibilities of natural language processing (NLP). NLP is a branch of artificial intelligence that helps computers understand and manipulate human language. Human language is an example of unstructured data. While structured data is neatly arranged in rows and columns like a spreadsheet, unstructured data is messy. It presents the double challenge of volume and untidiness.

On volume, we often need to review long documents: FCA consultation papers, research material from trade associations, such as AREF and INREV, academic papers and the like. For this type of data we use code to summarise or otherwise extract the meaning from words. The computer assembles an extractive summary – a subset of words and sentences containing the key points, using only words found in the original data. The summary is extracted by applying rules to rank words and phrases. Armed with a summary of each potentially interesting file, we can identify those which merit closer scrutiny.

Untidiness, the second challenge with text data, arises from the inherent complexity of language: words are less straightforward than numbers. The number 6 has no past tense or infinitive, but the word 'invest' does. NLP software can handle words and phrases almost as deftly as spreadsheets can handle numbers.

It can also detect sentiment or mood by calculating the balance between positive words ('epic') and negative ones ('appalling'). The chart below illustrates the mood of the European real estate industry as revealed by the language in an annual survey conducted by the Urban Land Institute and PwC since 2004.

Sentiment in European real estate as revealed by Emerging Trends in Real Estate Europe, 2004 to 2021



Sentiment is one pattern that NLP can detect, but not the only one. NLP can also detect recurrent topics. When the Didobi team was working on a life sciences real estate project earlier this year for the Urban Land Institute, we quickly identified Labs, Leases, Locations and Lifecycle Stages (startup, spin out, venture capital) as four key themes in this area. The pattern was extracted from a dataset comprising 300,000 words derived from interviews, roundtables and analysts' reports.

While coding is a sought-after skill in its own right, at Didobi we are not developers, but rather researchers: we use code primarily to improve our research process. We generate more nuanced insights because we are less likely to overlook data patterns that are hiding in plain sight. Where there is a large body of existing research, we identify the most relevant sources with confidence. When used wisely, coding saves time, adds value and transforms data from an output to an asset.

Having some coding skills is useful in real estate and immensely satisfying at a personal level. For real estate researchers, it can bring some interesting new angles and potentially an edge over the competition. It's not just younger people entering the workforce who should learn to code – everyone should consider it. Enjoy the spirit of collaboration, the processing speed, and the abstract concepts like truthiness and nil. Give it 20 hours and do not be discouraged, because (to quote Flann O'Brien again) 'Rome wasn't built in AD'.

Stephen Ryan is a research associate at Didobi (didobi.com), specialist advisers to the real estate industry. Stephen has worked in the real estate and wider financial services industry since 1988. Most recently he worked with INREV in Amsterdam, concentrating on real estate research, corporate governance and liquidity. Prior to INREV, Stephen was an investment consultant at Mercer, where he advised institutional investors on real estate and on defined contribution investments.