The Poisson distribution: From basic probability theory to regression models

Brief introduction to the Poisson distribution for modeling count data using the distributions3 package. The distribution is illustrated using the number of goals scored at the 2018 FIFA World Cup, suitable for self-study or as a classroom exercise.

Webscraping in R with Rvest

Web scraping has become an incredibly important tool in data science, as an easy way to generate new data. The main advantage is the automation of some pretty repetitive tasks. Web scraping can also be a good way of keeping up with new data on a website, assuming it doesn’t have a big change in its HTML structure.

Clang Static Analyzer and the Z3 constraint solver | Frederic Cambus

Notes on using the Z3 constraint solver with the Clang Static Analyzer

As far as static analyzers are concerned, one of the most important point to consider is filtering out false positives as much as possible, in order for the reports to be actionable.

This is an area on which Coverity did an excellent job, and likely a major reason why they got so popular within the open source community, despite being a closed-source product.
LLVM has the LLVM_ENABLE_Z3_SOLVER build option, which allows building LLVM against the Z3 constraint solver.

**Least Common Denominator APIs** [5]

Often, our instinct is to build for optionality. What if we change databases? What if we change clouds? We target the Least Common Denominator (LCD) interface to avoid vendor lock-in and make sure our software is portable — after all, Optimization is Fragile. LCD interfaces look like targeting the S3 API, a generic PubSub implementation, or vanilla ANSI SQL.

LCD interfaces are good enough most of the time, but when we need to run a specialized workload, sometimes they don't perform how we'd like. We could solve our problem quickly by narrowing the API — coupling it to a specific cloud or managed service, but that destroys our optionality. Here, you should probably fight your instinct to stick with the pure implementation and weigh the trade-offs — how many developer-hours and pain can you save by narrowing the interface?

Optimization and optionality are inherent trade-offs. There's a way to architecture services to be efficient and generic but also practical.

**Quantum computer programming for dummies** [6]

For would-be quantum programmers scratching their heads over how to jump into the game as quantum computers proliferate and become publicly accessible, a new beginner's guide provides a thorough introduction to quantum algorithms and their implementation on existing hardware.

"Writing quantum algorithms is radically different from writing classical computing programs and requires some understanding of quantum principles and the mathematics behind them," said Andrey Y. Lokhov, a scientist at Los Alamos National Laboratory and lead author of the recently published guide in ACM Transactions on Quantum Computing. "Our guide helps quantum programmers get started in the field, which is bound to grow as more and more quantum computers with more and more qubits become commonplace."

**Create new variables from existing variables in R** [7]

Create new variables from existing variables in R. To create new variables from existing variables, use the case when() function from the dplyr package in R.
Construct a Perfect Binary Tree with given Height [8]

Given an integer N, the task is to generate a perfect binary tree with height N such that each node has a value that is the same as its depth. Return the inorder traversal of the generated binary tree.

Announcing urllib3's bounty program [9]

We've recognized that one of the biggest challenges to shipping v2.0 is not having enough time to devote to contributions. Our bounty program is hoping to spur interest from the community in the urllib3 project and fairly pay contributors for their time and experience.

The bounty program works by marking issues with bounty amounts we're willing to pay for anyone to complete an issue. Don't worry if you're not an existing contributor?new contributors are welcome and encouraged!

Learning from Failure? Nitinol Fracture Mechanics in R | R-bloggers [10]

Despite our best efforts, nitinol implants fracture and fail. Sometimes we want them to fail (on the bench, to learn).

Every Sufficiently Advanced Configuration Language is Wrong [11]

Every sufficiently advanced configuration language is the wrong tool for the job.

[...]

The logical extreme is becoming more evident?advanced configuration in general-purpose programming languages. You can see this in the emergence of Typescript for Infrastructure-as-Code. For the most basic (and human 0x777) configuration needs, there will always be simple formats?YAML, JSON, INI, etc.).

Another Exercise In Encoding Reversing | Didier Stevens [12]

In this blog post, I will show how to decode a payload encoded in a variation of hexadecimal encoding, by performing statistical analysis and guessing some of the ?plaintext?. 
I do have the decoder too now (a .NET assembly), but here I’m going to show how you can try to decode a payload like this without having the decoder.

Examples Of Encoding Reversing | Didier Stevens [13]

I recently created 2 blog posts with corresponding videos for the reversing of encodings.

The first one is on the ISC diary: ?Decoding Obfuscated BASE64 Statistically?. The payload is encoded with a variation of BASE64, and I show how to analyze the encoded payload to figure out how to decode it.

An Introduction to Python: A Language for the Ages ? The New Stack [14]

For anyone just getting into software programming, one of your best friends will be Python. Why? Python is very simple to learn and easy to implement. Even better, what you can do with this language grows as you learn more. You can start with very simple text-based applications and migrate to GUI applications and much more. And because Python is supported by most major operating systems (Linux, macOS, and Windows), you can begin your journey, regardless of platform.

Python includes support for features such as lists, tuples, functions, variables, JSON, and ranges. But where did Python come from and why is it still so important today? Let’s dig in and find out. To follow our series of introductory tutorials, start here.

How To Write Comments In Python [15]

The way you think is reflected in programming in order to convey the individual steps that you took to solve an issue utilizing a computer. Commenting your code helps clarify your thinking process, which in turn makes it easier for you and other people to comprehend the purpose of your code in the future. Because of this, you will have an easier time locating bugs, fixing them, enhancing the code at a later time, and reusing it in other applications as well.

The act of commenting is essential to the completion of any and all tasks, regardless of how little, medium, or fairly enormous they may be. It should be considered standard procedure for software engineers since it is an important component of your workflow. Without comments, things have the potential to get quite complicated very quickly. In this post, we will cover the many techniques of commenting that Python offers, as well as how it may be utilized to automatically produce documentation for your code via the use of the so-called module-level docstrings.
Source URL: http://www.tuxmachines.org/node/166483

Links: