

Vereniging voor Statistiek
Contactgroep Statistische Programmatuur
Subgroep "Opzet van statistische pakketten
in het algemeen"

September 1978

Defaults, een zegen of een ramp?

Redactie : Floor van Nes

Eerste verslag van een werkgroep, bestaande uit:

Nol Bendermacher	(Kath. Universiteit, Nijmegen)
Arthur Cappel	(Hoogovens, IJmuiden)
Floor van Nes	(Centraal Planbureau, Den Haag)
Douwe van der Sluis	(Rijks Universiteit, Groningen)
Gerrit van der Vorst	(Rijks Universiteit, Utrecht)

1. Voorwoord.

Bij discussies binnen de Contactgroep Statistische Programmatuur, met name in de subgroep "Opzet van Statistische Pakketten", wordt herhaaldelijk blijk gegeven van ontevredenheid over het gebruik van defaults in statistische programmatuur. Enerzijds is er kritiek op de wijze waarop sommige groepen gebruikers met defaults omspringen, anderzijds op de keuze van defaultwaarden door de ontwerpers van (standaard) software.

Naar aanleiding hiervan is op 17 augustus 1977 in een vergadering van genoemde subgroep een werkgroep ingesteld bestaande uit Cappel, Van Nes, Van der Sluis en Van der Vorst, later aangevuld met Bendermacher. Het doel van deze werkgroep is om het defaultmechanisme nader te bestuderen en zo mogelijk richtlijnen te geven voor het gebruik ervan, in het bijzonder door ontwerpers van standaardprogrammatuur.

In maart 1978 heeft de werkgroep een voorlopig schriftelijk verslag uitgebracht aan de subgroep. Aangezien er weinig kritiek is ontvangen kan dit eerste verslag nu vrijwel ongewijzigd aan de jaarvergadering van de contactgroep worden aangeboden.

Hoewel het (nog) niet is gelukt om eenduidige richtlijnen op te stellen voor gebruikers en ontwerpers van statistische programmatuur, hopen wij dat de gegeven inventarisatie als basis kan dienen voor verdere discussie over gebruik en misbruik van defaultwaarden.

Literatuur over dit onderwerp hebben wij nagenoeg niet kunnen vinden. Vandaar de weinig gebruikelijke vorm van de literatuurlijst achterin. Lezers die wel op de hoogte zijn van literatuur over het defaultmechanisme worden uitgenodigd ons daarvan in kennis te stellen.

2. Wat zijn defaults?

Zoals bekend kan bij een computerprogramma door de gebruiker meestal stuurinformatie worden opgegeven door het toekennen van

waarden aan parameters. Zo'n waarde kan behalve numeriek ook symbolisch (alfanumeriek) of logisch (boolean) zijn. Bij veel programma's kiest de computer "zelf" een waarde voor de parameters waaraan de gebruiker geen waarde heeft toegekend. Dat gebeurt via het zogenaamde defaultmechanisme. De letterlijke betekenis van het woord default is : verzuim, in gebreke blijven. Bij computerverwerking verstaat men onder default de waarde die de computer voor een parameter invult wanneer deze niet door de gebruiker is opgegeven.

In de praktijk komt het nogal eens voor dat ook fouten in invoergegevens worden opgevangen via het defaultmechanisme. Naar onze mening ten onrechte, want als een gebruiker zelf een waarde heeft opgegeven was hij kennelijk niet van plan de defaultwaarde te kiezen. Een fout dient via een foutafhandelingsmechanisme te worden gesignaleerd aan de gebruiker, waarna in een beperkt aantal gevallen wellicht een herstelpoging door het programma kan volgen. Er moet daarom een scherpe scheiding worden aangebracht tussen toekenning van defaultwaarden enerzijds en afhandeling van fouten anderzijds.

De wijze waarop het defaultmechanisme werkt hangt ook samen met de manier waarop de parameters voor een bewerking aan het programma worden aangeboden. We onderscheiden twee methoden:

1. De parameterwaarden moeten in vaste volgorde worden aangeboden. Als een programma gebruik maakt van een vast invoerformaat betekent dit tevens dat elke parameter een vaste plaats heeft op de invoerkaart(en). Wanneer de gebruiker een defaultwaarde wil kiezen kan hij de plaats van de betreffende parameter blanco laten. Als het programma geen vast invoerformaat kent moet de gebruiker om een defaultwaarde te verkrijgen duidelijk aangeven dat hij de betreffende plaats overslaat, bijvoorbeeld door twee scheidingstekens achter elkaar te plaatsen.
2. Elke parameter heeft een label, zodat de gelabelde parameterwaarden in willekeurige volgorde kunnen worden aangeboden. De gebruiker behoeft nu alleen parameterwaarden die afwijken van de default aan te geven op de invoerkaarten.

De wijze waarop een en ander zou kunnen worden verwezenlijkt door een programma-ontwerper kan met behulp van een pseudo-programmeertaal worden geïllustreerd aan de hand van het volgende voorbeeld :

Bij een analyseprocedure, die zelf met een keyword wordt aangegeven, moeten n parameters een waarde krijgen. In de procedure zijn voor de parameters n defaultwaarden $d(i)$, $i=1, \dots, n$ gegeven. Met een niet nader te omschrijven procedure LEES (p,n) worden de parameterwaarden $p(i)$, $i=1, \dots, n$ van kaart gelezen. De controle vindt plaats in een, eveneens niet nader te omschrijven, Boolean procedure ILEGAL. ILEGAL (i,p(i)) is waar, als de toegekende waarde $p(i)$ niet is toegestaan voor parameter i. Programma 1 zou in dit geval het invoerprogramma kunnen zijn. (Zie pag. 3a).

Voor het invoeren van gelabelde parameterwaarden wordt in het voorbeeld de procedure LEES gewijzigd, zodat deze steeds de volgende combinatie label-parameterwaarde leest. Er zijn n labels $k(i)$, $i=1, \dots, n$ gegeven, zodanig dat $k(i)$ de label is die de i^e parameter aanduidt. Programma 2 zou het invoerprogramma bij willekeurige parameter volgorde kunnen zijn.

Het effect voor de gebruiker kan worden geïllustreerd aan de hand van een concreter voorbeeld. Zij POLREG een procedure voor het maken van gewogen polynoom regressies. Neem aan dat er vijf invoerparameters zijn, waarvoor in volgorde geldt:

	betekenis	label	default
1 ^e	de onafhankelijk veranderlijke	Y	geen
2 ^e	de afhankelijk veranderlijke	X	de tijdas
3 ^e	een vector met gewichten	W	vector met enen
4 ^e	minimale R^2	R	0.9
5 ^e	maximale graad van het polynoom	D	5

Een gebruiker die voor zijn variabele VARY een ongewogen regressie wil op een veelterm in de tijd van maximale graad 5 en daarbij genoeg neemt met een lagere graads polynoom wanneer de R^2

TITLE programma 1 vaste parameter volgorde	PROBLEM NUMBER
---	----------------

LEES(p, n)	for i := 1 step 1 until n	do if p(i) = blanko then p(i) := d(i) fi	if ILEGAL(i, p(i)) then error handling fi	od
------------	---------------------------	--	---	----

TITLE programma 2 gelabelde parameter waarden	PROBLEM NUMBER
--	----------------

comment zet eerst alle parameter waarden op default for i := 1 step 1 until n do p(i) := d(i) od comment wijzig par's waarvoor waarde wordt ingelezen while (. 7 end parameter list) do LEES(key, pval) for i := 1 step 1 until n do if key = k(i) then p(i) := pval fi go to next od error handling (illegal key) next: od comment controleer op fouten for i := 1 step 1 until n do if ILEGAL(i, p(i)) then error handling fi od				
---	--	--	--	--

groter is dan 0.9, moet nu bij vaste parametervolgorde als invoerkaart leveren:

```
POLREG, VARY,,,,
```

Bij gelabelde parameterinvoer wordt dit:

```
POLREG, Y = VARY
```

Een ongewogen regressie van VAR2 op een polynoom van maximale graad 4 in VAR1 en met een minimale R^2 van .81 kan worden verkregen met de kaart:

```
POLREG, VAR2, VAR1,, .81,4
```

bij vaste parametervolgorde. Bij gelabelde parameters zou dit kunnen met de kaart:

```
POLREG, D = 4, X = VAR1, Y = VAR2, R = .81
```

Het zal duidelijk zijn dat voor lange parameterlijsten waarbij veel defaultwaarden worden gebruikt de gelabelde parametermethode het aantrekkelijkst is. Zijn er slechts weinig parameters en wordt er niet zoveel gebruik gemaakt van defaults dan is de vaste parametervolgorde wel zo eenvoudig.

3. Bewust kiezen voor defaults.

3.1 Door de ontwerper.

Veel van de huidige standaardpakketten zijn door de ontwerper(s) niet als zodanig ontwikkeld. Aanvankelijk kende de ontwerper zijn gebruikers en het ontwerp was op hen afgestemd. Dat er ook ander gebruik van een procedure mogelijk is zal hij zich vaak niet eens gerealiseerd hebben. Wanneer het programma (bijv. door mond-tot-mond reclame) in breder kring gebruikt gaat worden, ontstaat wel behoefte aan deze varianten. Als er dan extra stuurgegevens nodig zijn wordt al te gemakkelijk naar het defaultmechanisme gegrepen om de gebruikers van het eerste uur geen overlast te bezorgen. In dit verband zou als oplossing

gedacht kunnen worden aan per gebruikersgroep instelbare defaultwaarden. Er zijn dan echter wel problemen te verwachten met documentatie, uitwisselbaarheid en versies die "een eigen leven" gaan leiden.

3.2 Door de gebruiker.

Ons inziens dient een gebruiker bewust te kiezen voor de waarde die hij gebruikt, ook als dit de defaultwaarde is. Gebruik van het defaultmechanisme zonder te weten waarvoor men kiest, zonder de consequenties van de keuze te beseffen of gewoon uit vergetachtigheid is afkeurenswaardig. Helaas kan een programma dit soort misbruik niet detecteren, hetgeen een van de nadelen van het defaultmechanisme is.

4. Defaults voor wie, waar en hoe?

Het is onmogelijk algemene uitspraken over het defaultmechanisme te doen. Dit moet van geval tot geval worden bezien, waarbij de navolgende aspecten in de overwegingen worden betrokken :

- de gebruikersomgeving waarin de betreffende software functioneert
- de soort parameters waaraan de defaultwaarde kan worden toegekend
- de wijze waarop de defaultwaarde uit de verzameling toegestane waarden wordt gekozen
- de wijze waarop het gebruik van de betreffende parameters is gedocumenteerd.

Elk van deze aspecten verdient eerst nader te worden gepreciseerd.

4.1 Gebruikersomgeving.

Naar het soort werk kan men drie essentieel verschillende gebruikersomgevingen onderscheiden :

- productie Hierbij is de analysemethode van te voren min of meer vastgelegd. In het algemeen gaat het om de analyse van nieuwe gegevens met een bestaand programma.

- onderzoek Hierbij is de analysemethode zelf (mede) onderwerp van studie. Hoewel dit principieel onjuist is zullen veel onderzoekers zich uit praktische overwegingen beperken tot die analysemethoden die in de vorm van standaardprogrammatuur beschikbaar zijn.

- onderwijs Hierbij is het doel de gebruiker vertrouwd te maken met de eigenschappen van bepaalde gegevens, analysemethoden en standaardprogrammatuur.

Ook de wijze waarop de gebruiker zijn werk aanbiedt is van belang :

- via batch Hierbij wordt een stapel kaarten ingeleverd die men al dan niet met een stapel output terugkrijgt.

- via een interactieve terminal
 Hierbij moet de invoer (althans ten dele) ter plaatse worden ingetoetst. Het opgeven van veel standaardwaarden voor parameters kan tijdrovend zijn. De mogelijkheid van vraag en antwoord bestaat.

Superpositie van beide indelingen levert zes soorten gebruikers.

4.2 Parametersoort.

Voor ons doel kunnen ruwweg drie soorten parameters worden onderscheiden :

- programma-technische bijvoorbeeld t.a.v. I/O (files, standaarduitvoer, etc.), geheugengebruik, representatie missing data
- reken-technische bijvoorbeeld convergentiecriteria, max. aantal iteraties en startwaarden bij iteratieve processen
- model-technische d.w.z. betreffende de analysemethode; bijvoorbeeld uit te voeren transformaties, keuze van de methode en varianten daarvan (bijv. methode = regressie; variant = wel of niet door de oorsprong) en andere opties.

4.3 Keuze defaultwaarde.

Bij de keuze van de waarde die default zal zijn voor een parameter kan men zich, afhankelijk van gebruikersomgeving en parametersoort, door verschillende principes laten leiden.

- modaal kiezen default is de waarde die van alle toegestane waarden het meest voorkomt.
- conservatief kiezen default is een veilige waarde die zelden wordt overschreden, meestal niet destructief is, etc. Dit kan wel eens leiden tot overbodig rekenwerk en onnodige output.
- zuinig kiezen default is een waarde die zo min mogelijk reken-tijd en geheugen vraagt, weinig output geeft, etc.
- logisch kiezen in sommige gevallen kan het mogelijk zijn als default een waarde te kiezen die logisch samenhangt met een aantal voorgaande keuzen.

4.4 Documentatie.

4.4.1 Programmabeschrijving.

Een goede programmabeschrijving is uiteraard ten alle tijde noodzakelijk. Dit aspect is toch opgenomen omdat een slechte documentatie het onjuist gebruik van defaults in de hand werkt. Een gebruiker kan er licht toe overgaan de defaultwaarde te "kiezen" omdat de betekenis van de parameter niet duidelijk is of omdat hij zelfs van het bestaan ervan niet afweet.

In het algemeen mag men van goede documentatie verwachten dat betekenis en gebruikswijze van alle parameters er duidelijk in worden behandeld. Wanneer voor een parameter een defaultwaarde bestaat dient deze uiteraard in de documentatie te worden vermeld, terwijl het in veel gevallen zinvol is als tevens de keuze van deze waarde enigermate wordt toegelicht. Vooral voor de beginnende gebruiker, die aan de hand van de documentatie parameterwaarden gaat kiezen en daarbij ongetwijfeld een voorkeur heeft voor zo veel mogelijke defaultwaarden, is het van belang dat duidelijk is aangegeven wanneer wel en wanneer niet met de default kan worden volstaan.

4.4.2 Programma-uitvoer.

Voor de documentatie van de gebruiker is het van belang dat steeds de gebruikte parameterwaarde, ook als dit de defaultwaarde is, in de uitvoer wordt geëchood. Uiteraard moet de uitvoer overigens ook voldoende informatie bevatten over het verloop van het analyseproces om de resultaten te kunnen evalueren. Verkeerde keuzen van parameterwaarden, dus ook onjuist gebruik van defaultwaarden, dienen daarbij zo veel mogelijk aan het licht te komen.

5. Zegen of ramp?

Zoals uit het voorgaande blijkt kan deze vraag niet zonder meer worden beantwoord. Of het gebruik van defaultwaarden wenselijk is en hoe deze moeten worden gekozen kan, afhankelijk van soort gebruiker en soort parameter, in het volgende tableau worden aangegeven :

parameter gebruiker		programma- technisch	reken- technisch	model- technisch
Produktie	Batch			
	Int.			
Onderzoek	Batch			
	Int.			
Onderwijs	Batch			
	Int.			

In elk vakje moeten twee gegevens worden geplaatst :

- een teken dat de wenselijkheid van het gebruik van defaults aangeeft : + = zo veel mogelijk
 = met mate
 - = zo weinig mogelijk
- een woord dat het doorslaggevend keuzecriterium aangeeft (modaal, conservatief, zuinig). Uiteraard zal men zo veel mogelijk trachten aan alle criteria te voldoen. Het is de bedoeling aan te geven wat de doorslag moet geven als er

conflicterende keuzemogelijkheden zijn. Het logisch keuze criterium is niet opgenomen omdat dit lang niet altijd mogelijk is, maar als het mogelijk is op grond van voorgaande gegevens een consistente defaultwaarde te kiezen verdient dit de voorkeur. De lezer wordt uitgenodigd dit tableau zelf in te vullen. Hierbij ware het volgende te overwegen :

- t.a.v. batch/interactief

Bij interactief gebruik is de behoefte aan defaultwaarden groter dan bij batch. Door interactie kunnen ten onrechte gekozen defaults worden hersteld. Interactief is de neiging tot zuinige keuze van de waarde groot (responsietijd !)

- t.a.v. produktie/onderzoek/onderwijs

De produktieomgeving leent zich het meest voor het gebruik van zuinige defaults. De onderzoek omgeving is minder geschikt voor het gebruik van defaults wanneer de methode zelf nog in studie is. Bij onderwijs zullen liefst de parameters die niet nodig zijn voor het begrip een (conservatieve) defaultwaarde hebben, maar de andere juist niet.

- t.a.v. parametersoort

De programma- en rekentechnische parameters zijn meer geschikt voor defaultwaarden dan de analysemethode. Waar defaults wenselijk zijn zal een programmatische parameter het meest in aanmerking komen voor een zuinige keuze. Bij de rekentechniek en de analysemethode is dit sterk afhankelijk van de omgeving.

6. Wat nu ?

Het antwoord op de vraag "zegen of ramp ?" kan wat ons betreft bevestigend luiden. Want hoewel het zeer instructief is te proberen het tableau uit de vorige paragraaf in te vullen, zijn we er niet in geslaagd dit op bevredigende wijze te doen.

Het grootste probleem bleek te zijn, dat de indeling naar parametersoort nog te grof is om eenduidige antwoorden mogelijk te maken. Voor een specifieke parameter is het antwoord meestal wel te geven.

Toch lijkt het zinvol dat ontwerpers van standaard software voor zichzelf een soortgelijk tableau maken. Het aantal gebruikersgroepen zal dan meestal kleiner zijn en het aantal parametersoorten kan sterk worden uitgebreid, toegesneden op de betreffende programmatuur. Wanneer in teamverband wordt gewerkt zal een dergelijk schema te meer nodig zijn.

Te zijner tijd hopen wij nader in te gaan op de wijze waarop het defaultmechanisme functioneert in enige ons bekende statistische pakketten, zoals BMDP, P-STAT, SPSS, WESP etc.

7. Literatuur.

Er moet voor de stuurinformatie zoveel mogelijk van defaults gebruik worden gemaakt.

KUNST : Gids voor programmeurs,
intern rapport, Kath. Universiteit,
Nijmegen (1977)

... Allow defaults; echo in output...

Kernighan & Plaughter, The elements of
programming style, New York, 1974.

...Lots of statisticians have been complaining of the abuse of statistical methods and statistical software. If you just know the computer technique it is possible, with little effort, to try a lot of methods hoping to get some results that will suit. And if you know too little to set parameter values the programs usually provide defaults, so there will always be some result. My hope is, that as interactive programs reduce the need of knowledge about computers, more efforts could be made on selecting a proper statistical method. So, by omitting defaults in INSTAT, I take for granted that the user knows what he is doing and therefore will have no problems to supply parameters...

Gullbrith Palm, ontwerper van het inter-
aktieve statistische pakket INSTAT, per-
soonlijke brief (1977)

...That is, the default case does not help us much if it is used only a small fraction of the time. As there are more and more possible cases, the burden of remembering which is the default can become greater than simply writing the complete specification. If the cases occur with more or less equal frequency, the default system then loses its value because it cannot be used often enough to make it worthwhile. On the other hand, if the defaults are used often, we may not remember the non-default cases when we have to use them. Although we have not studied this situation empirically, we might conjecture that defaults are most useful when they are used fairly frequently, but not too frequently...

Weinberg, The psychology of computer programming, New York, 1971

...In our opinion quite a number of defaults should be reconsidered. In many cases other values should be chosen and in some cases defaults should be abolished altogether...

VanHoboken & Niemöller, Package development, paper gepresenteerd op ECPR, Leuven, April 8-14, 1976.