Q&A | Webinar 22 april 2020
Vind je weg in Excel met Python en R

Vraag #1
“Is er een voorkeur voor Python of R?”

Niet zonder meer, beide talen zijn zeer geschikt om data te bewerken, te analyseren, te modelleren en te rapporteren. In de loop van de tijd hebben ze ook veel concepten van elkaar "geleend", zoals bijvoorbeeld het dataframe (een Excel-achtige datastructuur) en de mogelijkheid om meerdere commando's in een chain achter elkaar te zetten, zoals we zagen in het webinar. Maar wat moet je dan kiezen? Zijn er binnen jullie afdeling veel mensen met een statistische achtergrond? Dan zou ik (iets) meer neigen naar R. Of is er behoefte om modellen snel in productie te brengen en alles vanuit IT-perspectief te organiseren? Dan zou ik (iets) meer neigen naar Python. Dit verschil is klein en valt in de praktijk vaak in het niet bij andere overwegingen, zoals de reeds aanwezige kennis van een van beide talen.

Vraag #2
“Zijn die foutkansen in R of Python dan minder?”

De foutkans hangt veel meer af van de programmeur dan van de taal. Sommige mensen hebben wel een voorkeur voor de syntax van Python of R omdat ze dat prettiger vinden lezen, maar dat is meer een kwestie van smaak en vaak ook een kwestie van wennen.

In Python word je gedwongen om je aan een bepaalde vorm van syntax opmaak te houden, deze opmaak is namelijk onderdeel van de syntax. Samen met de Python standaard (PEP-8) zorgt dit ervoor dat we Python code er soms wat uniformer uit vinden zien dan R, maar ook in R kun je prima goed gestructureerde, overdraagbare en onderhoudbare code schrijven. Uiteindelijk valt of staat alles met de programmeur en de inrichting van het code review proces.

Vraag #3
“Waarom de info niet direct uit het bronsysteem halen of een daaraan gekoppelde dwh?”
Dat zou zelfs mijn voorkeur hebben maar was in deze specifieke case nog niet mogelijk omdat de opdrachtgever een applicatie rondom de database had zitten die niet rechtstreeks met SQL queries toegankelijk was. Vaak is dat wel mogelijk en dan kun je vanuit Python/R rechtstreeks SQL queries draaien op het dwh bijv. via een ODBC driver. Ik weet niet welke database jullie gebruiken, maar waar Python/R in ieder geval prima mee overweg kan zijn databases als Oracle, MySQL, PostgreSQL en SQLServer.

Vraag #4
“Mijn ervaring met Python is dat je met deze taal "alles kan" door het ontbreken van datatypes, en dat deze taal eenvoudig te leren is. Mijn ervaring met R is dat deze taal de gebruiker soms "ongevraagd aanvult". Beide ervaringen impliceren soms schijnzekerheid. Hoe ziet u dat?"

Afhankelijk van de toepassing, ben ik het daar tot op zekere hoogte mee eens. R en Python zijn allebei scripting talen die "duck typing" toestaan (i.t.t. de meeste gecompileerde talen zoals C/++/#): je hoeft van te voren niet hard te definiëren welk type (int, double, etc) je variabele heeft. Dat maakt e.e.a. zeer flexibel omdat je tussentijds eenvoudig kunt switchen van het ene naar het andere object/datatype. De keerzijde is dat hierdoor ongewenste fouten kunnen ontstaan omdat de interpreter niet begint te "piepen" wanneer je tussentijds het datatype wijzigt. In algemene zin raden wij daarom aan om altijd gebruik te maken van unit testen. Op die manier kun je elk moment aantonen dat je programma doet was je verwacht dat het zou moeten doen.

Vraag #5
"Maar in welk format staan die tabellen dan, staan die in Excel?”

De tabellen kunnen in elk gewenst format worden ingelezen, zowel Python als R zijn in dat opzicht bijzonder flexibel. De tabellen kunnen staan in Excel, meerdere CSV-files of in een DWH. Ook is het mogelijk om data te “scrapen” van openbare bronnen zoals websites.

Vraag #6
“Dus waar haalt Python de brondata vandaan?”

In de usecase die we tijdens het webinar bespraken (het automatiseren van een reken- en rapportagestraat van ca. 30 Excel tabbladen) haalden we de brondata uit het datawarehouse. Dat gaat nu nog via een tussenstap in Excel, maar de je kunt vanuit Python/R ook heel makkelijk rechtstreeks SQL queries draaien op het datawarehouse. Hoewel deze aanpak hier om technische reden (nog) niet mogelijk was, zou daar wel de voorkeur naar uitgaan. Database omgevingen zien er namelijk veel "strenger" op toe dat de data strak georganiseerd blijven, waardoor de kans op invoerfouten (nog) kleiner is.

Vraag #7
“Programmeren jullie in R met Tidyverse?”

Ja, Tidyverse is een krachtig package dat ook tijdens het webinair werd gebruikt om de verschillende datatransformaties (Filter --> Merge --> Bereken --> Groepeer --> Sommeer --> Sorteer --> Plot) in één logische keten (ook wel “chain” genoemd) achter elkaar te zetten.

Vraag #8
“Of kan die data in iedere willekeurige database staan?”

Ja, de meeste databases zijn vanuit Python/R te benaderen via een zogenaamde ODBC driver. Ik weet niet welke database jullie gebruiken, maar waar Python/R in ieder geval prima mee overweg kan zijn databases als Oracle, MySQL, PostgreSQL en SQLServer.

Vraag #9
"Ik bedoelde eigenlijk: waarom in R of Python programmeren als het ook direct in het bronsysteem is vast te leggen of in een DWH? Dan hoef ik als gebruiker niet meer Excel, Python of R te gebruiken."

Helemaal mee eens: in algemene zin heeft het de voorkeur om de gegevens via een query rechtstreeks uit het DWH te halen. Dat kan natuurlijk gewoon in SQL en in die omgeving kun je ook prima allerlei rekenbewerkingen uitvoeren en (eenvoudige) rapportages maken. Wat Python/R daarboven toevoegen, is dat je ook toegang hebt tot geavanceerdere rekenmethoden (zoals statistische en ML-modellen) en automatisch een rapport kunt genereren in elk gewenst format (niet alleen PDF of PowerPoint, maar ook als online dashboard, etc.).

Vraag #10
“Waarom geen macro in Excel maken? Wanneer kies je voor R of alternatief en niet meer voor macro in VBA?”

Je kiest voor R als alternatief voor VBA wanneer je data wil inlezen uit andere bronnen dan alleen maar Excel, wanneer je processen gestructureerd wilt automatiseren en wanneer je het reken- of rapportageproces ook in de toekomst onderhoudbaar en uitbreidbaar wil houden. Tevens heb je direct toegang tot het volledige arsenaal van analysetechnieken die Python en R bieden.

Vraag #11
“**Zowel Python als R zijn ook EUC-toepassingen. Hoe beheers je dat dan?”**

Python en R kunnen inderdaad worden ingezet als “End User Computing” oplossingen, maar dat hoeft niet. Afhankelijk van het type gebruiker kan de code zelfs geheel verborgen blijven achter een grafische user interface of website. De gebruiker is zich er dan niet eens van bewust dat het werk achter de schermen wordt gedaan door Python of R.

Vraag #12
"Excuses, wat is de naam van de package in R?"

Tidyverse (vraag 7).

Vraag #13
“Kun je met Power BI hetzelfde bereiken als met Python of R?”

Nee, Power BI is primair een visualisatietool waarmee je met “drag-en-drop” mooie interactieve dashboards kunt maken. Er zit ook wel een macrotaal achter waarmee je kunt programmeren maar die is specifiek gericht op visualisaties. Zowel Python als R zijn “general purpose” talen

waarmee je ook visualisaties kunt maken maar daarnaast veel meer mogelijkheden bieden, zoals het automatiseren van medium scale reken- en rapportageprocessen.

Vraag #14
“Waar blijft de 8 dagen aan werk per maand in zitten in het voorbeeld?”

We hebben nu een deel van het rapportageproces geautomatiseerd (in dit geval met R). Het andere deel wordt nu op vergelijkbare wijze geautomatiseerd en levert naar verwachting nog eens 3 dagen extra verkorting van de doorlooptijd op. De resterende 5 dagen zullen naar verwachting nodig blijven om de invoergegevens af te stemmen met de business en de resultaten te communiceren naar alle stakeholders.

Vraag #15
“In de oude situatie werkten de medewerkers met Excel. Konden deze medewerkers zomaar aan de slag met R?“

Ja, mensen met Excel ervaring snappen over het algemeen goed hoe je functies maakt en die kennis is goed om te zetten naar R, dat in essentie ook een functie georiënteerde taal is. Het aanleren van de eerste programmeertaal vergt wel een "leercurve", die afhankelijk van de situatie enkele dagen, weken of maanden in beslag neemt. De ervaring leert wel, dat het leren van programmeertalen een zekere overeenkomst heeft met het aanleren van natuurlijke talen: wanneer je eenmaal 1 taal beheerst, gaat het leren van een tweede een stuk sneller.