

Methodiek voor koppeling van registratiebestanden

Dr. P.H. Polak & drs. F.D. Bijleveld

D-2002-5

Methodiek voor koppeling van registratiebestanden

Documentbeschrijving

Rapportnummer:	D-2002-5
Titel:	Methodiek voor koppeling van registratiebestanden
Auteur(s):	Dr. P.H. Polak & drs. F.D. Bijleveld
Onderzoeksthema:	Weggebruikers: de relatie tussen gedrag, omgeving en ongevallen
Themaleider:	Drs. I.N.L.G. van Schagen
Projectnummer SWOV:	37.301
Trefwoord(en):	Data bank, hospital, accident, injury, data processing, transport mode, classification, statistics, calculation, method, program (computer), Netherlands
Projectinhoud:	Nadere uitleg van principes en uitwerking van de footprint-methode die in een eerder SWOV-onderzoek (Polak, 2001) is gebruikt om een koppeling te maken tussen twee registratiebestanden. Het ging hierbij om het bestand van verkeersslachtoffers zoals door de politie geregistreerd en het bestand van ziekenhuisopnamen zoals door de ziekenhuizen geregistreerd. De koppeling werd gemaakt om de werkelijke omvang te bepalen van de jaarlijkse aantallen verkeersslachtoffers in Nederland die in een ziekenhuis moesten worden opgenomen.
Aantal pagina's:	12 + 11
Prijs:	€ 8,75
Uitgave:	SWOV, Leidschendam, 2002

Stichting Wetenschappelijk Onderzoek Verkeersveiligheid SWOV
Postbus 1090
2260 BB Leidschendam
Telefoon 070-3173333
Telefax 070-3201261

Samenvatting

In een eerder SWOV-onderzoek (Polak, 2001) is een koppeling gemaakt tussen de bestanden van verkeersslachtoffers zoals door de politie geregistreerd en van ziekenhuisopnamen zoals door de ziekenhuizen geregistreerd. Het doel was het bepalen van de werkelijke omvang van de jaarlijkse aantallen verkeersslachtoffers in Nederland die in een ziekenhuis moesten worden opgenomen. Bij de koppeling werden records in beide bestanden gezocht die zoveel mogelijk overeenstemden. Ook werd een nieuwe methode gebruikt om vast te stellen hoeveel gematchte records werkelijk hetzelfde slachtoffer betroffen. Deze zogenaamde footprintmethode maakt gebruik van een variabele die geen rol speelt bij het matchen: in het onderhavige rapport en in het onderzoek uit 2001 is dat de vervoerswijze van het slachtoffer.

Het gebruik van de footprintmethode heeft voor- en nadelen. Wanneer de kleinste-kwadratenmethode wordt gebruikt, treden soms negatieve waarden op. Bij de footprintmethode komen deze niet voor. Een nadeel van de methode, echter, is de noodzaak een geavanceerd rekensysteem als Mathematica te moeten gebruiken.

De nieuwe methode is in het SWOV-rapport uit 2001 slechts summier gerapporteerd. In dit rapport wordt een gedetailleerdere beschrijving van principes en uitwerking van de footprintmethode gegeven.

Abstract

Method for linking registration files

In an earlier SWOV study (Polak, 2001) a linkage was made between the data files of road accident victims (as recorded by the police) and in-patients (as recorded by hospitals). The purpose was to determine the real number of road accident victims annually admitted to hospital in the Netherlands. The linkage used records from both files that were as similar as possible. Also, a new method was used to determine how many matched records actually referred to one and the same victim. This so-called footprint method uses a variable that takes no part in the matching: in the present and previous study, this is the mode of transport of the victim.

Using the footprint method has its advantages and disadvantages. When the least squares method is used, there are sometimes negative values. When the footprint method is used, there are none. However, a disadvantage of the method is the necessity of having to use an advanced calculation system such as Mathematica.

In the 2001 SWOV report by Polak, the new method was only briefly dealt with. This report presents a more detailed description of the principles and execution of the footprint method.

Inhoud

1.	Inleiding	7
2.	De footprintmethode	8
2.1.	Principe	9
2.2.	Uitwerking	9
2.3.	Conclusie	11
	Literatuur	12
	Bijlage Listing computerprogramma	13

1. Inleiding

In het kader van het project *Herijking taakstelling* is in opdracht van het Ministerie van Verkeer & Waterstaat gerapporteerd (Polak, 2001) over een koppeling tussen de bestanden van verkeersslachtoffers zoals door de politie geregistreerd (VOR-bestand) en van ziekenhuisopnamen zoals door de ziekenhuizen (LMR-bestand) geregistreerd. Het doel was het bepalen van de werkelijke omvang van de jaarlijkse aantallen verkeersslachtoffers in Nederland die in een ziekenhuis moesten worden opgenomen.

Bij zo'n koppeling tussen twee bestanden worden records in beide bestanden gezocht die zoveel mogelijk overeenstemmen. Als twee records meer overeenkomsten hebben met elkaar dan een van beide met andere records, worden ze aan elkaar toegewezen, gematcht. In hoofdzaak is daarbij dezelfde werkwijze gevolgd als bij een eerdere koppeling (Polak, 1997), afgezien van enkele verbeteringen. Eén daarvan is een nieuwe methode om vast te stellen hoeveel gematchte records werkelijk hetzelfde slachtoffer betreffen. Bij deze zogenaamde footprintmethode wordt gebruik gemaakt van een variabele die geen rol heeft gespeeld bij het matchen: de vervoerswijze van het slachtoffer. De nieuwe methode heeft twee voordelen. De methode is preciezer en kan met behulp van het speciaal daarvoor ontworpen computerprogramma geautomatiseerd worden uitgevoerd, terwijl de oude methode op handmatige bewerking berustte.

De nieuwe methode maakte geen deel uit van de opdracht tot het hierboven genoemde project en is daarin slechts summier gerapporteerd. In dit rapport wordt, in hoofdstuk 2, een beschrijving van de nieuwe methode gegeven. Een listing van het gebruikte computerprogramma is opgenomen als bijlage.

2. De footprintmethode

Het matchen van records uit het VOR-bestand en het LMR-bestand gebeurt door middel van een afstandsfunctie die aangeeft hoeveel de twee records van elkaar verschillen in de zes koppelvariabelen:

- geboortedatum;
- datum en tijdstip van het ongeval respectievelijk opname;
- geslacht;
- ziekenhuis;
- ernst verwonding;
- type ongeval.

De *afstand* tussen een *record* in het *VOR-bestand* en een *record* in het *LMR-bestand* is groter naarmate de twee records meer van elkaar verschillen, in de waarden die ze hebben voor de *koppelvariabelen*. De *afstand* is nul als alle waarden bekend en aan elkaar gelijk zijn. Bij een *afstand* 100 is de aannemelijkheid dat de records hetzelfde slachtoffer betreffen (terechte *match*) circa 50%. Het is aannemelijk dat onder de (grote) groep best gematchte recordparen slechts enkele ten onrechte gematcht zijn (zie Polak, 1997).

Om nu van de eveneens grote groep met grotere afstand gematchte records te bepalen hoeveel terecht gematcht zijn, is de footprintmethode ontwikkeld. Deze maakt gebruik van een additionele variabele die in beide bestanden voorkomt: de vervoerswijze van het slachtoffer. Deze variabele is in de twee bestanden nogal verschillend gedefinieerd. Bovendien is de vervoerswijze in het ziekenhuisbestand in zo'n 10% van de gevallen als onbekend geregistreerd. Daarom is de vervoerswijze in eerste instantie niet als koppelvariabele gebruikt. De variabele is echter om die reden wel zeer geschikt voor de footprintmethode.

In een tabel waarvan de twee assen de vervoerswijze vormen volgens elk der bestanden, zouden bij perfecte overeenstemming alleen de diagonaalcellen gevuld zijn. In werkelijkheid valt ruim 20% van de records buiten de diagonaal, in een sterk niet random patroon. Een voorbeeld is opgenomen als *Tabel 1*.

1993	Voet	Fiets	Brom	Motor	Auto	Bu/Vr	Overi	Niet g.	Totaal
Voet	589	43	10	1	44	60	1	61	809
Fiets	241	1114	15	2	113	14	2	102	1603
Brom	3	53	1176	46	15	2	64	57	1443
Motor	3	1	67	615	12	0	0	44	742
Auto	152	64	13	4	3186	20	1	485	3925
Bu/Vr	3	0	0	0	18	37	0	3	61
Overi	9	3	3	0	3	0	8	9	35
Totaal	1027	1278	1584	668	3391	133	76	761	8618

Tabel 1. *Footprinttabel Mvtg 1993. Horizontaal de vervoerswijze volgens de LMR, verticaal volgens de VOR, voor de best gematchte records.*

2.1. Principe

De methode om te bepalen hoeveel gematchte recordparen in een groep terecht gematcht zijn gaat ervan uit dat deze groep bestaat uit twee delen: een groep terecht gematchte recordparen en een groep willekeurig gematchte recordparen. Van de eerste groep wordt aangenomen dat hij in alle relevante eigenschappen (bij grote benadering) identiek is aan de eerder genoemde groep die met een afstand van praktisch nul gematcht is. Van de tweede groep wordt aangenomen dat hij random is in die zin dat de celwaarden weinig verschillen met de waarden die volgen uit de randtotalen.

De methode gaat nu in principe als volgt. De tabel van minder goed gematchte recordparen K wordt opgevat als de som van een numerieke factor p maal de footprinttabel F plus een random tabel R . In formule:

$$K = p * F + R$$

zodat p maal het aantal records in F het aantal terecht gematchte records geeft. Omdat zowel F als K uit een eindig aantal records bestaan en opgevat moeten worden als steekproeven uit een universum van mogelijke jaargegevens, dient de methode daar rekening mee te houden. Dit gebeurt door achterliggende tabellen F en K te schatten op basis van het principe van meest aannemelijke schatters ('maximum likelihood'). Hierbij worden op waarden voor de onbekenden zo gekozen dat de kans om bij een nieuw experiment dezelfde waarden voor de footprint te vinden, het grootst is.

2.2. Uitwerking

De footprintmethode werkt met twee frequentietabellen waarin horizontaal een LMR-classificatie naar vervoerswijze staat aangegeven (k -classificaties) en verticaal een VOR-classificatie is aangegeven (m -classificaties). In de cellen van de tabellen is aangegeven hoeveel keer gematchte slachtoffergegevens in de betreffende combinatie zijn voorgekomen. De footprinttabel is de tabel met combinaties waarvan uitgegaan is dat zij terecht gematcht zijn; de tweede tabel is die waarvoor geschat moet worden welk gedeelte terecht gematcht is.

Beide tabellen worden verondersteld het resultaat van een experiment te zijn, zodat de celvullingen dus mede van toeval afhankelijk zijn.

In het volgende wordt met F de footprinttabel bedoeld. Met F_i wordt de i -de rij van tabel F bedoeld, dus de aantallen voor de wijze van vervoer i , oplopend van 1 tot en met n , bijvoorbeeld voetgangers, fietsers, enzovoort. Met K wordt de tabel bedoeld waarvan het niet zeker is dat er correct gematcht is, K_i is hiervan de i -de rij.

Ten slotte stelt θ het geheel aan onbekenden voor, die als vector gebruikt zal worden. Met $P(F | \theta)$ wordt de kans bedoeld dat de tabel F gevonden wordt als tabel van terecht gematchten, gegeven dat θ de juiste waarden bevat. $P(K | \theta)$ stelt de kans voor de niet-zeker-gematchten voor. Gegeven een gevonden F en K moet nu een θ gevonden worden zo, dat $P(F | \theta) * P(K | \theta)$ zo groot mogelijk is of, equivalent $\ln[P(F | \theta)] + \ln[P(K | \theta)]$ zo groot mogelijk is.

$P(\mathbf{F} | \theta)$ kan via zijn rijen ontbonden worden in VOR-vervoerswijzen, zodat

$$P(\mathbf{F} | \theta) = P(\mathbf{F}_1 | \theta, n_1) * \dots * P(\mathbf{F}_m | \theta, n_m) * P(\text{som van } \mathbf{F}_1 = n_1 \text{ en } \dots \text{ en som van } \mathbf{F}_m = n_m).$$

$P(\text{som van } \mathbf{F}_1 = n_1 \text{ en } \dots \text{ en som van } \mathbf{F}_m = n_m)$ is niet afhankelijk van θ . Een vergelijkbare ontbinding kan worden uitgevoerd voor \mathbf{K} .

Het vinden van het maximum voor θ van $\text{Ln}[P(\mathbf{F} | \theta)] + \text{Ln}[P(\mathbf{F} | \theta)]$ is nu gelijkwaardig aan het vinden van het maximum voor θ van $\text{Ln}[P(\mathbf{F}_1 | \theta, n_1)] + \dots + \text{Ln}[P(\mathbf{F}_m | \theta, n_m)] + \text{Ln}[P(\mathbf{K}_1 | \theta, r_1)] + \dots + \text{Ln}[P(\mathbf{K}_m | \theta, r_m)]$, waarbij r_i gelijk is aan de som van de elementen van de rij \mathbf{K}_i .

De vraag is nu hoe $\text{Ln}[P(\mathbf{F}_i | \theta, n_i)]$ eruit ziet. Verondersteld is dat een slachtoffer (bijvoorbeeld een fietser in de VOR-gegevens) een bepaalde kans heeft om als fietser in de LMR terecht te komen. We nemen aan dat deze kans gelijk is aan f_{ij} , $f_{ij} \geq 0$ en de som over de j waarden (per rij, wijze van vervoer volgens VOR) gelijk is aan 1. Dan zijn de aantallen in de rij \mathbf{F}_i multinomiaal verdeeld met parameters n_i en f_{i1}, \dots, f_{ik} (k LMR-vervoerswijzen). $\text{Ln}[P(\mathbf{F}_i | \theta, n_i)]$ is dan, na verwaarlozing van gedeelten die niet van θ afhankelijk zijn, gelijkwaardig aan:

$$\mathbf{F}_{i1} \text{Ln}(f_{i1}) + \dots + \mathbf{F}_{ik} \text{Ln}(f_{ik}).$$

Voor de matrix \mathbf{K} moet nog een extra aanname worden gedaan. Indien een slachtoffer terecht gematcht is, dan gelden dezelfde kansen als voor \mathbf{F} . Indien er niet terecht gematcht is, dan gelden er kansen ongeacht de wijze van vervoer volgens VOR: de kans is g_1 op LMR-vervoerswijze 1, tot en met g_k op LMR-vervoerswijze k .

Bijvoorbeeld: de fietser heeft in \mathbf{K}_i een kans ϕ_i om terecht gematcht te zijn en vervolgens f_{ij} om als LMR-slachtoffercategorie j in de boeken te komen en een kans van $(1-\phi_i)$ om niet terecht gematcht te worden en vervolgens g_j om als LMR-slachtoffercategorie j in de boeken te komen. De kans om in LMR-slachtoffercategorie j in de boeken te komen is dus gelijk aan $\phi_i f_{ij} + (1-\phi_i) g_j$.

De aantallen in rij \mathbf{K}_i zijn ook multinomiaal verdeeld, nu met parameters r_i en $\phi_i f_{i1} + (1-\phi_i) g_1, \dots, \phi_i f_{ik} + (1-\phi_i) g_k$, $\text{Ln}[P(\mathbf{K}_i | \theta, r_i)]$ is, na verwaarlozing van gedeelten die niet van θ afhankelijk zijn, gelijkwaardig aan

$$\mathbf{K}_{i1} \text{Ln}[\phi_i f_{i1} + (1-\phi_i) g_1] + \dots + \mathbf{K}_{ik} \text{Ln}[\phi_i f_{ik} + (1-\phi_i) g_k].$$

Een nog nader te onderzoeken hypothese is of de ϕ_i -waarden gelijk zijn, zodat er één fractie van terecht gematchten bestaat voor alle wijzen van vervoer volgens VOR. Voor het schatten van de waarden van ϕ_i moeten ook de waarden f_{ij} en g_j geschat worden.

Het hier beschreven algoritme is geprogrammeerd in Mathematica en leidt in de meeste gevallen tot een uniek maximum voor de 'likelijkheid'. Er zijn nu alleen puntschattingen voor de parameters geschat. Overwogen kan worden ook betrouwbaarheidsintervallen te schatten.

2.3. **Conclusie**

Het gebruik van de footprintmethode heeft voor- en nadelen. Wanneer de kleinste-kwadratenmethode wordt gebruikt, treden soms negatieve waarden op. Bij de footprintmethode komen deze niet voor. Een nadeel van de methode, echter, is de noodzaak een geavanceerd rekensysteem als Mathematica te moeten gebruiken.

Literatuur

Polak, P.H. (1997). *Registratiegraad van in ziekenhuizen opgenomen verkeersslachtoffers*. R-97-15. SWOV, Leidschendam.

Polak, P.H. (2001). *De aantallen in ziekenhuizen opgenomen verkeersgewonden, 1995-1997; Koppeling van gegevens van de verkeersongevallenregistratie en de registratie van de ziekenhuizen*. R-2000-26. SWOV, Leidschendam.

Bijlage

Listing computerprogramma


```
Needs["Statistics`Master`"];
Needs["Graphics`Master`"];

$DefaultFont = {"Times-Roman", 9};
standardStyles =
  {{AbsoluteThickness[1]}, {AbsoluteDashing[{3, 3, 3, 3}], AbsoluteThickness[1]},
   {AbsoluteDashing[{6, 3, 6, 3}], AbsoluteThickness[1]},
   {AbsoluteDashing[{6, 6, 6, 6}], AbsoluteThickness[1]},
   {AbsoluteDashing[{9, 3, 9, 3}], AbsoluteThickness[1]},
   {AbsoluteThickness[1]}, {AbsoluteDashing[{3, 3, 3, 3}], AbsoluteThickness[1]},
   {AbsoluteDashing[{6, 3, 6, 3}], AbsoluteThickness[1]},
   {AbsoluteDashing[{6, 6, 6, 6}], AbsoluteThickness[1]},
   {AbsoluteDashing[{9, 3, 9, 3}], AbsoluteThickness[1]}};

SetOptions[MultipleListPlot, PlotJoined → True,
  SymbolShape → {None, {AbsolutePointSize[4], Point[#]} &},
  Axes → False, Frame → True, LineStyles :> standardStyles,
  PlotRange → All, FrameTicks → {Automatic, Automatic, None, None},
  LegendSize → {1.2, 1},
  ImageSize → 500, FormatType → OutputForm, GridLines → None,
  LegendPosition → {1.1, -0.4}];

avvwijze[_] := "onb.";
avvwijze[10] = "voet";
avvwijze[20] = "fiets";
avvwijze[30] = "brom";
avvwijze[50] = "motor";
avvwijze[60] = "auto";
avvwijze[80] = "bus/vracht";
avvwijze[200] = "overig";
avvwijze[a_List] := StringJoin[ToString[Infix[Map[avvwijze[#] &, a], "+"]]];

lmrwijze[0] = "voet";
lmrwijze[1] = "fiets";
lmrwijze[2] = "brom";
lmrwijze[3] = "motor";
lmrwijze[4] = "auto";
lmrwijze[7] = "bus/vracht";
lmrwijze[8] = "overige";
lmrwijze[9] = "niet gesp.";

typetext[1] = "ecode 800 - 825";
typetext[2] = "ecode 826 - 829";

kwaltext[0] = "kopkwalb 1,2 en 3";
kwaltext[1] = "kopkwalb 1";
kwaltext[2] = "kopkwalb 2";
kwaltext[3] = "kopkwalb 3";
kwaltext[4] = "kopkwalb 4";
kwaltext[5] = "kopkwalb 5";
kwaltext[6] = "kopkwalb 6";
kwaltext[7] = "kopkwalb 4, 5 en 6";
```

```

kwaltext[a_List] :=
  StringJoin["Kopkwalb = ", ToString[Infix[Map[ToString[#] &, a], "+"]]];
kwaltext[a_] := StringJoin["Kopkwalb = ", ToString[a]];

foottext[a_List] :=
  StringJoin["Footprint = ", ToString[Infix[Map[ToString[#] &, a], "+"]]];
foottext[a_] := StringJoin["Footprint = ", ToString[a]];
foottext[0] := "Footprint = 1,2 en 3";

foottext2[a_List] :=
  StringJoin["Footprint(2) = ", ToString[Infix[Map[ToString[#] &, a], "+"]]];
foottext2[a_] := StringJoin["Footprint(2) = ", ToString[a]];
foottext2[0] := "Footprint(2) = 1,2 en 3";

jaartext[a_List] :=
  StringJoin["Jaar = ", ToString[Infix[Map[ToString[#] &, a], "+"]]];
jaartext[a_] := StringJoin["Jaar = ", ToString[a]];

func[a_, {index_}, n_] :=
  Join[Table[1, {index - 1}], {Cos[a]}, Table[Sin[a], {i, index + 1, n}]]

funcini[a_, {index_}, n_] :=
  Join[Table[1, {index - 1}], {Cos[a]}, Table[(index - 1) / index, {i, index + 1, n}]]

polarvec[a_, n_] := ((Times @@ MapIndexed[func[#1, #2, n] &, Array[a, {n - 1}]])^2)

polarvec[a_] := ((Times @@ MapIndexed[func[#1, #2, Length[a] + 1] &, a])^2)

polarvecini[a_] := MapIndexed[#1 -> N[ArcCos[Sqrt[1 / (Length[a] + 2 - First[#2])]]] &, a]

SetDirectory["g:\\pp\\"];

SetDirectory["FritsHD:PP:"];

SetDirectory::cdir : Cannot set current directory to FritsHD:PP:.

rawvvmdata = ReadList["vvm1.dat", {Number, Number, Number}];

rawvvlmrdata = ReadList["vvlmr1c.dat", {Number, Number, Number, Number, Number}];

rawvvtotdata =
  ReadList["vvtot1.dat", {Number, Number, Number, Number, Number, Number}];

vvmbyyear =
  Map[Transpose[Drop[Transpose[#], 1]] &, Split[rawvvmdata, #1[[1]] == #2[[1]] &]];

vvlmrpcodes = Union[Transpose[rawvvlmrdata][[3]]]
{0, 1, 2, 3, 4, 7, 8, 9}

vvmcodes = Union[Transpose[rawvvmdata][[2]]]
{10, 20, 30, 50, 60, 80, 200}

```

```

jaren = Union[Transpose[rawvvmdata][[1]]]
{1985, 1986, 1992, 1993, 1994, 1995, 1996, 1997}

Clear[vvm];
vvm[___] := 0;
Map[vvm#[[1]], #[[2]] = #[[3]]; &, rawvvmdata];
Map[vvm#[[1]], #[[2]], type_] = #[[3]]; &, rawvvmdata];
vvm[jaren_List, vv_] := Plus @@ Map[vvm[#, vv] &, jaren];

Clear[vvmlmr];
vvmlmr[___] := 0;
Map[vvmlmr#[[1]], #[[2]], #[[3]]] = #[[4]]; &, rawvvmlmrdata];
vvmlmr[jaren_List, type_, vv_] := Plus @@ Map[vvmlmr[#, type, vv] &, jaren];

Clear[vvmtot];
vvmtot[___] := 0;
Map[vvmtot#[[1]], #[[2]], #[[3]], #[[4]], #[[5]]] = #[[6]]; &, rawvvmtotdata];

vvmtot[jaar_, type_, 0, vvm_, vvmlmr_] := vvmtot[jaar, type, 1, vvm, vvmlmr] +
  vvmtot[jaar, type, 2, vvm, vvmlmr] + vvmtot[jaar, type, 3, vvm, vvmlmr];

vvmtot[jaar_, type_, 7, vvm_, vvmlmr_] := vvmtot[jaar, type, 4, vvm, vvmlmr] +
  vvmtot[jaar, type, 5, vvm, vvmlmr] + vvmtot[jaar, type, 6, vvm, vvmlmr];

Clear[comb0, comb];
comb0[jaar_, type_, kwalc_] :=
  Outer[vvmtot[jaar, type, kwalc, #1, #2] &, vvmcodes, vvmlmrcores];
comb0[jaar_List, type_, kwalc_] := Plus @@ Map[comb0[#, type, kwalc] &, jaar];
comb0[jaar_, type_, kwalc_List] := Plus @@ Map[comb0[jaar, type, #] &, kwalc];
comb[jaar_, type_, kwalc_] := comb[jaar, type, kwalc] = comb0[jaar, type, kwalc];

Clear[ncomb];
ncomb[jaar_, type_, kwalc_] := ncomb[jaar, type, kwalc] =
  Map[If[Plus @@ # == 0, Table[1/Length[#], {Length[#]}], #/Plus @@ #] &,
  comb[jaar, type, kwalc]]

foot = comb[1996, 1, {1, 2, 3}];
foot // MatrixForm

```

$$\begin{pmatrix} 551 & 49 & 8 & 1 & 18 & 46 & 0 & 55 \\ 197 & 1092 & 26 & 1 & 61 & 14 & 1 & 86 \\ 38 & 58 & 1315 & 47 & 38 & 6 & 63 & 70 \\ 3 & 1 & 44 & 684 & 12 & 2 & 0 & 30 \\ 137 & 52 & 16 & 17 & 3236 & 22 & 6 & 452 \\ 1 & 1 & 0 & 0 & 11 & 36 & 1 & 2 \\ 0 & 1 & 0 & 0 & 2 & 0 & 7 & 1 \end{pmatrix}$$

```
kwal = comb[1996, 1, 4]; kwal // MatrixForm
```

$$\begin{pmatrix} 35 & 1 & 0 & 0 & 3 & 3 & 0 & 2 \\ 6 & 29 & 1 & 4 & 11 & 1 & 0 & 4 \\ 4 & 2 & 49 & 10 & 9 & 0 & 6 & 5 \\ 0 & 1 & 6 & 20 & 2 & 0 & 0 & 2 \\ 13 & 3 & 13 & 10 & 103 & 3 & 1 & 19 \\ 0 & 0 & 1 & 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Clear[model3]
```

```
model3[jaar_, type_, footc_, kwalc_] := model3[jaar, type, footc, kwalc] =
Module[{}, Print[jaar, " ", type, " ", footc, " ", kwalc];
  foot = comb[jaar, type, footc];
  kwal = comb[jaar, type, kwalc];
  fparmsM = Table[Symbol[StringJoin["f", ToString[i], ToString[j]]],
    {i, 1, Length[vvmcodes]}, {j, 1, Length[vvmlmrcodes] - 1}];
  f = Map[polarvec[#] &, fparmsM];
  fparms = Flatten[fparmsM];
  muparms = Table[Symbol[StringJoin["mu", ToString[j]]],
    {j, 1, Length[vvmlmrcodes] - 1}]; mu = polarvec[muparms];
  mutab = Table[mu, {i, 1, Length[vvmcodes]}];
  llpart1 = Map[Plus @@ # &, foot Log[f]];
  theta = ArcTan[thetaparm] / Pi + 1 / 2;
  llpart2 = Map[Plus @@ # &, kwal Log[theta f + (1 - theta) mutab]];
  llparts = llpart1 + llpart2;
  ll = -Plus @@ Flatten[llparts];
  parms = Join[{thetaparm}, muparms, fparms];
  grad = Map[D[ll, #] &, parms];
  musol = polarvecini[muparms];
  thetasol = {thetaparm → 0.001};
  ll1 = Plus @@ llpart1;
  fcandidates = Flatten[Select[
    MapThread[If[Plus @@ #2 === 0, Null, #1] &, {fparmsM, foot}], # != Null &];
  finitial = Flatten[Select[MapThread[If[Plus @@ #2 === 0, polarvecini[#1], Null] &,
    {fparmsM, foot}], # != Null &];
  eqns = Map[(ee = D[ll1, #]; If[ee === 0, #, ee]) == 0 &, fcandidates];
  fsol = Flatten[{MapThread[FindRoot[#1, {#2, 0.01}, MaxIterations → 1000] &,
    {eqns, fcandidates}], finitial}]; llmu = (ll /. fsol);
  iniprms = Map[{{#[[1]], 1.0 #[[2]]} &, Join[thetasol, musol]];
  tmpsol1 = FindMinimum[llmu,
    Evaluate[Sequence @@ iniprms], MaxIterations → 1000, AccuracyGoal → 12];
  musol = Map[# → (# /. Last[tmpsol1]) &, muparms];
  thetasol = {thetaparm → (thetaparm /. Last[tmpsol1])};
  llf = -Plus @@ (llparts /. musol);
  iniprms = Map[{{#, 1.0 (# /. thetasol /. fsol)} &, Join[{thetaparm}, fparms]];
  tmpsol2 = FindMinimum[llf, Evaluate[Sequence @@ iniprms], MaxIterations → 1000];
  prmssol = Last[tmpsol2];
  fsol = Map[# → (# /. prmssol) &, fparms];
  thetasol = {thetaparm → (thetaparm /. prmssol)};
  iniprms = Map[{{#[[1]], 1.0 #[[2]]} &, Join[prmssol, musol]];
  tmpsol3 = FindMinimum[ll,
    Evaluate[Sequence @@ iniprms], MaxIterations → 1000, AccuracyGoal → 12];
  gradvec = grad /. Last[tmpsol3];
  {tmpsol3, gradvec}];
```

```
llparts
```

```
llparts
```

```
Length[llpart1]
```

```
7
```

```
llpart1 + llpart2
```

```
Length[llpart2]
```

```
7
```

```
model3[1985, 2, {1, 2, 3}, {4, 5, 6}]
```

```
1985 2 {1, 2, 3} {4, 5, 6}
```

```
{ {918.351, {thetaparm → -9.94995, f11 → 0.637958, f12 → 0.334571, f13 → 1.5708,
  f14 → 1.5708, f15 → 1.5708, f16 → 1.5708, f17 → 1.5708, f21 → 1.48186, f22 → 0.126451,
  f23 → 1.5708, f24 → 1.5708, f25 → 1.27614, f26 → 1.5708, f27 → 1.01991,
  f31 → 1.43063, f32 → 0.638028, f33 → 0.487139, f34 → 1.5708, f35 → 1.5708,
  f36 → 1.5708, f37 → 0.784701, f41 → 1.5708, f42 → 1.96605 × 10-17, f43 → 0.,
  f44 → 0., f45 → 0., f46 → 0., f47 → 0., f51 → 1.11051, f52 → 0.971133, f53 → 1.5708,
  f54 → 1.5708, f55 → 0.860186, f56 → 1.15378, f57 → 1.5708, f61 → 1.46487,
  f62 → 1.52096, f63 → 1.56356, f64 → 1.56296, f65 → -0.00471909, f66 → 0.955317,
  f67 → 0.785398, f71 → 1.5708, f72 → 1.5708, f73 → 1.5708, f74 → 1.07807,
  f75 → 1.5708, f76 → 1.5708, f77 → 0.712125, mu1 → 1.73091, mu2 → 0.264222,
  mu3 → 1.14255, mu4 → 1.45817, mu5 → 1.30237, mu6 → 1.56806, mu7 → 1.02343}},
{-0.0193903, -0.302786, 0.586514, -0.0296498, 0.1301, -0.436862, -0.232081,
0.159403, -0.125776, -0.0403538, -8.18747 × 10-10, -3.98902 × 10-9,
-3.98902 × 10-9, -3.98902 × 10-9, -2.68633 × 10-9, 0.329735, 0.269584,
5.61277 × 10-12, -2.30677 × 10-12, 0.00142819, -6.48134 × 10-12, 0.000899942,
-0.318172, 0.0119385, 0.0019926, -5.34205 × 10-10, -3.7976 × 10-9,
-5.34205 × 10-10, 0.00185294, -7.30904 × 10-9, 1.09503 × 10-16, 0., 0., 0.,
0., 0., -0.444368, 0.110294, -1.4877 × 10-10, -1.03304 × 10-11, 0.120818,
-0.124574, -1.66296 × 10-10, 0.0780291, -0.0393012, -0.0129548,
-0.0140275, -0.00844497, 0, 0, -1.29859 × 10-9, -2.3785 × 10-9,
-1.29859 × 10-9, 0.0126268, -7.20418 × 10-9, -7.20418 × 10-9, 0.0763244}}
```

```
Map[model3[#, 2, {1, 2, 3}, {4, 5, 6}] &, jaren];
```

```
1986 2 {1, 2, 3} {4, 5, 6}
```

```
1992 2 {1, 2, 3} {4, 5, 6}
```

```
1993 2 {1, 2, 3} {4, 5, 6}
```

```
1994 2 {1, 2, 3} {4, 5, 6}
```

```
1995 2 {1, 2, 3} {4, 5, 6}
```

```
1996 2 {1, 2, 3} {4, 5, 6}
```

```
FindMinimum::fmmp : Machine precision is insufficient to achieve the requested accuracy or precision.
```

```
1997 2 {1, 2, 3} {4, 5, 6}
```

```
FindMinimum::fmmp : Machine precision is insufficient to achieve the requested accuracy or precision.
```

```
Map[model3[#, 1, {1, 2, 3}, {4, 5, 6}] &, jaren];
```

1985 1 {1, 2, 3} {4, 5, 6}

1986 1 {1, 2, 3} {4, 5, 6}

1992 1 {1, 2, 3} {4, 5, 6}

1993 1 {1, 2, 3} {4, 5, 6}

1994 1 {1, 2, 3} {4, 5, 6}

1995 1 {1, 2, 3} {4, 5, 6}

1996 1 {1, 2, 3} {4, 5, 6}

1997 1 {1, 2, 3} {4, 5, 6}

```
Put[FullDefinition[model3], "model3.m"]
```

```
Max[Abs[gradvec]]
```

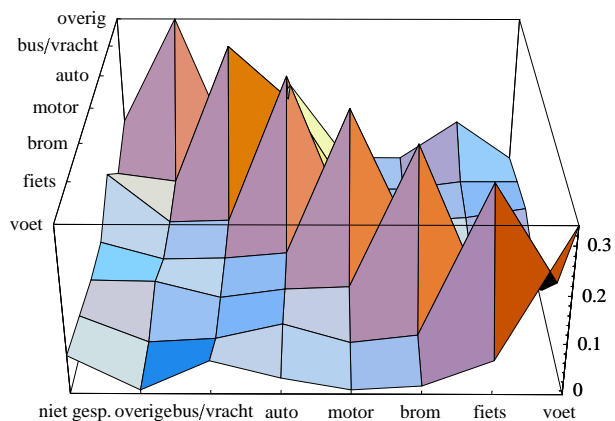
0.349834

```
fest = f /. Last[tmposol3];
```

```
MatrixForm[fest]
```

0.759849	0.0648199	0.0103775	0.00129773	0.0263029	0.0635184	2.72217
0.132848	0.739156	0.0172518	0.000679689	0.0417336	0.00955023	0.00065843
0.0232163	0.0351033	0.802823	0.0292173	0.0231321	0.00354521	0.0402867
0.00375779	0.00138559	0.0583551	0.879934	0.0152899	0.00250462	2.41538
0.0350614	0.0131799	0.00405989	0.00429618	0.821378	0.00566317	0.0015244
0.0186354	0.018636	9.50598×10^{-22}	3.19352×10^{-22}	0.218396	0.68842	0.0186375
1.31566×10^{-22}	0.0912911	1.85916×10^{-22}	2.99955×10^{-23}	0.181784	3.6066×10^{-23}	0.634805

```
ListPlot3D[Transpose[fest],
  Ticks -> {MapIndexed[{First[#2], avvwijze[#1]} &, vvmcodes],
    MapIndexed[{First[#2], lmrwijze[#1]} &, vvmLMRCodes], Automatic},
  ViewPoint -> {-2.949, -0.014, 1.659}]
```



- SurfaceGraphics -

```

Get["model2.m"]

{0.00511521, -0.00124248, -0.00219566, 0.00297985, 0.000899996, 0.00046937,
 0.00241366, 0.0120441, -0.00836824, -0.00765909, 0.00466411, -0.000524383,
 0.000712062, 0.00142823, 0.0140799, -0.00289017, -0.000205446, -0.00010143,
 -0.000714345, -0.00165264, 0.0026781, 0.00446595, -0.0229167, -0.00486204,
 0.00248911, 0.00879489, 0.00110461, 0.00243547, -0.00480043, -0.00146388,
 0.000432722, 0.00556659, -0.0113043, 0.0033924, 0.00254453, 0.00424874,
 0.000347211, -0.00257281, 0.00296086, -0.00471847, 1.1791931204596757×10-10,
 -0.00631821, 0.00231192, 0.00936551, 0.0100144, 0.00332539, 0.00603942,
 -0.00299719, 0.00274423, -0.00370095, 3.67245×10-10, 5.08648×10-10, 8.66014×10-10,
 0.000476227, 0.0047234, -0.000412186, 0.000588809, -1.5281907263957402×10-12,
 3.3907×10-11, 8.18163×10-12, 0.000303012, -0.00174727, -0.000213174}

analyse3[jaar_, type_, footc_, kwalc_] := Module[{},
  foot = comb[jaar, type, footc];
  kwal = comb[jaar, type, kwalc];
  vmarg = Map[Plus @@ # &, kwal];
  dubbel = (theta /. Last[model3[jaar, type, footc, kwalc][[1]]) vmarg;
  {jaar, type, footc, kwalc, Plus @@ Flatten[foot], Plus @@ Flatten[kwal],
   Plus @@ dubbel, 100 (Plus @@ dubbel) / (Plus @@ Flatten[kwal])}]

table3 = Join[Map[analyse3[#, 1, {1, 2, 3}, {4, 5, 6}] &, jaren],
  Map[analyse3[#, 2, {1, 2, 3}, {4, 5, 6}] &, jaren]]

{{1985, 1, {1, 2, 3}, {4, 5, 6}, 10533, 2689, 1054.48, 39.2144},
 {1986, 1, {1, 2, 3}, {4, 5, 6}, 10770, 3271, 1231.57, 37.6512},
 {1992, 1, {1, 2, 3}, {4, 5, 6}, 8734, 2904, 834.815, 28.7471},
 {1993, 1, {1, 2, 3}, {4, 5, 6}, 8618, 2881, 769.126, 26.6965},
 {1994, 1, {1, 2, 3}, {4, 5, 6}, 8851, 2692, 746.706, 27.738},
 {1995, 1, {1, 2, 3}, {4, 5, 6}, 8668, 2870, 808.443, 28.1687},
 {1996, 1, {1, 2, 3}, {4, 5, 6}, 8618, 2700, 759.472, 28.1286},
 {1997, 1, {1, 2, 3}, {4, 5, 6}, 8389, 3045, 816.097, 26.8012},
 {1985, 2, {1, 2, 3}, {4, 5, 6}, 878, 1477, 47.0927, 3.1884},
 {1986, 2, {1, 2, 3}, {4, 5, 6}, 865, 1855, 96.6358, 5.20948},
 {1992, 2, {1, 2, 3}, {4, 5, 6}, 711, 2214, 65.7891, 2.97151},
 {1993, 2, {1, 2, 3}, {4, 5, 6}, 650, 2505, 70.053, 2.79653},
 {1994, 2, {1, 2, 3}, {4, 5, 6}, 691, 2239, 65.6847, 2.93366},
 {1995, 2, {1, 2, 3}, {4, 5, 6}, 715, 2671, 146.975, 5.50263},
 {1996, 2, {1, 2, 3}, {4, 5, 6}, 641, 2573, 82.3378, 3.20007},
 {1997, 2, {1, 2, 3}, {4, 5, 6}, 672, 2785, 73.5178, 2.63978}}

Export["n:\fbij\foot3.dat", table3, "Table"]

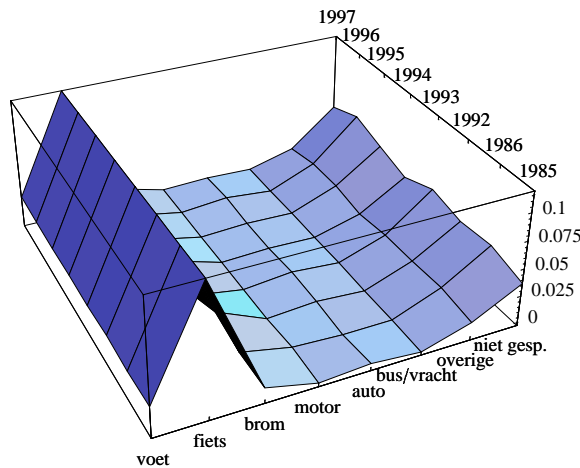
n:\fbij\foot3.dat

```

```
Map[mu /. model3[#, 2, {1, 2, 3}, {4, 5, 6}][[1, 2]] &, jaren]
```

```
{0.0254179, 0.908112, 0.0114629, 0.00069486, 0.00382032,
 3.77182×10-7, 0.0136762, 0.0368153}, {0.024865, 0.894476, 0.0125694,
0.0028436, 0.000580457, 0.000748292, 0.0183406, 0.0455763},
{0.0264761, 0.887802, 0.020572, 0.000465373, 0.00354088, 0.000931179,
0.0171531, 0.0430595}, {0.0246725, 0.893452, 0.00806465,
0.000410685, 0.00488748, 0.00164276, 0.0168382, 0.0500315},
{0.0256104, 0.904437, 0.0140473, 0.000460129, 0.00230064, 0.000920269,
0.0132571, 0.0389673}, {0.0247986, 0.896914, 0.00943587,
4.31553×10-10, 0.00221445, 0.00118862, 0.0180334, 0.0474149},
{0.0255459, 0.885114, 0.00897329, 0.000400965, 0.00452593,
0.000401684, 0.0181532, 0.0568849}, {0.026031, 0.899545, 0.012228,
0.000369015, 0.00214354, 0.000737986, 0.0122607, 0.046685}}
```

```
ListPlot3D[Map[mu /. model3[#, 2, {1, 2, 3}, {4, 5, 6}][[1, 2]] &, jaren],
  Ticks -> {MapIndexed[{First[#2], lmrwijze[#1]} &, vvmLMRCodes],
  MapIndexed[{First[#2], ToString[#]} &, jaren], Automatic},
  ViewPoint -> {-1.230, -2.605, 1.775}]
```

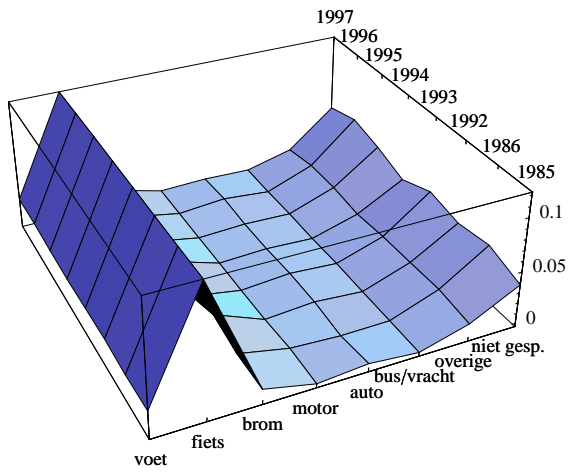


- SurfaceGraphics -

```
Map[mu /. model2[#, 2, {1, 2, 3}, {4, 5, 6}][[1, 2]] &, jaren] // TableForm
```

0.023351	0.907851	0.0113133	0.000708999	0.00464755	0.000704762
0.0230486	0.893581	0.0123937	0.00283982	0.000568005	0.000842282
0.0247825	0.889285	0.0193621	0.000465835	0.00427746	0.000932281
0.023864	0.894222	0.00832028	0.000410253	0.00451831	0.00163852
0.0236928	0.904151	0.0151046	0.00050523	0.00252259	0.00100676
0.0244225	0.896156	0.00870957	1.23286×10 ⁻¹³	0.0023453	0.0011873
0.0256176	0.885713	0.00889658	0.000401336	0.00456184	0.000401306
0.0257053	0.898313	0.012082	0.000384049	0.00214631	0.00076805


```
ListPlot3D[Map[mu /. model2[#, 2, {1, 2, 3}, {4, 5, 6}][[1, 2]] &, jaren],
  Ticks -> {MapIndexed[{First[#2], lmrwijze[#1]} &, vvmlmrcodes],
    MapIndexed[{First[#2], ToString[#]} &, jaren], Automatic},
  ViewPoint -> {-1.230, -2.605, 1.775}]
```



- SurfaceGraphics -