# ELEMENTAIRE INLEIDING SPSS-SYNTAX

**Ten behoeve van studenten Sociale Wetenschappen Vrije Universiteit Amsterdam**

**Harry B.G. Ganzeboom**[1]

**Versie 6, 11 november 2020**

**Inhoud**



[1] Dit document gaarne citeren als: Ganzeboom, Harry BG (2020). Elementaire SPSS syntax [Versie 6, 11 november 2020]. Amsterdam: Vrije Universiteit. Geraadpleegd via www.harryganzeboom.nl

| VERKEERDE GEWOONTEN | GOEDE GEWOONTEN |
|---|---|
| | |
| Eerst de datafile openen en dan pas de syntax. | Open de datafile via de syntax. |
| | |
| Het bewaren van output en data files. | Alleen je syntax files bewaren. Geef ze overzichtelijke namen en onthoud waar ze staan. |
| | |
| Het zoeken naar resultaten in de output file. | Opnieuw het stukje synrax runnen waarin je geïnteresseerd bent. |
| | |
| Constructie van index-variabelen via factorscores. | Gebruik liever: **`Compute index = mean (indicatoren)`**. Op die manier repareer je item-non-response. |
| | |
| Vergeten de instellingen van Edit > Options > Output labels in te stellen. | Altijd de instellingen van Edit > Options > Output labels instellen. |
| | |
| Het bewaren van tussentijdse resultaten in geconstrueerde variabelen. | Run gewoon je syntax opnieuw, dan weet je wat je gedaan hebt. |
| | |
| Het handmatig veranderen van data in het data-screen. | Werk altijd met recode in je syntax. |
| | |
| Het kiezen van zeer korte of zeer lange variabelennamen. | Hou het kort, maar duidelijk. Voeg uitvoerige informatie toe in het variable label. |
| | |
| Het bewerken (bv. ompolen) van oorspronkelijke variabelen. | Gebruik altijd **`recode .. into`** als je iets verandert aan een variabele. |
| | |
| Toevoegen van comments in je syntax om de bedoelingen toe te lichten | Wees spaarzaam met commentaar. De syntax moet zo helder zij dat het geen commentaar behoeft. |

# ELEMENTAIRE INLEIDING SPSS-SYNTAX

## WAAROM SPSS SYNTAX GEBRUIKEN?

Zoals veel windows-programma's kun je Spss aansturen door het aanklikken van schermpjes, menuutjes en opties. Er is een alternatief voorhanden, dit is de Spss-syntax. De syntax (een 'script-taal') geeft weer hoe Spss feitelijk werkt. Hoewel het voor de beginner handig lijkt om alleen de klikmenu's te gebruiken, en de syntax er dan nog gecompliceerd uitziet, is het belangrijk om vanaf het begin de aansturing van Spss via syntax uit te voeren. Dat heeft namelijk de volgende voordelen:

- Het gebruik van syntax is de enige manier om anderen inzage in je werkwijze te geven. Dat geldt niet alleen voor docenten, maar ook anderen met wie je samenwerkt in projecten.
- Alleen bij het gebruik van syntax kun je je statistische analyses systematisch en stapsgewijs opbouwen. Als je een fout maakt, laat zich die opsporen en verbeteren.
- Het gebruik van syntax is uiteindelijk veel sneller dan het vermoeiende en repetitieve aanklikken van variabelennamen, statistische opties etcetera. Met bestaande syntax kun je namelijk gemakkelijk variëren.
- Sommige geavanceerdere mogelijkheden van Spss zijn alleen maar beschikbaar in syntax.

Om deze redenen kiezen we er bij de Spss-practica ervoor om je alle opdrachten via een syntax te laten uitvoeren. De resultaten van je analyses moet je dan ook in de vorm van een syntax aanbieden. Helaas bieden de meest toegankelijk inleidingen in Spss (zoals het veel gebruikte en op zichzelf zeer handige boekje van De Vocht of het uitgebreide manual van Field) weinig leidraad voor het samenstelling van syntax[2]. Gelukkig kun je syntax laten maken door Spss zelf, door namelijk telkens via de optie *'paste'* de gevolgen van je klikhandelingen naar een 'syntax-file' te laten schrijven. Door deze syntax vervolgens goed te bestuderen zie je vanzelf hoe je hierop kunt variëren, voornamelijk door het vervangen en uitbreiden van variabele elementen (zoals variabelennamen en values). Wil je meer weten, dan kun je te rade gaan bij de helpfunctie.

Als je je in de syntax en de syntax hulp verdiept, zul je ontdekken dat de soor spss zelf gegenereerde syntax erg uitvoerig en daardoor onoverzichtelijk is. Een groot deel s\is overbodig omdat het standaard opties betreft, en een ander deel kun je handig afkorten. Dat maakt het allemaal veel overzichtelijker.

Hieronder geven we een aantal tips hoe met de syntax om te gaan en een overzicht van enige zeer belangrijke en veel gebruikte syntax statements. Eerst geven we enige tips over hoe je handig met Spss kunt omgaan.

## DRIE SOORTEN SPSS-SCHERMEN

- **Data-file**: dit is de data-matrix waarop je de statistische bewerkingen uitvoert. Je kunt hem bewerken als een spread-sheet, bv. kolommen copiëren en uitwisselen met Excel. Anders dan in Excel kun je in Spss altijd maar één datafile open hebben.

[2] Wel geschikt is: Ten Grotenhuis, M. & Chr. Visscher (2009), SPSS met syntax. Assen: Van Gorcum. Zie ook hun website: http://www.ru.nl/methodenentechnieken/syntax/home. Ook veel aandacht voor syntax heeft: Pallant, J. (2013). SPSS Survival Manual. 5th Edition. Maidenhead: Open University Press.

- Data-files worden door Spss standaard opgeslagen met de extensie **`.sav`** (staat voor: saved file). Je kunt ook opslaan in of lezen van Excel (en andere database formaten).
- Een datafile heeft twee 'views': 'data-view' en 'variables-view'. De data-view is de eigenlijke datamatrix, met de eenheden *(cases)* in de rijen en de variabelen in de kolommen. In de variables-view kun je variabelen labelen en hun format (bv. hoeveel zichtbare decimalen) definiëren.
- Hoewel het heel belangrijk is dat je weet hoe een data-matrix eruit ziet, is het onbelangrijk om er veel naar te kijken. Het is een verkeerde gewoonte om data in de datamatrix met de hand te veranderen of er variabelen bij te typen – doe dit via syntax, alleen dan kun je later zien wat je gedaan hebt.
- Een bewerkte datamatrix moet je niet te bewaren. Integendeel: je analyse moet erop gericht zijn om uit de oorspronkelijke datamatrix via een heldere syntax in één keer je statistische resultaten te verkrijgen. Bewaar ook geen tussenstappen, daar verlies je gauw het overzicht.

- **Output-files**: hierin schrijft SPSS de resultaten van je analyses. Ook van output-files is het onbelangrijk om ze te bewaren. Je kunt ze namelijk altijd opnieuw maken door de syntax opnieuw te runnen.
  - Het navigeren in de Spss output files is een beetje moeizaam, vooral als het om wat grotere tabellen gaat. Het helpt om een tabel te copiëren naar Excel, daarin kun je gemakkelijker zoeken en formatteren.
  - Het is een goede gewoonte output-files niet alleen na afloop, maar ook periodiek tijdens je analyses weg te gooien – op die manier zie je namelijk alleen het resultaat van je laatst uitgevoerde analyses en alleen daarin ben je meestal geïnteresseerd. Je kunt output-files weggooien door ze weg te klikken (X) of met de keystroke CTRL-A, DEL. Helaas bestaat hiervoor geen syntax-statement.
- **Syntax-files:** Syntax-files bevatten de commando's die de data-matrix bewerken tot de gewenste resultaten. Het uiteindelijke doel van je Spss-analyses moet zijn om een heldere en leesbare syntax-file maken. ***Syntax-files moet je dus goed bewaren!***
  - Syntax-files worden door Spss opgeslagen met de extensie **`.sps`**. Het zijn gewone text (ascii) files, die je ook met notepad (kladblok), wordpad e.d. kunt bewerken.
  - De syntax-files zijn de kern van je analyse-werk. Ga zo te werk dat je telkens een nieuw stukje syntax maakt, dat selecteert en runt, tot je over het geheel tevreden bent. Run dan het geheel.
  - Leer met de volgende keystrokes te werken.
    - SHIFT PIJL hiermee kun je stukken syntax selecteren
    - CTRL-R runt het geselecteerde deel van je syntax
    - CTRL-A CTRL-R runt de gehele syntax
    - ALT TAB voert je door alle open windows schermen en is de snelste manier om van viewer naar syntax te gaan.

Als je tabellen, figuren etc. van de viewer naar Word of Excel wilt overbrengen, gaat dat via de rechtermuisknop. Als je handig bent met tabellen in Word of formatteren in Excel kun je er al snel iets moois van maken. Bij Excel heb je dan wat meer mogelijkheden dan in Word.

**INSTELLINGEN EN OPTIES**

De begininstellingen van SPSS zijn niet erg geschikt voor serieus analysewerk. Je kunt ze wijzigen via 'options' in het 'edit'-menu. Enige tips:

- 'General': Display names, No scientific notation, Open only one dataset at a time.

- ‘Viewer’: ‘display commands in log’, kies als font: ‘Courier New 12 pt. Bold’.
- ‘Output labels’: zet dit allemaal op ‘names and labels’, respectievelijk ‘values and labels’.
- ‘Pivot tables’: kies een ‘Compact academic style’.
- File locations: Last folder used.

De syntax om deze instellingen te verkrijgen, is:

```
SET OVars Both ONumbers Both TVars Both TNumbers Both.
SET CTemplate None.
SET Scalemin=24.
SET TLook 'C:\Program Files\SPSS\Looks\Academic (VGA).tlo' TFit Labels.
SET CCA '-,,, ' CCB '-,,, ' CCC '-,,, ' CCD '-,,, ' CCE '-,,, ' .
SET Format=F4.0 Epoch=Automatic.
```

## UTILITIES, DATA-DEFINITIE

**Algemeen:** Elk Spss-statement begint met in de eerste positie van een regel en eindigt met een punt (.). Het is een goede gewoonte om vervolgstatements niet op de eerste positie te beginnen, maar het mag wel. ***Als je zelf syntax maakt, ontstaan er vaak fouten als je de punt op het einde vergeet.***

*****`commentaar`**.**

- Een statement dat wordt voorafgegaan door asterisk (*) en eindigt met een punt (.) is een commentaar-regel, die door Spss wordt afgedrukt maar niet wordt gelezen. Door commentaar in je syntax op te nemen kun je jezelf en anderen informeren waarom je een bepaalde analyse-stap doet, wat de resultaten waren, etc. Ook kun je stukken werkzame syntax in commentaar veranderen door er even een asterisk voor te zetten.
- Het commentaar mag zo lang zijn als je wilt, als je maar niet in de eerste positie begint. Het is echter een goede gewoonte elke regel te beginnen met een asterisk.
- Gebruik naast *****`commentaar` ook lege regels om je syntax-file leesbaar te houden.
- Wees zuinig met commentaar. Het gaat om de helderheid van de syntax, niet om het commentaar.

**Get file =** `"pad-naam\bestand.sav"`**.**

- In het eerste statement van je syntax instrueer je Spss een bestaande data-file te lezen. Het is handig om je syntax altijd hiermee te beginnen. Als je ontevreden bent over je tussenstappen of wanneer je een fataal verkeerde bewerking hebt uitgevoerd, kun je zo heel snel weer bij het begin beginnen.
- Laat het **get file** statement altijd door Spss genereren via ‘paste’. De Windows-padnamen kunnen bijzonder ingewikkeld zijn en leiden gemakkelijk tot fouten.
- Enkele en dubbele quotes zijn in Spss uitwisselbaar, maar mogen niet door elkaar gebruikt worden.

**Variabelen-namen**

- Een variabelen-naam in Spss mag bestaan uit max. 52 letters en cijfers (ook , _ en @ zijn toegestaan, maar niet , - + * & $ = ? etc.). Je moet beginnen met een letter. Gebruik niet meer dan 16 letters.
- Als je gegevens invoert uit een vragenlijst, kies dan NIET voor inhoudelijke namen (**`sekse`**, **`opleiding`**), maar voor variabelennamen die eenduidig corresponderen met de nummering van vragen in de vragenlijst (**`V10a`**, **`Var10x`**). Alleen op die manier kun je je gegevens helder documenteren. Inhoudelijke namen maak je aan wanneer je statistische bewerkingen doet. Kies daarbij voor namen die de richting van de gebruikte codering suggereren, bv. **`vrouw`** i.p.v. **`sekse`**.
- Grote en kleine letters maakt niet uit (spss is *case-insensitive*), maar worden wel gehandhaafd op basis van de eerste aanmaak van een variabele. Hiermee kun je variabelennamen leesbaarder maken.

**Soorten variabelen.**

- Er zijn twee belangrijke soorten variabelen in Spss: numerieke variabelen en alfanumerieke variabelen (door Spss 'strings' genoemd). In numerieke variabelen staan cijfers waarmee je als getallen kunt rekenen. In strings wordt tekst opgenomen – je kunt er wel overzichten van maken, maar niet mee rekenen. Let op dat ook cijfers strings kunnen zijn en dat je er dan niet mee kunt rekenen. Welke variabelen strings zijn, vind je in de variables-view van het data-scherm. Je kunt strings omzetten in numerieke gegevens en omgekeerd via **`recode`**.
- Er zijn nog veel meer soorten variabelen mogelijk in Spss, zoals datums of geldbedragen. Deze zijn voor de beginner in de praktijk niet belangrijk. Het gebruik van strings is soms wel erg handig.

**Varlist: `VAR1 to VAR10`**

- Veel Spss-procedures en –transformaties kun je in één klap op meerdere variabelen tegelijk uitvoeren. Deze hoef je niet allemaal op te sommen, je kunt ze aanduiden met **`VAR1 to VAR10`**. We duiden zo'n reeks ook wel aan als een *`varlist`*. Hoeveel en welke variabelen geïmpliceerd zijn, hangt af van hun volgorde in de data-file.

**`Var label`** *`varnaam "Het var-label van je voorkeur".`*

- Via het **`var label`** statement kun je snel een (nieuw) label toekennen aan een variabele. Dit kan ook via de variables-view van het data-scherm, maar met het var label statement gaat het gemakkelijker.
- Maak geen var-labels langer dan 60 letters.

**`Value labels`** *`varnaam`* **`(1)`** *`"label-1"`* **`(2)`** *`"label-2"`* **`.`**

- Via het **`value labels`** statement kun je een (nieuw) label toekennen aan de (nieuwe) waarden van een variabele. Dit kan ook via de variables-view van het data-scherm. Via je syntax statement kun je het voor meerdere variabelen tegelijk doen.
- Maak geen value labels langer dan 16 letters.
- Als je allemaal nieuwe value labels wilt maken, gebruik je **`value labels`**, als je labels wilt toevoegen en de oude wilt laten staan (voorzover je ze niet overschrijft), gebruik je **`add value labels`**.

**Missing values**

- We spreken van 'missing values' wanneer gegevens (gedeeltelijk) ontbreken, bijvoorbeeld omdat een respondent geweigerd heeft op een bepaalde vraag antwoord te geven. Missing values worden door Spss niet meegenomen in statistische bewerkingen.
- Er zijn twee soorten missing values:
  - **Sysmiss**: er is inderdaad geen waarde bekend – in de datamatrix zie je deze gevallen als een puntje.
  - 'User-defined missing values': er is wel een waarde, maar je hebt via de datadefinitie aangegeven die als missing te willen beschouwen. Het is een goede gewoonte voor deze missings een opvallende waarde te kiezen (bv. 99 of -1).
- User defined missing values geef je aan op het datadefinitiescherm of via **Missing values** *varlist* **(-1, 99).**
- Soms zijn missing values het gevolg van de structuur (filtering) van een vragenlijst en is het invullen van een plausible waarde mogelijk: bv. het arbeidsinkomen is 0 voor niet-werkenden. Een geavanceerde manier om met missing values om te gaan kun je vinden **Analyze > Missing Value Analysis** en **Analyze > Multiple Imputation**.
- De goede omgang met missing values is een heel belangrijk onderdeel van statistische analyses. Met name in vragenlijstgegevens treden er vaak veel – verspreide – missing values op. Bij elke stap in de analyse moet je je dan afvragen hoeveel geldige waarnemingen je hebt en hoe je verlies van gegevens kunt repareren.

**DATA-TRANSFORMATIES**

Onder 'Transform' zijn verschillende bewerkingen van de gegevens gedefinieerd. We behandelen twee zeer bruikbare: **recode** en **compute**. Ze hebben gemeen dat het nogal ingewikkeld is om ze via klikschermen te construeren, terwijl de gegenereerde syntax zelf niet zo ingewikkeld is. Je hebt er veel voordeel van als je hier zelf wat syntax beheerst.

**Hercodering**

```
Recode varnaam (1=100)(2=200)(3=500).
Recode varnaam (1 2 3=100)(4 thru 6=200) into newvar.
Recode varnaam (lo thru 1=0)(5 thru hi=5)(else=copy)
     into newvar.
Recode strnaam ('man'=0)('vrouw'=1)(else=sysmiss) into vrouw.
```

- Het **recode** statement is een bijzonder eenvoudig en krachtig statement om de waarden van een variabele te veranderen (bv. samen te voegen) of in een nieuwe variabele terecht te laten komen. Het is de eenvoudigste manier om een nieuwe kolom in de datamatrix aan te maken.
- Je kunt meerdere oude waarden opsommen **(1 2 3 4)** of via een **thru** een range aanduiden **(1 thru 4)**. De bestemmingswaarde is er altijd maar een.
- Met **recode** kun je bestaande variabelen veranderen (dan laat je **into** weg), maar het is meestal aan te raden om een nieuwe variabele via **into** aan te maken. Op die manier blijft de oude informatie ongemoeid.

- Als je een alfanumerieke variabele (een string) in een numerieke variabele hercodeert, moet je de bestaande waarden in quotes zetten.
- Let op de mogelijkheid om alle overige waarden via **`else`** te benoemen.
- De bestemmingswaarde **`sysmiss`** geeft aan dat de betreffende waarden als system missing worden behandeld. Als je bij **`recode .. into`** een value niet hercodeert, wordt dit vanzelf een **`sysmiss`**.
- Let op de goede gewoonte om dichotome (indicator, tweewaardige) variabelen te coderen in de values 0 en 1.

**Berekeningen**

```
Compute VAR3 = VAR1/VAR2.
Compute VAR3 = VAR1*VAR2.
Compute VAR3 = VAR1+VAR2.
Compute VAR3 = VAR1-VAR2.
Compute VAR3 = (VAR1+VAR2)/2.
Compute VAR3 = mean(VAR1,VAR2).
```

- **`Compute`** is de rekenfunctie van Spss. Je kunt er alles mee wat je rekenapparaat kan (en nog veel meer), alleen heeft het nu betrekking op variabelen (kolommen in de datamatrix met een reeks getallen), niet op één getal.
- De mogelijkheden zijn zeer uitgebreid, maar de elementaire syntax is zeer eenvoudig.
- Let op het zesde statement, waarin VAR3 het gemiddelde wordt van VAR1 en VAR2. Het bijzondere van dit statement is dat het doorrekent ook wanneer VAR1 of VAR2 missing values bevatten. Het neemt dan het gemiddelde van de niet missende waarden. Bij het vijfde statement wordt VAR3 een sysmiss als VAR1 of VAR2 missend is. Statement 5 en statement 6 geven bij missing values verschillende resultaten!

Een derde veel gebruikt transformatie is:

**`Rank`** *`VAR3`* **`/ percent.`**

- Het **`rank`** statement berekent verschillende soorten rangscores van een variabele (en is daarom een transformatie). De toevoeging **`/percent`** berekent percentielscores tussen 0 en 100, een belangrijke manier om variabelen te standaardiseren. De resulterende percentielscores heet *`PVAR3`*.
- Als je percentielscores tussen 0 en 1.00 wil, moet je het resultaat nog eens door 100 delen. Ook kun je /percent vervangen door **`/proportion`**.

Een soortgelijke transformatie vind je onder Analyze > Descriptives.

**`Desc VAR /save.`**

- Dit resulteert in een Zscore van VAR en die wordt door Spss standaard ZVAR genoemd.

## SELECTEREN, FILTEREN EN SPLITSEN VAN DE DATA-MATRIX

Het is vaak nuttig om statistische bewerkingen uit te voeren voor slechts een deel van je gegevens: bv. alleen vrouwen (en daarna de mannen). SPSS heeft daarvoor verschillende mogelijkheden:

```
Filter by female
(…)
Filter off.
```

Bij **filter** hebben alle procedures tussen **filter by** and **filter off** betrekking op het deel van je data dat gedefinieerd is door female=1. Handig, maar denk om twee dingen:

- De filter variabele (hier: *female*) moet een 0/1 variabele zijn. Wil je hierna de mannen doen, dan zul je ook een 0/1 variabele *male* moeten definiëren.
- Filter slaat alleen de data over die je hebt uitgefilterd, deze blijven in de data-matrix ook zichtbaar, met een streep erdoorheen. Als je maar op een klein stukje van je matrix wilt werken, kan filteren toch erg veel tijd kosten.

```
Temp.
Select if (female = 0).
(…)
```

Selecteren verwijdert het stuk van je data matrix dat je niet geselecteerd hebt. Het wordt permanent verwijderd, tenzij je er het woordje **temp (temporary)** voor zet. Maar in dat geval blijft de **select** alleen maar gehandhaafd tot de volgende procedure. Je kunt **Select** dus niet zoals **Filter** voor meer procedures gebruiken.

```
Sort cases by female.
Split file by female.
(…)
Split file off.
```

**Split file** zorgt ervoor dat al je procedures (…) worden uitgevoerd voor alle groepen (in dit geval: mannen en vrouwen) afzonderlijk. Dit is erg handig, bv. bij landenvergelijkend onderzoek en ook nog eens heel erg snel.

Je data moeten wel eerst gesorteerd zijn op de split-variabele. Dat kan wel tijd kosten.

## LOGISCHE RELATIES

In SELECT IF komen logische relaties voor. Ze kunnen zowel met woorden als met symbolen aangeduid worden. De belangrijkste ervan zijn:

| Woord | Symbool | Betekenis |
|---|---|---|
| EQ | = | Equal |
| NE | <> | Not equal |
| LE | <= | Lower or equal |
| LT | < | Lower than |
| GE | >= | Greater or equal |
| GT | > | Greater than |

| | | |
|---|---|---|
| **AND** | **&** | And |
| **OR** | **\|** | Or |

De volgende statements selecteren vrouwen in de leeftijd 18-64:

```
select if (age ge 18 and age le 64).
select if (female eq 1).
```

Het kan ook gecombineerd en symbolisch worden opgeschreven:

```
select if ((age <= 18 & age <= 64) & (female = 1)).
```

Een andere toepassing van de logische relaties vind je in de **DO IF (...). (transformaties) END IF**. Hiermee kun je aangegeven transformaties conditioneel uitvoeren. Enige voorbeelden.

```
do if (age ge 18 and age le 64).
recode agecat (21=1)(else=0) into agecat21.
recode agecat (30=1)(else=0) into agecat30.
recode agecat (40=1)(else=0) into agecat40.
recode agecat (50=1)(else=0) into agecat50.
recode agecat (60=1)(else=0) into agecat60.
end if.
```

Deze statements maken dummyvariabelen aan die **sysmiss** zijn voor alle leeftijden buiten de range 18-64.

## STATISTISCHE PROCEDURES

Voor statistische procedures (je vindt deze onder het tabblad 'Analyze') geldt dat het moeilijker is om de syntax zelf correct te produceren. Hier zul je vaker de syntax door Spss zelf laten genereren. Daarop kun je dan weer variëren door variabelennamen te vervangen. De door Spss gegenereerde syntax bevat wel vaak veel overbodige details. Daarom behandelen we de volgende zeer veel voorkomende procedures in hun verkorte vorm.

**Freq** *varlist*.

- Toont de frequentie-verdeling van de variabelen. Vaak een goed begin van je bewerking, of handig om het resultaat van een eerdere bewerking te checken.

**Desc** *varlist*.
**Desc** *varlist* **/save.**

- Berekent de descriptives (beschrijvende grootheden, zoals gemiddelde, standaarddeviatie, range, mediaan, etc.) van de variabelen.
- **Desc** kun je niet toepassen op alfanumerieke (string) variabelen.
- Het tweede statement is zeer nuttig: het produceert Z-scores van de betrokken variabelen, die weer worden aangeduid als *ZVAR1*, *ZVAR2*, etc. Dit is naast **Rank** de belangrijkste manier om variabelen te standaardiseren.

**Crosstabs** *VAR1* **by** *VAR2* **/cells=count row col /stat=chisq corr.**

- **Crosstabs** geeft een kruistabel van twee variabelen, en kan daarbij rij/kolom percentages en associatiematen en bijbehorende significantietoetsen berekenen. De belangrijkste daarvan zijn de chi-kwadraat en pearson / spearman correlaties.

- Je kunt ook meerdimensionele kruistabellen maken door meer **by**'s toe te voegen,

**Means** *VAR1* **by** *VAR2*.

- Geeft conditionele gemiddelden van VAR1 voor categorieën van VAR2. Dit is de meest gebruikelijk voorwas voor variantie-analytische en regressie-modellen.
- *VAR2* mag een string-variabele zijn.
- Via het **Means** statement kun je ook een (oneway) ANOVA opvragen. Voeg dan de optie **/stat=anova** toe.
- Je kunt ook een meerdimensionele opsplitsing opvragen door meerdere **by**'s toe te voegen.

**Regr /dep=y /enter=x1 /enter=x2.**

- Geeft een multipel lineair regressiemodel van de vorm Y = B0 + B1*X1 + B2*X2.
- Door meerdere malen een **/enter** statement te geven, krijg je een stapsgewijs model, waarin je gemakkelijk bestudeert hoe de toevoeging van controlevariabelen / mediatoren een eerder geschatte coëfficiënt verandert.
- Multipele regressie is de meest gebruikte statistische techniek om de invloed van oorzaakvariabelen (X-vars) op een enkele gevolg (Y) te modelleren. Andere vormen van statistische analyse zijn hiermee equivalent (zoals ANOVA), varianten (zoals logistische regressie) of uitbreidingen (zoals het General Linear Model). Een goed begrip van alle details van het regressiemodel is onontbeerlijk voor elke kwantitatieve analyse.
- Interpretatie van de SPSS regressie output wordt uitvoerig besproken in een andere handout. Tip: lees de output van onder naar boven. Bekijk de informatie in de volgorde:
  - Het regressiemodel staat onder B. Het gestandaardiseerde regressiemodel staat onder Beta.
  - De coefficienten worden op significantie getoetst mbv de SE, T en sig. (= p=waarde).
  - De ANOVA tabel bevat de sum-of-squares en mean-squares, waarmee een over-all F-test wordt geconstrueerd.
  - De verklaarde variantie wordt weergegeven via (adjusted) R-squared. De multiple correlatie R is hieruit de wortel en geeft aan in welke mate de Y-var met de gemodelleerde X-vars voorspeld kan worden.

## HET MAKEN VAN EEN MULTIPLE INDICATOR INDEX

Een veelvoorkomend analyseprobleem is hoe we uit een verzameling indicatoren op de beste wijze een samenvattende index kunnen samenstellen. We gaan uit van de volgende situatie:

- We veronderstellen een meetmodel waarin een 'latente variabele' zich uitdrukt in de score op meerdere geobserveerde indicatoren. Dit model impliceert dat de indicatoren onderling gecorreleerd zijn; de kwaliteit van de meting hangt af van de samenhang tussen de latente score en de geobserveerde scores.
- Er zijn tenminste drie indicatoren per veronderstelde latente score. Met één indicator alleen kun je niets zeggen over meetkwaliteit. Met twee indicatoren kun je wel zien dat de meting slecht of goed is, maar weet je niet aan welke indicator het ligt. Alleen bij ten minste drie indicatoren of meer kun je ontdekken wat een slechte en wat een goede indicator is.

We nemen standaard vijf stappen.

### Stap 1: beschrijving van de variabelen

```
DESC VAR1 TO VAR10.
```

De opgevraagde descriptives geven aan hoeveel geldige waarnemingen er zijn per variabele, wat het bereik van de scores is, wat hun gemiddelde en standaarddeviatie is. Let op de volgende dingen:

- Hoe is het patroon van geldige en missende waarnemingen? Zijn er veel missing values, hoe komt dit? Is hier iets te repareren, door een plausible waarde in te vullen?
- Zijn er uitschieters of wild codes? Zo ja: repareren door ze terug te coderen tot een aannemelijke waarde of missing te maken.
- Zijn de standaarddeviaties een beetje in dezelfde orde van grootte? ***Zo niet: standaardiseren via P- of Z-transformatie.***

### Stap 2: Dimensionele analyse

De meest gebruikelijke veronderstelling bij het construeren van een index uit meerdere indicatoren is dat de indicatoren een gemeenschappelijke inhoud hebben, waardoor respondenten daar op soortgelijke manier op reageren. De toets of deze veronderstelling juist is, ligt bij de samenhang tussen de indicatoren: ze dienen onderling gecorreleerd te zijn en die correlaties dienen eendimensioneel te zijn, dat wil zeggen niet in subgroepjes uiteen te vallen.

De meest eenvoudige manier om dimensionaliteit te onderzoeken is het bestuderen van de correlatiematrix:

```
CORR VAR1 VAR2 VAR3 VAR4.
```

Door het variëren van de volgorde van de variabelen krijg je gemakkelijk meer overzicht. Groepeer de hoog correlerende indicatoren naast elkaar en dan zie je vanzelf of er subgroepjes zijn.

Een meer formele manier van zoeken naar multidimensionaliteit is factor-analyse > Analyze > Data Reduction > Factor. Zie hiervoor aparte hand-out / stappenplan.

### Stap 3: Betrouwbaarheidsanalyse

In deze stap bekijken we of de onderlinge samenhang tussen de indicatoren groot genoeg is om ze bij elkaar te kunnen nemen.

```
RELI /VAR=VAR1 TO VAR10 / SUMM=TOTAL.
```

Dit geeft een tabelletje waarin wordt berekend hoe hoog Cronbach's alfa (de betrouwbaarheidscoëfficiënt van een multiple indicator index) zou zijn bij weglating van een indicator. De betrouwbaarheid van een index is een functie van twee zaken:

- De gemiddelde correlatie tussen indicatoren
- De hoeveelheid indicatoren

Je kunt een lage gemiddelde correlatie tussen indicatoren (duidend op geringe samenhang tussen latente en geobserveerde scores) compenseren door veel indicatoren! Het tabelletje dat de output van **`reliability`** daarover biedt, geeft meestal uitsluitsel wat de optimale keuze

is van hoeveelheid indicatoren en de sterkte van hun onderlinge correlatues. Let bij de analyse op de volgende dingen:

- **`Reliability`** veronderstelt eendimensionaliteit (alle indicatorene meten een en hetzelfde kenmerk en ook dat de alle indicatoren in dezelfde richting gepoold zijn.
- **`Reliability`** is een van de weinige programma's in spss, die niet kan omgaan met pairwise deletion of missing values. Blijf altijd goed naar de N kijken om te weten of je er niet teveel kwijt bent.
- Betrouwbaarheidanalyse moet je met één stap per keer doen. Als er twee indicatoren zijn die weggelaten kunnen worden, doe het dan eerst met de ene, bereken daarna opnieuw de **`reliability`** en doe pas daarna de tweede stap.
- We spreken van voldoende betrouwbaarheid wanneer alfa .80 of hoger is. Tussen 0.65 en 0.80 is de betrouwbaarheid nog redelijk, maar treedt toch al behoorlijk vertekening van verbanden op door onbetrouwbare meting.

Het einde van deze analyse is dat we besluiten om met een beperkte set indicatoren verder te werken.

NB: Betrouwbaarheidsanalyse veronderstelt eendimensionaliteit, je kunt er meerdimensionaliteit niet mee opsporen, daarvoor moet je correlaties of factor-analyse gebruiken. Factoranalyse geeft ook aanwijzingen over hoe betrouwbaar de individuele indicatoren zijn, maar geeft geen uitsluitsel over de betrouwbaarheid van de index.

**Stap 4: Bereken de index**

Veruit de beste manier om een index uit meerdere indicatoren te berekenen is via hun ongewogen gemiddelde:

```
COMPUTE INDEX = MEAN(VAR1,VAR2,VAR3,VAR4,VAR6).
```

Kanttekeningen:

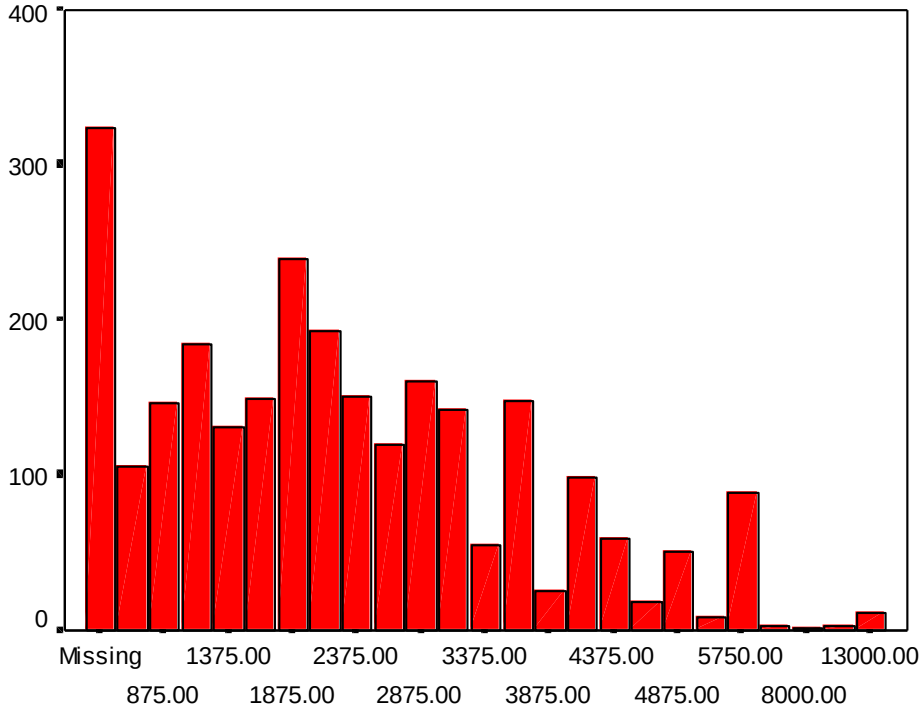
- Het statement is heel geschikt om van missing values problemen af te komen: als iemand op een van de indicatoren niet scoort, wordt een gemiddelde berekend over de resterende indicatoren.
- De syntax luistert hier nogal nauw (je mag de komma's niet door spaties vervangen) en het klikscherm werkt hier niet goed.

**Stap 5: Beschrijf en standaardiseer de index**

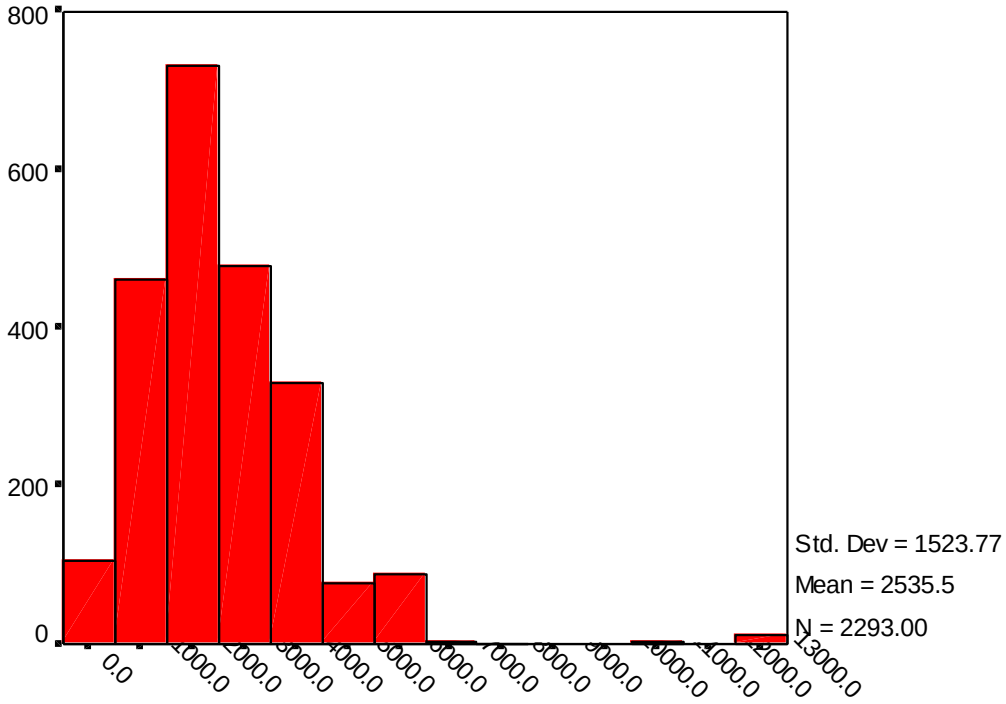
Het is goed om de score die je zo gemaakt hebt nog eens te beschrijven. Dit kan weer via **`desc`** of **`freq`**. Vooral als je veel indicatoren hebt gebruikt, zie je een soort normale verdeling staan. Als je niet gelukkig bent met de ontstane meeteenheid, dan kun je als laatste stap nog standaardiseren.

```
DESC INDEX / SAVE.
```

```
** DIT IS MIJN EERSTE SPSS-SYNTAX.
** DE RESULTATEN ZIJN OPGENOMEN IN DE SYNTAX.

GET FILE='F:\)DATA\ONDERW\STATISTIEK\eenzaam98.sav'.

** Het volgende dient om telkens dezelfde 50% steekproef
** uit de gegevens te trekken.

SET SEED 1232770.
sample .50.

freq country.
recode country (2=1)(1=0)
  into toscane.
var label toscane "land van woonachtigheid".
value labels toscane (1)"yes" (0)"no".
freq toscane.

freq age/stat=all.
recode age (55 thru 59=57)
   (60 thru 64=62)
   (65 thru 69=67)
   (70 thru 74=72)
   (75 thru 79=77)
   (80 thru 84=82)
   (85 thru 89=87)
  into agecat.
var label agecat "leeftijd categorie".
value labels agecat (57) "55-59" (62) "60-64"
 (67) "65-69" (72) "70-74" (77) "75-79" (82) "80-84"
 (87) "85-89".

freq agecat/stat=all.

freq inc.
recode inc (1=375) (2=875) (3=1125) (4=1375) (5=1625)
   (6=1875) (7=2125) 8=2375) (9=2625) (10=2875) (11=3125)
 (12=3375) (13=3625) (14=3875) (15=4125) (16=4375)
 (17=4625) (18=4875) (19=5250) (20=5750) (21=6500)
 (22=8000) (23=11000) (24=13000)    into incmid.
freq incmid/stat=all.

GRAPH
  /HISTOGRAM=incmid .

GRAPH
  /BAR(SIMPLE)=COUNT BY incmid.
```




INCMID




INCMID