Op 9 september 1990 overleed Prof. Dr. Albert Verbeek. Zijn betekenis voor de statistiek, voor de VVS, en voor KM, is verwoord door Prof. Dr. Tom Snijders in een "In Memoriam" in het VVS-Bulletin van november 1990. Daarin spreekt Tom ook over de "gedegenheid en bruikbaarheid" van Alberts artikelen. Men kan zich van deze kwalificaties overtuigen door lezing van het volgende artikel. Het werd aangetroffen in Alberts nalatenschap, en kon dus niet aan een revisie worden onderworpen. De redactie meent dat een revisie in het geheel niet nodig zou zijn geweest, en plaatst het artikel als laatste eerbewijs aan Albert Verbeek.

Redactie KM

KM 36(1991)
pag 101-113

Van verzamelde data tot data-analyse ofwel fouten maken we allemaal

Albert Verbeek *
ICS, RUU †

Samenvatting

Deze notitie bevat een aantal praktische opmerkingen voor het controleren en manipuleren van data ter voorkoming van grove, niet herkende fouten bij data-analyse. De opmerkingen zijn toegespitst op het gebruik van BMDP, en SPSS-PC (of SPSS-X), terwijl SPSS-9 soms als vergelijking wordt gebruikt.

1 Veel voorkomende fouten

Verzamelde data moeten gecontroleerd en gecorrigeerd ('gewassen', 'gecleand') worden, waarna meestal een aantal nieuwe variabelen aangemaakt moet worden. Deze fase biedt talloze mogelijkheden tot het maken van grove fouten, die wel tot onzin leiden, maar vaak op zo'n manier dat dit niet herkend wordt. (Een interpretatie verzinnen lukt altijd wel.) Deze notitie gaat over de kwaliteitsbeheersing van dit complexe proces, en is vooral geschreven voor beginnende, sociaalwetenschappelijke onderzoekers. De meest effectieve valkuilen op de weg van ruwe data tot data-analyse zijn:

a. Codeer- en typefouten. Deze zijn onvermijdelijk, maar het is wel van belang de omvang enigszins te kennen en zo veel mogelijk te reduceren.
b. Fouten van de respondent; vooral: verkeerd begrepen vragen, suggestieve vragen, vragen met een sociaal wenselijk antwoord. Ook deze zijn ten dele onvermijdelijk, maar soms is het wel mogelijk een schatting van de nauwkeurigheid te krijgen, bijvoorbeeld door te werken met verschillende vraagformuleringen.
c. Het omcoderen van codes zoals 0-en-1 zonder dat dit in de beschrijving (of `VALUE LABELS`) zorgvuldig vastgelegd wordt. Vaak denkt men: 'Ik onthoud het wel', maar vergeet het later toch.
d. Logische expressies met `NOT`, `AND` en `OR`. Dat een analyse op de verkeerde cases is uitgevoerd, wordt vaak niet ontdekt. Twee voorbeelden uit de praktijk. Wat betekent

```
A NE 3 OR 4?
```

Uitgeschreven wordt het `(A NE 3) OR (A NE 4)` en is dus *altijd* waar! Bedoeld was

*Met dank aan Jos Dessens, Cora Maas en Ronald Batenburg voor nuttige discussies en commentaar.
†adres: ICS, Fac.Soc.Wet., Heidelberglaan 1, 3584CS Utrecht

```
NOT (A EQ 3 OR A EQ 4)
```

Vermijd de dubbelzinnigheid in ...AND...OR.... Immers,

```
(A EQ 3 AND B EQ 4) OR C EQ 5
```

is iets anders dan

```
A EQ 3 AND (B EQ 4 OR C EQ 5)
```

Tips: Kort niet teveel af. Wees royaal met haakjes. Kies een formulering die zo duidelijk mogelijk is, en documenteer uitdrukkingen die niet direct duidelijk zijn. Test logische expressies met geschikt gekozen kruistabellen.

e. Ontbrekende waarden in logische expressies. Men moet zich goed realiseren wat `IF (...)`, `SELECT IF (...)`, en `WEIGHT BY ...` doen als '...' een ontbrekende waarde heeft. SPSS-9 deed dit bijvoorbeeld anders (en veel slechter) dan SPSS-X en BMDP. Bij de laatste twee geldt voor logische expressies een driewaardige logica:

A AND B:

		B		
		waar	onwaar	ontbr.
A	waar	waar	onwaar	ontbr.
	onwaar	onwaar	onwaar	onwaar
	ontbr	ontbr.	onwaar	ontbr.

A OR B:

		B		
		waar	onwaar	ontbr.
A	waar	waar	waar	waar
	onwaar	waar	onwaar	ontbr.
	ontbr	waar	ontbr.	ontbr.

NOT A:

A	NOT A
waar	onwaar
onwaar	waar
ontbr.	ontbr.

En zo hoort het ook. Maar bijvoorbeeld SPSS 9, SPSS-PC en STATA denken daar anders over. In logische expressies herkennen ze geen ontbrekende waarden! Dat is een valkuil voor gebruikers.

f. Overige fouten met transformaties van variabelen met ontbrekende waarden. Zoals:

```
COMPUTE SUM = X + Y + Z
```

Er is een belangrijk verschil tussen programmatuur met een vaste interne representatie voor ontbrekende waarden (`$SYSMIS` in SPSS-PC, `XMIS` in BMDP) en programmatuur waarin zo'n representatie ontbreekt (zoals SPSS-9). Als zo'n representatie ontbreekt, kan het programma niet 'op eigen houtje' een ontbrekende waarde aan `SUM` toekennen, telkens wanneer dat nodig is. De simpele reden is dat het programma de representatie van 'ontbrekende waarde' niet kent. Bij SPSS-9 werd dit opgelost met `ASSIGN MISSING`. Behalve als dat door de gebruiker vergeten werd ...

Bij programma's met een 'interne ontbrekende waarde' wordt over het algemeen bij transformaties aan het linkerlid de waarde 'ontbrekend' toegekend, zodra minstens één van de variabelen die rechts voorkomt de waarde 'ontbrekend' heeft, en ook als rechts een functie een argument heeft waarvoor hij niet gedefinieerd is (zoals $1/x$ met $x = 0$, of $\log(x)$ met $x \leq 0$).

In de informatica wordt een ontbrekende waarde wel een NaN genoemd, een afkorting van 'not a number'. Zo is het resultaat van $1/0$, $\sqrt{-1}$ of (10**10000) altijd een NaN. De

wortel van een negatief getal levert dus meestal een ontbrekende waarde op (soms, zoals in GLIM, een nul!). Bedenk dat ook een vrij simpele berekening door afrondfouten wel eens kan leiden tot een 'onterecht' ontbrekende waarde.

g. Fouten met het samenvoegen (= mergen = matchen) van twee files. Controleren of er correct samengevoegd is, wordt vaak geheel achterwege gelaten, omdat het inderdaad niet makkelijk is. Hier twee hints. Zie ook sectie 2.
   - Vertrouw er niet te gauw op dat de i-de case in file 1 hetzelfde is als de i-de case in file 2. Neem zo mogelijk in beide files een sleutel op (zoals respondent nummer). Als dat niet kan, neem dan een bijna-sleutel (een continue variabele zoals inkomen) mee en test achteraf of beide variabelen gelijk zijn.
   - Bekijk altijd 10 à 20 cases om te zien of het goed gegaan is. En graag vooral ook de laatste 5 cases. Als het mis gaat, is dat vaak achteraan te zien. Kloppen aantallen cases en variabelen na het samenvoegen?

   Sommige programma's eisen bij het gebruik van een sleutel, dat beide files op die sleutel gesorteerd zijn. Sommige slechte programma's geven geen melding als hieraan niet voldaan blijkt, en produceren dan gewoon rotzooi.
   In STATA kun je heel mooi en veilig matchen. Een mooie truc is, dat STATA automatisch een variabele toevoegt met de waarde 1 voor cases met waarden uit file 1 en allemaal ontbrekende waarden uit file 2; waarde 2 voor cases met waarden uit file 2 en allemaal ontbrekende waarden uit file 1; waarde 3 voor echt gematchde cases. Het is een goed idee zo'n variabele zelf te maken in andere pakketten, en achteraf de frequentieverdeling van deze variabele even te bekijken.

h. Fouten met wegen, met name als men niet aan de output kan zien of de resultaten überhaupt gewogen zijn of niet. Begin een job voor zo'n programma *altijd* met het laten printen van min en max, gemiddelde en standaardafwijking van de gewichten! In SPSS:

   ```
   CONDESCRIPTIVE wvar
   ```

   waarbij wvar de variabele is die de gewichten bevat. In SPSS-9 was dat altijd `CASWGT`, in SPSS-PC is het geen systeem-variabele meer; nu toont zowel `GET FILE` als `SHOW WEIGHT` wvar. Zie verder sectie 5.

2 Namen en labels

Kies verstandige namen voor variabelen: zinvolle, makkelijk te typen afkortingen (`LFT` ipv `LEEFTIJD`; `IJ` of `Y` geeft veel verwarring). Gebruik overvloedig labels (lange namen) als toelichting, voor variabelen (`VAR LABELS`), voor waarden (`VALUE LABELS`), en voor verschillende versies van een bestand of van een job. Dat spaart later bergen ergernis en fouten. Tussen twee haakjes, `VAR096B` behoort niet tot de verstandige afkortingen. Het is niet eens kort; kort dan in hemelsnaam tenminste af tot `V96B`.

Als er geen case-identificatie (respondentnummer of zo) in de data aanwezig is, maak er dan één. Desnoods via zoiets als `COMPUTE CASEID = casenummer`. Als je later de data gaat sorteren, en misschien bepaalde variabelen wegschrijft die je er later weer met een merge-opdracht aan toevoegt, dan kun je via `CASEID` mergen, of achteraf via `CASEID` testen of het goed gegaan is.

Zoveel mogelijk één ontbrekende waarde voor alle variabelen, en liefst één die duidelijk afwijkt en opvalt, ook als je er per ongeluk verder mee rekent (bijv. NaN, -999, of -9E99).

Duidelijkheid en zelfdocumentatie zijn uiterst belangrijke wapenen in de strijd voor

kwaliteit!

3 Data-controle: de eerste fase

Tips.

a. Laat (een deel van) het coderen en typen dubbel doen. Alles onafhankelijk dubbel doen verbetert de kwaliteit aanzienlijk. Alles of een deel (zeg 100 enquêtes) dubbel doen, maakt een schatting mogelijk van de percentages toevallige fouten (maar niet van systematische fouten zoals een verkeerd begrepen instructie). Een belangrijk hulpmiddel bij de kwaliteitsbewaking is het meten van kwaliteit, om te zien of een verdere inspanning tot kwaliteitsverhoging nodig is. 'Not everything worthwhile doing, is worthwhile to do it well.' Maar je moet wel weten of het goed genoeg gedaan is.

b. Controleer het bereik van elke variabele afzonderlijk. (Vooral min en max per variabele.)

c. Controleer of de routing van de vragenlijst correct gevolgd is. (Een voorbeeld van routing: 'Als vraag 33 = JA, dan moeten vraag 34 t/m 38 ontbreken.')

d. Maak kruistabellen of scattergrammen van variabelen die duidelijk met elkaar samenhangen. Bijvoorbeeld: leeftijd tegen aantal kinderen; wel/geen auto tegen prijs-van-die-auto. Je haalt er gegarandeerd inconsistenties mee uit!

e. Van sommige variabelen is de populatieverdeling bekend, wellicht uit CBS-publicaties. Vergelijk dan de steekproefverdeling met de populatieverdeling (toets: Pearsons X^2 of Kolmogorov-Smirnov, zie bijvoorbeeld Siegel, 1956). Denk na over de oorzaak en over de gevolgen van over- of ondervertegenwoordiging van bepaalde groepen, en overweeg of je hiervoor bijvoorbeeld door wegen ('poststratificatie') enigszins zult corrigeren. Zorg er dan voor dat de gewichten gemiddeld 1 zijn. Zie sectie 5

f. Vergelijk fouten met het oorspronkelijke enquêteformulier om de correcte score te achterhalen. Is dat onmogelijk, hercodeer dan naar ontbrekende waarde.

g. Alfanumerieke waarden zijn soms echt nodig, bijvoorbeeld voor case labels of autonummers. Gebruik ze echter alleen als er een erg goede reden voor is.

h. Last, but not least: Als je een schoon bestand hebt, maak dan een overzicht van alle 1-dimensionale verdelingen, liefst met BMDP2D met `/PRINT ESTIM. NO COUNT.`, desnoods met SPSS en `CONDESCRIPTIVE`. BMDP2D is zoveel informatiever en mooier ... Daar wordt je stil van. Leg deze output in een archief waar je tijdens de analyse goed bij kunt. Je zult ze bij latere data-analyses voortdurend nodig hebben.

4 Ontbrekende waarden

Met ontbrekende waarden wordt hier bedoeld: 'gaten in de datamatrix'. Verwar afwezigheid van ontbrekende waarnemingen niet met veel strengere eisen zoals 'volledig gekruiste opzetten' of 'gebalanceerde data'. Zo is een onvolledige, meerdimensionale tabel met frequenties niet volledig gekruist en niet gebalanceerd, maar bevat toch geen ontbrekende waarnemingen. Er zijn als het ware hele rijen weggelaten uit de datamatrix. Onvolledige waarnemingen betekent: onvolledige rijen.

Het BMDP manual geeft een keurig overzicht welke methode voor ontbrekende waarden in welk programma beschikbaar is. Bedenk dat veel modules in veel pakketten helemaal geen ontbrekende waarden aankunnen. Dit geldt bijvoorbeeld voor de meeste modules voor loglineaire of logistische modellen, en voor alle modules van GLIM. Als je met zo'n module of programma

wilt analyseren moet je dus eerst alle cases met één of meer ontbrekende waarden verwijderen. Invullen van schattingen voor ontbrekende waarden (met BMDPAM) geeft natuurlijk een (iets) te rooskleurig beeld, omdat de residuele storingsterm (het verschil tussen ware waarde en schatting) genegeerd wordt. Maar het kan beter zijn dan 'listwise deletion'. Een recent, briljant idee is het splitsen van een case met ontbrekende waarden in twee cases met het halve gewicht, en het invullen van twee verschillende waarden in de twee halve cases. Het verschil tussen die twee waarden reflecteert onze onzekerheid over de juiste waarde, en corrigeert tot op zekere hoogte voor het hierboven geschetste, te rooskleurige beeld van enkelvoudige imputatie. Zie Rubin (1986) en Little & Rubin (1987).

Onderzoek vooral ook de multivariate structuur van ontbrekende waarden met BMDP8D en BMDPAM, en leg ook die output in het archief. Als je met die programma's nog nooit gewerkt hebt, lees dan tenminste het BMDP-manual eens hierop door, omdat het een mooi en nuttig stukje methodologie is (ontwikkeld door BMDP).

Een rare, grove fout zit in SPSS-X versie 2 (maar niet in SPSS-PC). De interne representatie van de ontbrekende waarde ('SYSMIS') is daar een groot, negatief getal. Als je dan hercodeert zoals in de volgende opdracht

```
RECODE varlist ... (LO THRU -5=-1) ...
```

dan wordt ook de system missing value mee gehercodeerd naar -1! Dat was vrijwel zeker niet de bedoeling, en de kans is erg groot, dat de gebruiker hier nooit achterkomt. Een typisch voorbeeld van een bijzonder gevaarlijke situatie. (Ter herinnering: hier is SPSS-PC dus beter dan SPSS X, maar bij ontbrekende waarden in logische expressies gaat juist SPSS PC in de fout en SPSS X niet.)

Zie ook 1.e.

5 Wegen

De definitie van wegen, en het effect van wegen zijn eenduidig en eenvoudig gedefiniëerd voor onder andere steekproefgrootheden zoals gemiddelden en (co)varianties, en dus ook voor schatters voor de coeffiënten in regressie-analyse, en LISREL en factorscores en factorladingen in factor-analyse, waarvoor de geschatte covariantiematrix en de geschatte gemiddelden voldoende grootheden zijn. Het spreekt echter geenszins vanzelf hoe je bij gewogen data vrijheidsgraden moet tellen, en hoe je schattingsfouten ('standard errors') moet schatten. Allebei kan op minstens twee manieren. Dit leidt tot minstens drie verschillende definities. Elke manier berust op een ander model, dat in bepaalde gevallen toepasbaar is. Maar onderling zijn deze definities strijdig, en leiden ze tot sterk verschillende conclusies. Dus bij elke toepassing is er hoogstens één correct.

In de eerste definitie hangt het aantal vrijheidsgraden niet af van de gewichten, en het aantal onafhankelijke waarnemingen is het aantal regels uit de datamatrix (met gewicht > 0). De schattingsfouten hangen alleen van de *relatieve* gewichten af; dat wil zeggen, en als je alle gewichten met een constante vermenigvuldigt, blijven alle schattingen en schattingsfouten hetzelfde. Deze definitie wordt onder andere gebruikt in STATA en BMDP, behalve in BMDP4F. Bij gewogen kleinste kwadraten methoden en bij poststratifikatie (corrigeren voor ongelijke insluitkansen) is dit de goede methode.

In de tweede definitie zijn de vrijheidsgraden en het aantal onafhankelijke waarnemingen op dezelfde manier gedefinieerd, maar hangen de schattingsfouten wel af van de schaal van de gewichten. Dit is de definitie die GLIM hanteert.

In de derde definitie is het aantal onafhankelijke waarnemingen gelijk aan de som van de gewichten. Deze definitie is onder andere heel handig als je een tabel met frequenties in wilt lezen, zoals bij loglineaire analyse en bij logit-analyse. Gewicht twee betekent hier 'frequentie twee': twee onafhankelijke waarnemingen, die toevallig identieke 'records' opleveren. Deze definitie wordt onder andere gebruikt in SPSS-X, BMDP4F en in STATA met de optie NOSCALE, die het schalen van de gewichten afzet. Hierbij hangen de schattingsfouten dus net als bij definitie 2 direct af van de schaal van de gewichten, in het algemeen zijn ze evenredig met deze schaal tot de macht -1/2. (Niet-geheeltallige gewichten zijn wat moeilijk te interpreteren, maar statistisch is dit bij de meeste modellen geen probleem.)

Veel manuals negeren deze verschillen (bijvoorbeeld SPSS sinds jaar en dag). Welke definitie een pakket gebruikt, kun je makkelijk achterhalen door de uitkomsten van een ongewogen analyse te vergelijken met de uitkomsten van een analyse waarbij alle waarnemingen gewicht 4 hebben. De schattingen zijn in beide gevallen hetzelfde, en bij de eerste definitie zijn ook de schattingsfouten hetzelfde, maar bij definitie 2 en 3 zijn de schattingsfouten bij gewicht 4 half zo groot als bij ongewogen analyse. Dan is alles '2 maal zo significant'. (Een mooie truc om gratis alles significant te maken, of, desgewenst, niet-significant. Het zal geen referee, geen promotor, geen redacteur, kortom geen hond opvallen.) De vrijheidsgraden differentiëren tussen enerzijds definitie 1 en 2, en anderzijds definitie 3.

Als een pakket definitie 2 of 3 gebruikt, terwijl je definitie 1 moet hebben (zoals bij post-stratificatie), dan kun je dat bereiken door er zelf voor te zorgen dat de gewichten gemiddeld 1 zijn; vermenigvuldig met 1/gemiddelde gewicht, dan wordt de som van de gewichten gelijk aan het aantal 'records'. In die zin zijn definitie 2 en 3 krachtiger, althans voor gewaarschuwde, deskundige gebruikers.

Wanneer wegen wenselijk en nodig is, is geen makkelijk te beantwoorden vraag. Indien de steekproef niet enkelvoudig aselect getrokken is, is het zeker wenselijk om bij de databeschrijving ook de wijze van steekproeftrekken zorgvuldig te beschrijven. Bij sommige analyses zal men hierbij rekening willen of moeten houden. Een ruwe vuistregel is, dat men in het algemeen voor een model met afhankelijke en onafhankelijke variabelen, beter niet aselect kan trekken uit de populatie, maar moet zorgen voor een 'optimale' verdeling van de waarnemingen over de onafhankelijke variabelen. Als het model correct is, hoeft men vervolgens niet te wegen. Als het model alleen een erg ruwe benadering is, en men is meer in populatieparameters geïnteresseerd dan in mechanismen (zoals bijvoorbeeld het geval is bij een Centraal Bureau voor de Statistiek), dan doet men er goed aan 'terug te wegen' naar de populatieverdeling. De keuze van een goede steekproefopzet, kan makkelijk meer dan 20% schelen in efficiëntie, dat wil zeggen, in vereiste steekproefomvang. Deze keuze vereist grondige kennis van de statistiek en inhoudelijke kennis van het onderzoeksgebied. Het is belangrijk hiervoor in een vroeg stadium een statisticus te consulteren.

Zie ook 1.h.

6 Secondaire analyse

Als je een schoon bestand van een ander krijgt, maak dan ook weer eerst een overzicht van alle variabelen (zie sectie 3.h), en bekijk hun verdelingen. Welke variabelen zijn continu, welke discreet? Zijn de categorieen goed gedocumenteerd? Zijn er gewichten (`SHOW WEIGHT` in SPSS-PC), hoe varieren die (min, max, gemiddelde, standaard afwijking)? Welke variabelen hebben erg weinig variantie, zijn erg scheef verdeeld ('skewness' > 1) of (veel zeldzamer) hebben erg

dikke staarten ('kurtosis' > 1).

De data-analist moet ten minste ruwweg moet weten hoe de eendimensionale verdelingen eruit zien, omdat dit voor sommige analyses consequenties heeft. Er is echter bij veel manuals, leerboeken, en toepassers onduidelijkheid en onzekerheid over de veronderstellingen zelfs van standaardtechnieken zoals regressie-analyse en LISREL. Velen denken bijvoorbeeld ten onrechte dat hierbij alle variabelen multivariaat normaal verdeeld moeten zijn, terwijl dit in het algemeen alleen van het residu en van eventuele latente variabelen verondersteld wordt (en dus a fortiori ook geldt voor lineaire combinaties hiervan). Bijvoorbeeld bij factoranalyse en lineaire of kwadratische discriminant-analyse wordt wel vrij zwaar geleund op de veronderstelling dat de verdeling multivariaat normaal is (bij discriminant-analyse per groep). De keus van een eventuele transformatie wordt kort besproken aan het eind van de volgende sectie.

Maak ook enkele kruistabellen en scattergrammen, om enig gevoel te krijgen voor de data. Kijken of je de verbanden ziet die je verwacht, is een goede test of de specificaties duidelijk waren. Leg die ook in het archief.

Kijk ook goed hoe ontbrekende waarden gedefinieerd zijn. Liggen ze buiten de 'legale' range? Hercodeer de ontbrekende waarden zonodig naar de interne waarde voor NaN, of althans naar één uniforme waarde zoals -999 of -9E99. Rommelig gedefinieerde ontbrekende waarden leiden al gauw tot fouten en veel tijdverlies.

Als er geen case-identificatie (zoals respondentnummer) in de data aanwezig is, maak er dan één. Zie 1.g.

7 Datacleaning en -transformaties: de volgende fase

Zorg dat je van elke stap tijdens het cleanen van de data en van elke latere transformatie kunt bewijzen hoe je het gedaan hebt. Alle stervelingen maken af en toe fouten. Onderzoekers zijn stervelingen. Modus ponens doet de rest. Zorg dat je tenminste terug kunt vinden wat er precies gedaan is. Bewaar hiertoe alle jobs die de data wijzigen (editten, transformaties, nieuwe variabelen). Voeg ze af en toe samen tot jobs van een redelijke omvang, en bewaar de ruwe data en de tussenresultaten op hard disk, tape of flop. Dus:

$$\text{ruwedata} \xrightarrow{\text{job1}} \text{dataset1} \xrightarrow{\text{job2}} \text{dataset2} \cdots \xrightarrow{\text{jobk}} \text{datasetk}$$

Als je dan in job i een fout ontdekt, kun je dat corrigeren, en de data opnieuw vanaf dataset(i-1) reconstrueren.

Ook als je op de PC werkt, is het raadzaam om zo 'batch-jobs' op te bouwen, eventueel door editten uit logfiles van interactieve sessies, en om de definitieve transformaties met die batchjobs (nogmaals) uit te voeren. (Controleer of ze hetzelfde resultaat opleveren!) Alleen op die manier kun je hard bewijzen wat je gedaan hebt.

Interactief verbeteren of transformeren is gevaarlijk. Maak minstens een log (liefst automatisch, desnoods met de hand). Fouten die je hier maakt zijn bijna niet te achterhalen, maar wel snel gemaakt. Werk uiterst secuur. Interactief werken is eigenlijk alleen geschikt om datatransformaties en andere jobs even uit te testen. Maar daar is het dan ook onmisbaar voor. Een voorbeeld van het gevaar van interactief werken is het volgende. Een onderzoeker voerde in SPSS de volgende hercodering uit:

```
RECODE PARTIJ (2=1)(ELSE=2)
```

Hij was zo zorgvuldig gelijk de labels aan te passen.

```
VALUE LABELS PARTIJ (1) VVD (2) OVERIGE
```

Maar er ging iets mis in een andere opdracht, er kwam een foutmelding, en samen met een stel andere opdrachten werden deze twee opnieuw aangeboden. En nog eens. En nog eens. En toen wist de onderzoeker niet meer hoevaak die hercodering nu correct uitgevoerd was ... Elke keer verwisselden de waarden. Moraal: hercodeer liever een kopie van een variabele, dan het origineel:

```
COMPUTE PARTIJ2 = PARTIJ
RECODE PARTIJ2 (2=1)(ELSE=2)
```

Dit gaat ook bij herhaald uitvoeren goed, en het is te controleren door een kruistabel van PARTIJ tegen PARTIJ2 te maken. Print na elke datatransformatiejob de eerste 5 à 20 cases, en kijk of de transformaties uitpakken zoals bedoeld. Let vooral ook op ontbrekende waarden. Test zonodig zelfs met een klein datasetje met gefingeerde data. Bekijk altijd de verdeling van nieuwe en van veranderde variabelen met CONDESCRIPTIVE, FREQUENCIES en eventueel enkele CROSSTABS. Kloppen min, max, aantal ontbrekende waarden? Bij ingewikkelde logische expressies even een bijbehorende kruistabel maken om te zien of alles goed gegaan is.

Documenteer (met datum) welke veranderingen je aangebracht hebt als je een nieuwe versie bewaard. Het kan heel nuttig zijn een schriftje bij te houden, waarin je per dag kort opschrijft wat je gedaan hebt, welke transformaties en nieuwe files je waarom gemaakt hebt, en welke problemen je tegengekomen bent en wel, half of niet opgelost hebt. Documenteer transformaties ook in de file zelf (een 'update history'). De 'creation date' of datum + tijd van de file is vaak een nuttige indicator, maar volgende maand weet je gegarandeerd niet meer welke versie van welke dag je dan moet hebben. Vergissingen tussen verschillende versies van één databestand liggen erg voor de hand, worden veel gemaakt, worden soms ontdekt, en kunnen gemakkelijk catastrofaal zijn.

Maak niet onnodig veel variabelen apart aan. Een variabele continu èn gedichotomiseerd in het bestand opnemen is alleen nuttig als je die dichotome versie erg veel gebruikt. Anders dichotomiseer je hem met één RECODE vlak voor je hem nodig hebt.

Als je klaar bent met cleanen en met het aanmaken van nieuwe variabelen, maak dan weer een overzicht van alle 1-dimensionale verdelingen (zie het eind van sectie 3). Gooi het vorige overzicht niet weg! Het kan nog belangrijk zijn ze later voor een bepaalde variabele te vergelijken, als er gekke dingen gebeuren (en die gebeuren).

Soms is het om inhoudelijke of om methodologische redenen wenselijk om een variabele eendimensionaal te transformeren voor je hem in het model opneemt. Inhoudelijke redenen zijn bijvoorbeeld: bodem- en plafondeffecten, drempeleffecten, een theorie of onderzoek waaruit volgt dat een bepaalde invloed niet-lineair is, zoals bij 'afnemend grensnut' en bij 'breakdown'. Methodologische redenen zijn in het algemeen afgeleid van een veronderstelling van (vaak multivariate) normaliteit. Let goed op of dit voor de marginale verdelingen verondersteld wordt of bijvoorbeeld (zoals bij regressie) alleen voor bepaalde conditionele verdelingen, of alleen voor residuen. Bij regressie is het effect van een transformatie van de Y-variabele op de residuen in het algemeen niet eenvoudig te doorgronden. Maar goed, als we dan multivariate normaliteit van een stel variabelen nodig hebben, dan impliceert dit univariate normaliteit. De voornaamste, mogelijke schendingen zijn dan: grote scheefheid, ernstige discreetheid (minder dan 5 categorieën), en dikke staarten. Dikke staarten komen in de sociale wetenschappen niet echt veel voor. Het belangrijkste voorbeeld is inkomen, maar dat is toch al voor veel modellen een tamelijk problematische variabele door de grote asymmetrie en door het bodemeffect dat veroorzaakt wordt door het mimimumloon. (Voor een predictor zijn deze bezwaren niet zo groot.) Er is dus nauwelijks een traditie in het 'inkorten' van dikke staarten. Je zou een onevenmachts wortel kunnen nemen, of heel rigoreus transformeren naar rangnummers. Discreetheid

kun je niet door transformeren verbeteren, dus daar is weinig aan te doen. Berucht zijn vooral zeer scheve vijfpuntsschaaltjes. Die leveren weinig informatie op, en wat ze opleveren is vaak methodologisch moeilijk te verwerken. Blijft over: asymmetrie. Dat komt vrij veel voor, terwijl veel methoden daar nogal gevoelig voor zijn. Om een asymmetrische, positiefwaardige variabele te symmetriseren ligt het voor de hand een transformatie te kiezen uit het volgende rijtje:

$$1/x^k, \quad 1/x^2, \quad 1/x, \quad 1/\sqrt{x}, \quad \log x, \quad \sqrt{x}, \quad x^2, \quad x^k$$

Welke transformatie het meest symmetrische beeld oplevert, is vaak goed te beoordelen door voor al deze transformaties de volgende grootheden te berekenen: de mediaan, het midden van het eerste en derde kwartiel, het midden van de grootste en de kleinste waarneming (of van het 5-percentiel en het 95-percentiel). Dat de logaritme wel degelijk op de aangegeven plaats in dit rijtje thuishoort ziet u door de afgeleiden van deze transformaties op te schrijven. Voor variabelen zonder natuurlijke ondergrens, bestaat niet zo'n standaard verzameling transformaties. Daar is dus meer knutselwerk voor nodig.

De vraag wanneer het nodig of wenselijk is variabelen te transformeren, om theoretische redenenen, of om ze meer normaal verdeeld te maken, is niet met een eenvoudige vuistregel te beantwoorden. Zoals gezegd hangt het zowel van inhoudelijke als van statistische argumenten af. Het is verstandig hiervoor tijdig een statisticus te raadplegen.

8 Testen

Test en controleer zoveel als je (zonder al te veel moeite) kunt:

`controleren + controleren + controleren ⇒ kwaliteit`

Computers doen zelden wat je wilt, maar altijd wel wat je vraagt. Daar zit vaak net een vervelend verschil tussen. Controleer voortdurend of er gebeurd is wat je bedoelde.

Ga er NOOIT van uit dat data consistent zijn. Als je dat nodig hebt, test er dan op, en doe iets verstandigs met inconsistente cases (desnoods: weggooien).

Het is een goede gewoonte om bij haast elke run (ook in de analysefase) altijd de eerste 5 cases even te printen (een aantal BMDP-programma's doet dat automatisch). Dan weet je of je de *goede* versie van de *goede* file hebt, of de cases gewogen zijn, en nog het één en ander. Ook een extra `CONDESCRIPTIVE` van de variabelen die je in een multivariate analyse gebruikt is nooit weg (een aantal BMDP-programma's doet dat automatisch). Liefst met ontbrekende waarden op dezelfde manier behandeld als in die multivariate analyse Draai ook een `CONDESCRIPTIVE wvar` (zie 1.h. hierboven).

Als je correlaties uitrekent, maak dan ook wat scattergrammetjes. Is het ongeveer bivariaat normaal? Zijn er uitbijters? Als je regressie doet, bekijk dan de residuen (normaliteit, uitbijters) en bekijk of er erg invloedrijke waarnemingen zijn. (Zie Belsley, Kuh & Welsch, 1980; BMDP-manual; SAS-manual; Weisberg, 1980)

Als je een 'beste model' kiest (bijvoorbeeld predictoren-selectie bij regressie), zet dan een random deel (zeg 100-200 cases) van je cases apart, om daaruit een onafhankelijke en meer zuivere schatting te krijgen van de goodness-of-fit. (BMDP9R levert daar mooie automatische voorzieningen voor.) Dit heet kruisvalidering. Als je het 'beste model' kiest dreig je namelijk teveel aan te passen aan deze toevallige data, en is het moeilijk op andere wijze de grootte van deze kanskapitalisatie te beoordelen.

Test grote jobs, zo mogelijk, eerst alleen op de syntax, en daarna met een kleine dataset. In de meeste programma's kun je de dataset eenvoudig beperken door zoiets op te geven als

```
N OF CASES 100
```

Beperk op een mainframe eventueel ook de aangevraagde hoeveelheid geheugen en CPU-tijd, zodat je een hoge prioriteit krijgt. Testjobs hoor je met hoge prioriteit te kunnen draaien. Bij sommige analyses gaat het mis als je de eerste 100 cases pakt; als de cases gesorteerd liggen, kunnen de eerste honderd cases best variantie nul hebben op een variabele waarop je variantie nodig hebt voor de analyse. Dan kun je misschien testen door een steekproef te trekken uit je data. Ongeveer een fractie x krijg je met zoiets als

```
COMPUTE X = UNIFORM(1)
SELECT IF (X LT x)
```

waarbij `UNIFORM(1)` een random getal tussen 0 en 1 oplevert. In SPSS-PC kan dit korter met

```
SAMPLE x
```

Het schonen van survey data kost meestal vele maanden. Ook een kleine data-analyse waarin alleen maar één vrij eenvoudig model aangepast wordt, met wat variaties en residu-analyse kost al gauw minstens enkele dagen (mits je er erg bedreven in bent en alles direct goed gaat). Bij het analyseren kom je altijd weer fouten in de data tegen, waardoor veel werk overgedaan moet worden. Daarom is een zware investering in het schonen van data zo lonend.

Aan het eind van alle analyses heb je een schoon databestand (stel dat zo mogelijk ter beschikking aan het Steinmetz-archief) en een paar goed passende modellen. Het is een goed idee die paar cruciale analyses nog eens over te doen, of nog beter: over te laten doen door iemand anders, met een ander pakket, en liefst op een deel van de data dat niet eerder gebruikt is en speciaal voor dit doel is weggezet. Vooral ook om te kijken of je de analyse zo duidelijk beschreven hebt, dat een ander het inderdaad na zou kunnen doen. (Een leerzaam stage-project.) Dat zou toch moeten kunnen? Maar dat blijkt in de praktijk eerder uitzondering dan regel ...

Het begrip kwaliteitscontrole wordt natuurlijk veel ruimer gehanteerd dan alleen bij databeheer en dataverwerking. Bij veel productieprocessen vindt kwaliteitscontrole op een professionele manier plaats. De belangrijkste aandachtspunten zijn dan:

- (kwetsbaarheid:) in welke fasen liggen fouten voor de hand,
- (belang:) in welke fasen is hoge kwaliteit belangrijk, en
- (versterking:) hoe zijn die fasen te controleren en bewaken.

Controleren en bewaken is als het maken van extra verbindingen in een steiger. Dubbel uitvoeren maakt het dubbel sterker, maar zwakke verbindingen blijven relatief zwak. Onafhankelijke tests zijn als goede diagonaalverbindingen. Ze maken de constructie veel sterker.

Onder andere doordat wetenschappelijk onderzoek telkens nieuw en telkens anders is, is het net zo kwetsbaar als een nieuw productieproces, nieuwe procedures voor wat-dan-ook, of telkens wijzigende processen (berucht zijn automatisering en sommige onderwijsprogramma's). En als je niet verschrikkelijk hard werkt aan kwaliteit, analyseer je alleen ruis. (Waar wellicht nooit iemand achterkomt, want gerepliceerd wordt er toch bitter weinig.)

9 Welk pakket

Voor het opzetten van vragenlijsten, voor computerondersteund interviewen, en voor datacontrole en serieuze datacleaning zijn de standaard pakketten voor data-analyse echt geschikt. Daarvoor is het PC-pakket Blaise van het CBS een prima hulpmiddel.

Voor heel eenvoudige analyses, eenvoudige regressie en tabelleren, eventueel een hoofdassenanalyse, een factoranalyse of een clusteranalyse kun je uit veel pakketten kiezen. Alle pakketten doen dat wel goed. Gebruiksvriendelijkheid en wat je gewend bent geven dan terecht de doorslag. Voor serieuze standaardanalyses zijn BMDP (op een mainframe) of SAS (ook op de PC) de aangewezen pakketten. Voor iets ingewikkelder analyses en heel mooie grafieken (zoals een correlatietabel van scattergrammen) is bijvoorbeeld STATA (op PC, versie 2.0) een goed doordacht en zeer snel alternatief. GLIM (ook op PC) is prima voor lineaire modellen, en ook voor log-lineaire en logistische regressie. Sterke punten zijn met name de behandeling van categorische variabelen en van interacties, de mogelijkheden om een heel stel modellen te vergelijken, en de filosofie van het pakket. Maar, net als andere programma's voor log-lineaire modellen of logistische regressie, kan het geen ontbrekende waarden aan; en verder is de maximale datamatrix niet echt royaal: op de PC wat minder dan 48000 (aantal eenheden maal (4+aantal variabelen)).

En SPSS? Over SPSS is veel goeds te vertellen, maar niet alleen maar goeds. Voor eenvoudige analyses is SPSS prima. Het manual is overzichtelijk, prima index, handige voorbeelden, uitleg bij interpretatie. De PC-versie loopt lekker. Maar voor iets meer complexe problemen is SPSS kwalitatief veel minder dan BMDP en SAS, en wel op drie punten. Methodologisch is zowel het pakket als het manual dan zwak; het ontbreken van nauwkeurige beschrijvingen en definities wordt nu hinderlijk; en ik kom in SPSS zelfs af en toe echte fouten tegen, waarvan de reparatie toch wel vaak 2-5 jaar op zich laat wachten. Met echte fouten bedoel ik fouten in dingen als significanties in rxc-tabellen (SPSS-9 en ouder), alles met logistische regressie (verbeterd in versie X-3), veel in MANOVA (huidige status mij onbekend), alles met contrasten (in versie X-3 bijna goed), en vrijheidsgraden in log-lineaire modellen met geschatte nulcellen.

Het belangrijkste is echter de methodologische zwakte, die zich daarin uit dat SPSS vrij vaak onopvallend onvolledige of misleidende uitvoer produceert, die minder ervaren data-analisten op het verkeerde been zet. Tussen computeroutput en het begrip van de lezer gaat vaak iets mis. Dit is een berucht zwakke schakel in het hele proces, en SPSS zet juist daar te weinig of zelfs onjuiste waarschuwingsbordjes bij. Het is als een snelle rit over een slingerende bergweg. Een populaire weg, een modieuze sportwagen, schitterend uitzicht, diepe ravijnen, en geen vangrail. Maar het gaat zo onoverzichtelijk hard. En veiligheid (hier: kwaliteit) heeft negatieve marktwaarde, omdat dat het leven niet simpeler maakt. Een handvol voorbeelden. Geen woord in het manual over de tamelijk problematische aspecten van pairwise deletion of van stepwise methoden bij regressie. Geen mogelijkheden voor kruisvalidering daar. Eindelijk zijn er residuenplotjes, maar het zijn de verkeerde (p-p ipv q-q): plotjes waarin extreme residuen juist *niet* te herkennen zijn. Hoofdassenanalyse ondergebracht bij factoranalyse, waardoor de hemelsbrede verschillen tussen beide methoden onder het tapijt verdwijnen, en waardoor veel SPSSers denken dat het èèn een variant van het ander is. Misleidende en zeer onvolledige informatie in het manual over gewichten; in output van veel modulen ontbreekt elke informatie over eventuele weging. Als je, zoals de massa, met dat soort complicaties liever niet geconfronteerd wil worden, dan is SPSS een voor de hand liggend keus.

Voor cruciale analyses die complexer zijn dan wat datamanipulatie, tabelleren en heel eenvoudige regressie-analyse, vertrouw ik persoonlijk toch liever niet op één pakket.

Uiteraard zijn er verder in en buiten het ICS talloze gespecialiseerde programma's voorhanden voor allerlei problemen waar de grote pakketten (nog) geen oplossing voor bieden. Multidimensionale schaling, lineaire structurele relaties, latente variabalen, robuuste schatters, kleine-steekproefverdelingen in rxc-tabellen, ontvouwing, multiniveau modellen, allerlei asso-

ciatiemodellen in rxc-tabellen, om er enkele te noemen.

Voor het zelf programmeren van nieuwe methoden waarvoor nog helemaal geen programmatuur beschikbaar is, kan men het best gebruik maken van een pakket met goede macrofaciliteiten, zoals GLIM, Gauss, Matlab, ISP, APL en SAS-IML. De laatste twee zijn klassiekers. Een macro is een programma of subprogramma geschreven in de gebruikerstaal van een pakket, waarin je naast besturingsopdrachten zoals 'IF ... THEN ... ELSE ...' gebruik kunt maken van de opdrachten van het pakket voor bijvoorbeeld datamanipulatie, regressie, en (liefst ook) matrixmanipulatie. Uiteraard is voor het schrijven van macro's meer kennis en ervaring nodig dan voor het simpele gebruik van een standaardpakket.

Samenvattende conclusies

- Ook als je absoluut zeker van je zaak denkt te zijn heb je het in een paar procent van de gevallen mis. Bouw daarom controles in. Doe dingen dubbel, of beter nog, laat een ander het de tweede keer doen; en doe het de tweede of derde keer op een andere manier dan de eerste keer. Bedenk dat fouten cumulatief werken. Telkens een paar procent fouten levert al gauw rotzooi op. Test zoveel je kan.
- Zorg dat je van alle stappen kunt bewijzen hoe je het gedaan hebt. Bewaar en documenteer datamodificatie-jobs. Zorg dat je ze makkelijk opnieuw kunt uitvoeren vanaf een oude datafile als je er een fout in ontdekt. (Gebeurt zeker.)
- Wees je ervan bewust waar kwaliteit van de data belangrijk is, en probeer het daar te meten, en op voldoende niveau te brengen.
- Zorg dat de data zelfdocumenterend zijn: goede namen die makkelijk te typen zijn, correcte en verhelderende labels voor variabelen, waarden, cases, en voor verschillende versies van files. Toelichting bij eventuele gewichten. Overzichtelijke definities en een goede organisatie en administratie zijn het halve werk.
- Kruisvalideer modelselecties of nog beter: de hele analyse.
- Consulteer een deskundige, waar je zelf mogelijk niet over voldoende kennis of ervaring beschikt. Mogelijk statistische onderwerpen zijn: steekproefopzet, operationalisaties, schaalconstructie, standaardisatie (of niet), modelkeus, opzet van de analyse, het vergelijken van groepen, wegen, behandeling van ontbrekende waarnemingen, behandeling van zeer scheefverdeelde variabelen, goodness-of-fit en modelselectie, kruisvalidatie, controle van modelassumpties (inclusief robuustheid en invloedrijke waarnemingen), problemen met kleine steekproeven en met hele grote steekproeven.
  Wees kritisch ten aanzien van methoden 'die je in de literatuur veel tegenkomt'; methoden en technieken zijn in de sociale wetenschappen nogal modegevoelig; er wordt veel slaafs nageaapt, minder goede gewoonten hebben een akelig lange overlevingsduur, en er zijn veel publikaties (inclusief manuals en leerboeken) waar methodologisch veel op aan te merken is.

Literatuur

Belsley, D.A., E. Kuh, & R.E. Welsch, Regression diagnostics. Wiley (1980).
Little, R.J.A. & D.B. Rubin, Statistical Analysis of Missing Data. Wiley (1987).
Rubin, D.B., Multiple imputation for nonresponse in surveys. Wiley (1986).
Siegel, S., Nonparametric statistics. McGraw-Hill (1956).
Weisberg, S., Applied linear Regression, Wiley (1980).