rijksuniversiteit groningen

faculteit Wiskunde en Natuurwetenschappen

Horror-winter op komst?

Voorspelling van de gemiddelde maandtemperatuur van januari 2012 op basis van lineaire regressie

Bacheloronderzoek Wiskunde

Oktober 2011 - Februari 2012

Student: L.R. Jansen

Eerste Begeleider: E.C. Wit

Tweede Begeleider: F.W. Wubs

Bron plaatje op de voorkant: http://boeken.blogo.nl/2011/10/31/de-winter-staat-voor-de-deur-enjoy-de-winter/, bezocht op 7 februari 2012

Inhoudsopgave

1 Inleiding **5**

2 Theorie **7**
2.1 Het lineaire regressiemodel . . . 7
2.2 Schattingen voor β . . . 8
2.2.1 Least Squares Estimator . . . 8
2.2.2 Maximum Likelihood Estimator (MLE) . . . 9
2.2.3 Bayesiaanse benadering: prior en posterior . . . 12
2.3 Voorspelling . . . 15
2.4 Regression diagnostics . . . 16
2.4.1 De hat-matrix . . . 16
2.4.2 Bayesiaanse benadering: predictive densities . . . 17
2.5 Model selectie . . . 18
2.5.1 Mallows' C_p . . . 18
2.5.2 AIC . . . 20
2.5.3 Cross-validation . . . 22
2.5.4 De p-waarde en een plot van de residuen . . . 24
2.5.5 Bayesiaanse benadering: Bayes factor . . . 28
2.6 Gemengde lineaire modellen . . . 29
2.7 Ridge regressie . . . 30

3 Interessante artikelen **31**
3.1 Linear Short-term Climate Predictive Skill in the Northern Hemisphere [4] . . . 31
3.2 Predicting Summertime Caribbean Pressure in Early April [5] . . . 33

4 Weersvoorspelling **36**
4.1 Factoren die van invloed zijn op het weer . . . 36
4.2 Het model . . . 38

5 Standaard lineaire regressie en selectie met AIC **39**
5.1 Regressie over de temperaturen . . . 39
5.2 Regressie over de SST . . . 40
5.2.1 El Niño . . . 40
5.2.2 Noordzee, Oostzee en Atlantische Oceaan . . . 43
5.3 Arctische Oscillatie . . . 45
5.4 Modellen samenvoegen . . . 45
5.5 LOOCV . . . 49
5.6 Voorspellingen voor de gemiddelde temperatuur in januari 2012 . . . 50
5.7 Conclusie; welk model is het beste op basis van standaard lineaire regressie? . . . 51
5.8 Wat zegt het eindmodel? . . . 51

6 Verbeteringen in het model door andere regressie? **53**
6.1 Mixed models . . . 53
6.2 Ridge regressie . . . 54

7 Conclusie **57**

8 Nawoord **57**

9 Literatuur **58**

10 Bijlagen **60**
10.1 Bijlage 1: R-script van de voorbeelden uit hoofdstuk 2 60
10.2 Bijlage 2: R-script van de berekeningen uit hoofdstuk 5 en 6 60

1 Inleiding

Afgelopen zomer was koud en nat. Alle campings zaten vol teleurgestelde vakantiegangers, de zwembaden waren veel leger dan normaal en de ijscokar kon wel in de schuur blijven. Elke dag werd het weerbericht door vele mensen bekeken. Iedereen hoopte op een zonnetje en een graad of 25. Of in ieder geval niet van dat druilerige sombere weer.
Het weer is altijd al een goed gespreksonderwerp, maar afgelopen zomer hielden we er maar niet over op. Bekenden, onbekenden, met iedereen werd het verlangen naar lekker zomers weer gedeeld. Helaas kwam het gevraagde warme weer niet echt. Maar dat is geen reden om het gespreksonderwerp "weer"links te laten liggen. In tegendeel, iedereen begon over de naderende winter. Deze kon wel eens erg koud worden, gezien het tegenvallende weer in de zomer. Er gingen zelfs speculaties rond over een horrorwinter. Iedereen wilde er wel wat over zeggen, maar de meteorologen kwamen er nog niet helemaal uit. Het onderwerp haalde zelfs de landelijke kranten, zoals dit artikeltje uit de telegraaf van 27 oktober 2011 [19]:

"**'Winter des doods' is niet te voorspellen**
DE BILT - Drie maanden Siberische kou, een dik pak sneeuw en verraderlijke ijzel. De winter slaat begin december in West-Europa genadeloos toe, zo voorspellen meerdere weerbureaus in Duitsland, Groot-Brittannië en de VS op basis van seizoensverwachtingen. Maar het KNMI heeft zijn twijfels over dergelijke langetermijnvoorspellingen. "De voorspellingshorizon voor het weer is 10 dagen", zegt Cees Molenaars van het KNMI.
Het KNMI volgt "met belangstelling"de langetermijnverwachtingen en voorspellingen over hete zomers en koude winters. Er bestaan heel veel weermodellen, maar er is onvoldoende wetenschappelijke grond om ze echt serieus te nemen, aldus Molenaars. "In het voorjaar zeiden sommigen dat we een lange hete zomer zouden krijgen. We kennen het resultaat."West-Europa is qua weer afhankelijk van de Atlantische Oceaan. "Elke depressie die voor de kust van Canada wordt geboren, zie je aankomen."Verder voorspellen blijft een hachelijke zaak. "Het weer is zeer complex."
Toch buitelen weerbureaus over elkaar heen met langetermijnverwachtingen. Het Duitse weerbureau Donnerwetter voorspelt een 'horrorwinter' en Britse meteorologen spreken nu al over een 'winter des doods'. De kans dat we dit jaar een witte kerst beleven, zou 40 procent zijn en de temperaturen kunnen in januari 's nachts tot min 25 graden dalen.
Ook Meteo Consult verwacht een strenge winter, gebaseerd op de inzichten van de World Climate Service. Die voorspelde vorig jaar een koude winter en dit jaar een koele en natte zomer. Beide verwachtingen kwamen grotendeels uit, meldt Meteo Consult op zijn website. Ook tv-weerman Piet Paulusma schrijft op zijn website: "Ik ga uit van een relatief koude winter 2011-2012. Allereerst verwacht ik dat de winter vroeg begint, in november wordt het al koud en krijgen we winterweer."
De winterse voorspellingen van de World Climate Service zijn onder meer gebaseerd op de temperaturen in de zomer rond de poolcirkel en de te verwachten luchtstromen in de winter. Berekeningen geven aan dat er de komende maanden veel polaire en dus koude lucht naar West-Europa zal stromen. "Dat zijn slechts 2 factoren die van invloed kunnen zijn", relativeert Cees Molenaars van het KNMI. "Het weer is van zo veel factoren afhankelijk.

Dit alles leek mij voldoende reden om mij er ook eens mee te gaan bemoeien. Ik ga in dit onderzoek met behulp van lineaire regressie kijken of ik een voorspelling kan doen voor januari 2012. Er zal gekeken worden naar de gemiddelde maandtemperatuur. Zodoende hoop ik dan uiteindelijk te kunnen zeggen of de voorspelde horrorwinter (wat zich uit in een erg lage gemiddelde maandtemperatuur) ook daadwerkelijk komt.
Allereerst ga ik mij verdiepen in de theorie achter de lineaire regressie en model selectie. Met

behulp van een aantal simpele voorbeeldjes leer ik de theorie toe te passen en met R te werken. Vervolgens ga ik wat artikeltjes bestuderen om eens te kijken hoe de lineaire regressie kan worden toegepast. Daarna ga ik zelf aan de slag. Met datasets van het KNMI en van de NOAA begin ik simpel en vervolgens worden de simpele modelletjes gecombineerd. Dit resulteert in een overall-model voor de gemiddelde temperatuur in januari op basis van standaard lineaire regressie. Een aantal waarden die aangeven hoe goed de fit is, worden bepaald en vervolgens wordt er gekeken of het model nog beter gemaakt kan worden. Dat kan bijvoorbeeld door het meenemen van random-effecten of door het invoeren van een soort "straffactor". Ook wordt de standaarddeviatie van de voorspelling bekeken. Uiteindelijk hoop ik mij dan ook in de discussie over de horrorwinter te kunnen mengen.

2 Theorie

2.1 Het lineaire regressiemodel

Lineaire regressie wordt gebruikt bij het beschrijven van een verdeling van een random variabele Y die varieert met een andere variabele of een andere set van variabelen x [1]. Deze variabelen kunnen bijvoorbeeld gemeten worden en dan kan er een verband gezocht worden. Er wordt gesproken van enkelvoudige lineaire regressie als er in de vergelijking één afhankelijke variabele en ook één onafhankelijke variabele is. Een voorbeeld hiervan is de oppervlaktespanning van vloeistoffen als functie van de temperatuur [8]. De vergelijking van de oppervlaktespanning γ (in N/m) als functie van de temperatuur T (in graden Celsius) kan weergegeven worden als $\gamma = \beta_0 + \beta_1 T$, waarbij β_0 en β_1 constanten zijn. Het doel van lineaire regressie is nu om de waarden van deze β_0 en β_1 te bepalen uit een aantal waarden van γ en T.
Voor meervoudige lineaire regressie zijn er meerdere onafhankelijke variabelen en daarmee dus meerdere β's die bepaald moeten worden. Een goed voorbeeld hiervan is de opbrengst (yield) van een chemische reactie als functie van de temperatuur en de druk [7]. Deze functie heeft dan de vorm: yield $= \beta_0 + \beta_1 T + \beta_2 P$. Vaak wordt er echter een extra term meegenomen, namelijk de interactie tussen P en T, wat weergegeven wordt als T*P (en in R wordt het weergegeven als T:P). Om te begrijpen wat interactie inhoudt, is het goed om te kijken naar de helling van de yield als functie van de temperatuur [7]. Zonder interactie is deze helling gelijk aan $\frac{\partial yield}{\partial T} = \beta_0$. Dit betekent dat de helling van de yield constant is. Als de interactie meegenomen wordt, wordt de functie yield $= \beta_0 + \beta_1 T + \beta_2 P + \beta_3 P * T$. De afgeleide hiervan naar de temperatuur is gelijk aan $\beta_1 + \beta_3 P$. Nu is de helling van de yield dus niet langer constant, maar afhankelijk van de druk. Dus de interactie zorgt ervoor dat de helling van de yield versus temperatuur kan veranderen als de druk verandert. Dit geldt natuurlijk ook voor de helling van de yield versus de druk, door de interactie kan deze nu veranderen als functie van de temperatuur. In drie dimensies betekent dit dat zonder interactie de oplossing een plat vlak wordt, terwijl met interactie dit vlak gebogen kan worden. Een gebogen vlak is vaak beter in staat om de meetgegevens te beschrijven [7].
De gegevens die gemeten worden, kunnen als vectoren of als matrices weergegeven worden. Bij enkelvoudige lineaire regressie vormen de meetwaarden een vector, als er meervoudige regressie gedaan wordt, vormen de gemeten waarden voor de onafhankelijke variabelen een matrix. In matrixnotatie wordt het lineaire regressiemodel weergegeven als

$$y = X\beta + \epsilon \tag{1}$$

Hierbij is β de vector die gevormd wordt door de waarden van de β_i's die horen bij de onafhankelijke variabelen x_i. De onafhankelijke variabelen x_i vormen de matrix X. Deze matrix kan een kolom enen bevatten als er een as-afsnede is die meegenomen moet worden, zoals bij het voorbeeld van de yield het geval is. Hier bestaat de matrix X uit een kolom met enen en daarna een kolom met meetwaarden voor de druk en temperatuur. De vector y wordt dan gegeven door de meetwaarden voor de yield. In het voorbeeld van de oppervlaktespanning bestaat de vector y uit meetwaarden voor γ, de matrix X bestaat uit een kolom enen en daarnaast een kolom met meetwaarden voor de temperatuur. De vector β wordt dan gevormd door β_0 en β_1.
Er dienen een aantal fundamentele aannames gedaan te worden met betrekking tot 1 [1]:

- X is een niet-stochastische $n \times p$ matrix met $p < n$
- de matrix X heeft rang p
- de elementen van de $n \times 1$ vector y zijn waarneembare random variabelen

- de elementen van de $n \times 1$ vector ϵ zijn niet-waarneembare random variabelen zodanig dat $E(\epsilon) = 0$ en $Cov(\epsilon) = \sigma^2 I_n$ met $\sigma^2 > 0$, in het kort wordt dit weergegeven door ϵ $(0, \sigma^2 I_n)$

Als bovendien onderstaande aannames kloppen, is er te spreken van een zogenaamde classical linear regression method [1].

- de vector ϵ is n-variaat normaal verdeeld
- de ongelijkheid $p \geq 3$ is geldig
- de gelijkheid $X^T X = I_p$ is geldig
- de parametervector β voldoet aan Rβ = r voor een gegeven $m \times p$ matrix R met volledige rijruimte en een gegeven $m \times 1$ vector r

Deze functie kan ook worden weergegeven als een normale verdeling, namelijk:

$$y \sim N(X\beta, \sigma^2 I). \tag{2}$$

2.2 Schattingen voor β

2.2.1 Least Squares Estimator

Allereerst zal er gekeken worden naar de least squares estimator, welke gegeven wordt door

$$\hat{\beta} = \beta_{OLS} = (X^T X)^{-1} X^T y. \tag{3}$$

Hierbij staat het subscript OLS voor de ordinary least squares oplossing. De least squares variance estimator wordt gegeven door

$$\hat{\sigma}^2 = \frac{1}{n-p}(y - X\hat{\beta})^T (y - X\hat{\beta}). \tag{4}$$

Hoe zijn deze waarden nou berekend, wat is het idee erachter? Als een systeem y = X β_* oplosbaar is naar β_*, dan is er een oplossing β_0 waarvoor geldt $\|y - X\beta_0\|^2 = 0$. Als het systeem y=Xβ_* echter niet oplosbaar is, dan kan er een vector $\hat{\beta}$ bepaald worden waarvoor geldt $\|y - X\hat{\beta}\|^2 \leq \|y - X\beta_*\|^2$ voor elke vector $\beta_* \in \mathbb{R}^p$. De vector $\hat{\beta}$ wordt de least squares solution genoemd, want bovenstaande uitdrukking kan ook geschreven worden als $\sum_{i=1}^{n}(y - X\hat{\beta})_i^2 \leq \sum_{i=1}^{n}(y - X\beta_*)_i^2$.
Neem nu $\epsilon_{*i} := (y - X\beta_*)_i$ voor het i'de residu van de oplossing β_*, dan is de sum of squared residuals (SSR) minimaal als $\beta_* = \hat{\beta}$, dus dat betekent dat $\hat{\beta}$ de kleinste SSR heeft.
De oplossing voor $\hat{\beta}$ kan nu verkregen worden door de functie $\|y - X\beta_*\|^2$ te differentiëren naar β_* en daarna nul te stellen.

$$\frac{\partial}{\partial \beta_*}\|y - X\beta_*\|^2 = 2X^T X\beta_* - 2X^T y$$

Als dit op nul gesteld wordt, levert dat op $\hat{\beta} = (X^T X)^{-1} X^T y$. Deze vector wordt de ordinary least squares estimator (OLS) voor β genoemd [2].
De least squares variance estimator is gebaseerd op de vector die gevormd wordt door de residuen: $\hat{\epsilon} = y - X\hat{\beta}$. Deze variance estimator is unbiased voor σ^2. De matrix X wordt de modelmatrix genoemd. Als y de geoberveerde uitkomsten zijn, dan is de matrix (y,X) de observatiematrix.

2.2.2 Maximum Likelihood Estimator (MLE)

Het is ook mogelijk om de waarde van $\hat{\beta}$ te vinden uit de Maximum Likelihood Estimator (MLE). $\hat{\beta}$ is de MLE van β gegeven y. De likelihood wordt gegeven door

$$p(y|X,\beta,\sigma^2) \propto \exp(-\frac{1}{2\sigma^2}\text{SSR}(\beta)) = \exp(-\frac{1}{2\sigma^2}[y^T y - 2\beta^T X^T y + \beta^T X^T X\beta]). \tag{5}$$

Om deze te kunnen maximaliseren is het vaak makkelijker om de logaritme van de functie te nemen en deze te maximaliseren. Het nemen van de logaritme verandert niets aan de ligging van het maximum [6]. Dit kan gedaan worden en levert dus op

$$\ln(\exp(-\frac{1}{2\sigma^2}\text{SSR}(\beta))) = -\frac{1}{2\sigma^2}\text{SSR}(\beta).$$

Nu moet deze functie gemaximaliseerd worden ten opzichte van β. In de functie komt σ^2 voor, wat positief is. Bovendien is $\text{SSR}(\beta)$ ook positief, dus de gehele functie zal negatief zijn. Om deze negatieve functie te maximaliseren, zal een zo klein mogelijk negatieve waarde gevonden moeten worden. Dat kan door $SSR(\beta)$ te minimaliseren. Dit betekent dat er weer uitgerekend dient te worden wanneer $SSR(\beta)$ minimaal is. Dit is hierboven ook gedaan en toen is er gevonden dat dit de waarde $\hat{\beta} = (X^T X)^{-1} X^T y$ oplevert. Het is nu dus zo dat de schatting van $\hat{\beta}$ op basis van de minimalisatie van de som van de residuen (dus de least squares schatting) en de schatting op basis van de Maximum Likelihood Estimator de waarde $\hat{\beta} = (X^T X)^{-1} X^T y$ opleveren.
Voor het voorbeeld van de oppervlaktespanning kan de oplossing voor $\hat{\beta}$ bepaald worden. Voor de oppervlaktespanning en de temperatuur zijn de meetwaarden uit tabel 1 bekend [8].

temperatuur (graden Celsius)	oppervlaktespanning ($*10^{-3}$ N/m)
40	41,6
57	39,5
69,7	38,3
70	37,8
80	36,6
100	35,0
100	34,4
110	33,2
120	32,3
140	30,4
160	27,9
180	25,7
200	23,3

Tabel 1: meetwaarden voor temperatuur en oppervlaktespanning [8]

Nu kan dus de matrix X gemaakt worden door een kolom enen en een kolom met de waarden

van de temperatuur naast elkaar te zetten:

$$\begin{bmatrix} 1 & 40 \\ 1 & 57 \\ 1 & 69,7 \\ 1 & 70 \\ 1 & 80 \\ 1 & 100 \\ 1 & 100 \\ 1 & 110 \\ 1 & 120 \\ 1 & 140 \\ 1 & 160 \\ 1 & 180 \\ 1 & 200 \end{bmatrix}$$

De vector y bestaat uit de waarden van de oppervlaktespanning;

$$y = (41.6, 39.5, 38.3, 37.8, 37.2, 36.6, 35, 34.4, 33.2, 32.3, 30.4, 27.9, 25.7, 23.3).$$

Nu kan de least squares oplossing (die gelijk is aan de oplossing van de MLE) berekend worden door $\hat{\beta} = (X^T X)^{-1} X^T y$. Dit kan gedaan worden in het rekenprogramma R door de matrix X en de vector y aan te maken en dan de berekening van de least squares in te voeren, namelijk:

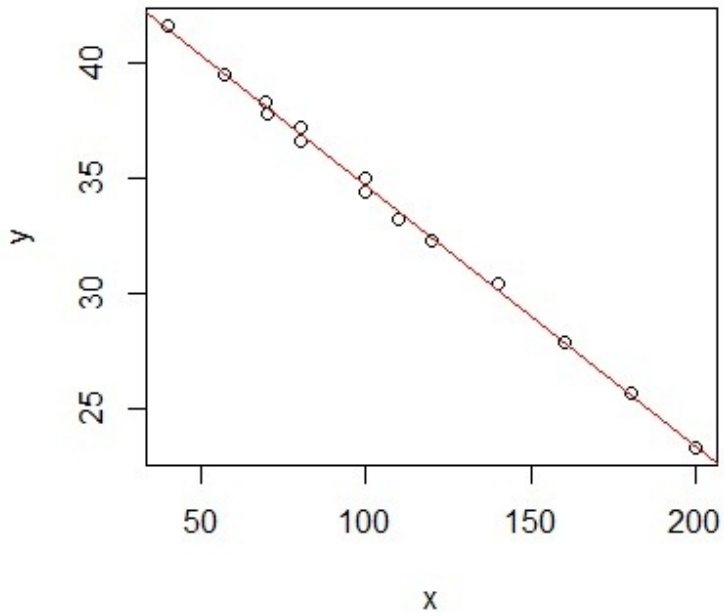
$$\text{data} = c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,40,57,69.7,70,80,80,100,100,110,120,140,160,180,200)$$

$$\text{X=matrix(data,nrow=14,ncol=2)}$$

$$y = c(41.6, 39.5, 38.3, 37.8, 37.2, 36.6, 35, 34.4, 33.2, 32.3, 30.4, 27.9, 25.7, 23.3)$$

$$\text{leastsquares} = \text{solve}(t(X)\% * \%X)\% * \%t(X)\% * \%y$$

Dit geeft als uitkomst voor de leastsquares-vector (45.9629122, -0.1130157), wat betekent dat $\beta_0 = 45.9629122$ en $\beta_1 = -0.1130157$, dus de vergelijking die gevonden wordt met de least squares-methode is $\gamma = \beta_0 + \beta_1 T = 45.9629122 - 0.1130157\, T$.
Het kan ook sneller. R heeft namelijk een ingebouwde functie die de oplossing op basis van de least squares-benadering berekent. Dit is de lm-functie en deze functie fit een lineair model voor gegeven data. Hiervoor is alleen een vector met de waarden van de oppervlaktespanning (vector y) nodig en een vector met de waarden van de temperatuur (vector x). De vectoren kunnen in R worden aangemaakt en dan kan via lm(y∼x) de oplossing worden berekend. Dit is dezelfde oplossing als net met de handmatige berekening is verkregen. Hieronder (figuur 1) staat een plotje van de punten en de daarbij gevonden least squares oplossing (de rode lijn).



Figuur 1: Plot van de meetwaarden en least squares fit

Voor meervoudige lineaire regressie is deze ingebouwde functie extra handig, omdat het ook eenvoudig de interactie kan meenemen. Voor het voorbeeld van de yield kunnen de vectoren P voor de druk, T voor de temperatuur en Y voor de yield worden gemaakt door simpelweg een vector te maken van de betreffende meetwaarden uit tabel 2 [7].

temperatuur (graden Celsius)	druk (bar)	yield (%)
50	1	68
50	1	65
50	5.5	81
50	5.5	82
50	10	57
50	10	65
125	1	52
125	1	53
125	5.5	82
125	5.5	85
125	5.5	80
125	5.5	83
125	10	76
125	10	76
200	1	28
200	1	27
200	5.5	70
200	5.5	68
200	10	80
200	10	84

Tabel 2: meetwaarden voor temperatuur, druk en yield [7]

Nu kan in R de least squares fit bepaald worden. Dit kan door in te voeren lm(Y∼T+P+T:P), wat betekent dat er een fit bepaald wordt voor de yield als functie van de temperatuur, druk en de interactie tussen temperatuur en druk. Dit geeft voor de as-afsnede een waarde van 92.35926, dus $\beta_0 = 92.35926$. Verder wordt er gevonden dat $\beta_1 = -0.31222$, $\beta_2 = -2.87037$, $\beta_3 = 0.04444$, dus de vergelijking die nu gevonden wordt, is

$$\text{yield } = \beta_0 + \beta_1 T + \beta_2 P + \beta_3 P * T$$

$$= 92.35926 - 0.31222\, T - 2.87037\, P + 0.04444\, P * T.$$

2.2.3 Bayesiaanse benadering: prior en posterior

Het is ook mogelijk om de lineaire regressie vanuit Bayesiaans oogpunt te benaderen. Vanuit dit oogpunt moet er een prior worden gevonden voor β. De posterior hiervan zal dan de voorspelling van β zijn. Het is hier nodig om een geconjugeerde prior te definiëren. Dat kan door te kijken naar de rol van β in de exponent in de verdeling. Deze lijkt erg op de rol van y en de verdeling van y is multivariaat normaal. Dit suggereert dat een multivariaat normale verdeling voor β een geconjugeerde prior is. Dit kan gecontroleerd worden door aan te nemen dat $\beta \sim$ multivariaat normaal (β_0, Σ_0) [2]. Dan moet gelden

$$p(\beta|y, X, \sigma^2) \propto p(y|X, \beta, \sigma^2) \times p(\beta)$$

$$\propto \exp[-\frac{1}{2}(-2\beta^T X^T y/\sigma^2 + \beta^T X^T X\beta/\sigma^2) - \frac{1}{2}(-2\beta^T \Sigma_0^{-1}\beta_0 + \beta^T \Sigma_0^{-1}\beta)]$$

$$= \exp[\beta^T(\Sigma_0^{-1}\beta_0 + X^T y/\sigma^2) - \frac{1}{2}\beta^T(\Sigma_0^{-1} + X^T X/\sigma^2)\beta]$$

Dit betekent dus dat voor β een multivariaat normale verdeling met

$$\text{Var}[\beta|y, X, \sigma^2] = (\Sigma_0^{-1} + X^T X/\sigma^2)^{-1} \tag{6}$$

$$E[\beta|y, X, \sigma^2] = (\Sigma_0^{-1} + X^T X/\sigma^2)^{-1}(\Sigma_0^{-1}\beta_0 + X^T y/\sigma^2) \tag{7}$$

een geconjugeerde prior is. Als nu de elementen van de precisiematrix (Σ_0^{-1}) klein zijn, dan zal de verwachtingswaarde E$[\beta|y, X, \sigma^2]$ ongeveer gelijk zijn aan $(X^T X)^{-1}X^T y$, wat precies de least squares oplossing is. Als de elementen in de precisiematrix echter juist groot zijn, zal de verwachtingswaarde gelijk zijn aan β_0, wat de priorverwachting is.
De semi-geconjugeerde prior voor σ^2 heeft een inverse-gamma verdeling. Als $\gamma = 1/\sigma^2$ als maat wordt genomen, is dat makkelijk te zien als $\gamma \sim \text{gamma}(\nu_0/2, \nu_0\sigma_0^2/2)$.

$$\begin{aligned} p(\gamma|y, X, \beta) &\propto p(\gamma)p(y|X, \beta, \gamma) \\ &\propto [\gamma^{\nu_0/2-1}\exp(-\gamma\nu_0\sigma_0^2/2)] \times [\gamma^{n/2}\exp(-\gamma \times SSR(\beta)/2)] \\ &= \gamma^{(\nu_0+n)/2-1}\exp(-\gamma[\nu_0\sigma_0^2 + SSR(\beta)/2)] \end{aligned}$$

Hierin is SSR(β) de som van de kwadraten van de residuen. Bovenstaande verdeling is een gammaverdeling, waardoor nu aangetoond is dat σ^2 een inverse-gammaverdeling heeft, namelijk $\{\sigma^2|y, X, \beta\} \sim$ inverse-gamma $([v_0+n]/2, [v_0\sigma_0^2+SSR(\beta)]/2)$. Hiervan kan eenvoudig een Gibbs-sample gemaakt worden, wat dan dus eigenlijk de posterior van β bepaalt:

- update β
  - bereken $V = Var[\beta|y, X, \sigma^2]$ en $m = E[\beta|y, X, \sigma^2]$
  - sample $\beta^{(s+1)} \sim$ multivariaat normaal (m, V)
- update σ^2
  - bereken SSR$(\beta^{(s+1)})$
  - sample $\sigma^{2(s+1)} \sim$ inverse-gamma $([v_0 + n]/2, [v_0\sigma_0^2 + SSR(\beta)]/2)$

Door dit algoritme wordt de posterior bepaald en daarmee wordt dus de voorspeller van β bepaald.

Unit information prior
Het kan voorkomen dat er geen geschikte prior bekend is voor β. In zo'n geval kan ervoor gekozen worden om het dan maar bij een least squares benadering te laten. Echter, er kan ook voor gekozen worden om een prior met andere criteria te nemen. Allereerst kan er dan voor gekozen worden om een weinig-informatieve prior te nemen. Dit representeert het feit dat er begonnen is met niet zo veel kennis over de populatie die bestudeerd wordt. De eerste mogelijkheid hierin is een unit information prior [2]. Dit is een prior die net zo veel informatie bezit als bij één meting wordt verkregen. Bijvoorbeeld de precisie is gelijk aan de inverse-variantie en wordt voor één meting gegeven door $(X^T X)/\sigma^2$. Als dit uitgebreid wordt naar n metingen, wordt er vanuit gegaan dat deze waarde $1/n$ keer zo groot is voor een enkele meting, dus dan wordt de precisie van de unit information prior gegeven door $(X^T X)/(n\sigma^2)$. Ook is er voorgesteld door Kass en Westerman (1995) om $\beta_0 = \beta_{OLS}$ te nemen, dus om de prior te centreren om de least squares oplossing [2]. Een kanttekening die bij deze prior geplaatst dient te worden, is het feit dat het misschien niet meer een echte prior mag heten, omdat er gebruik gemaakt wordt van informatie van y om de prior te construeren.
g-prior

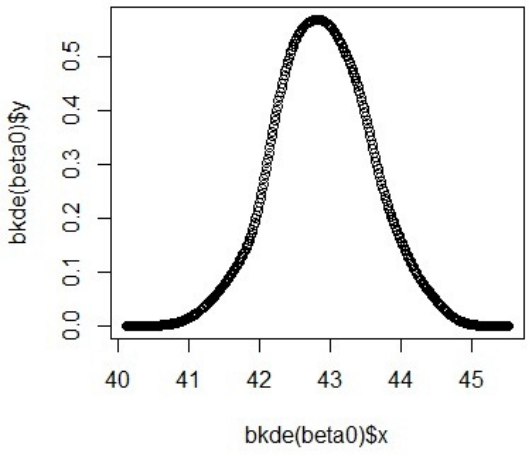
Een andere manier om toch een prior te definiëren is gebaseerd op het idee dat een prior invariant zou moeten zijn aan veranderingen in de schaal van de regressors (dat wil zeggen dat een regressie met voorspellers in maanden gelijk moet zijn aan de regressie met voorspellers in jaren). Om dit te kunnen bewerkstelligen zou een regressie met een set van regressors X en een set van regressors $\tilde{X} = XH$ hetzelfde moeten zijn. Dat kan worden bereikt door te kiezen $\beta_0 = 0$ en $\Sigma_0 = k(X^T X)^{-1}$, voor een positieve waarde van k. Een populaire specificatie voor k is gerelateerd aan de variantie van de fout, σ^2, zodat $k = g\sigma^2$ voor een positieve waarde van g. Dit definieert de zogenaamde g-prior (Zellner, 1986) [2]. Met deze invariante g-prior blijft de voorwaardelijke verdeling van β gegeven (y, X, σ^2) multivariaat normaal verdeeld, maar 6 en 7 worden gereduceerd tot

$$Var[\beta|y, X, \sigma^2] = [X^T X/(g\sigma^2) + X^T X/\sigma^2]^{-1} = \frac{g}{g+1}\sigma^2(X^T X)^{-1} \tag{8}$$

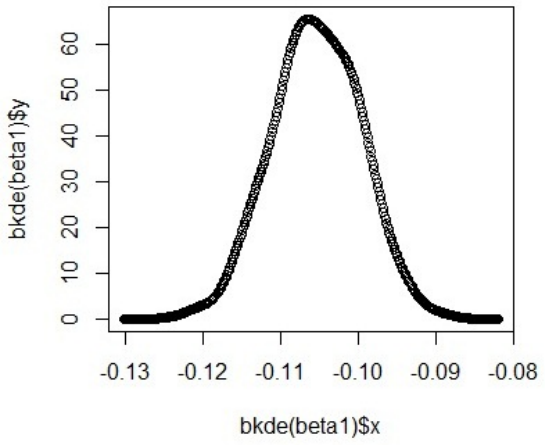
en

$$E[\beta|y, X, \sigma^2] = [X^T X/(g\sigma^2) + X^T X/\sigma^2]^{-1} X^T y/\sigma^2 = \frac{g}{g+1}(X^T X)^{-1} X^T y \tag{9}$$

Het benaderen van de posterior verdeling is nu ook versimpeld; met bovenstaande prior heeft $p(\sigma^2|y, X)$ een inverse-gamma verdeling en dus kan (σ^2, β) direct gesampled worden uit de posterior door eerst te samplen uit $p(\sigma^2|y, X)$ en daarna uit $p(\beta|\sigma^2, y, X)$.
Met behulp van de g-prior kan ook voor het voorbeeld van de oppervlaktespanning van de vloeistoffen een waarde voor β worden bepaald. Dit is al gedaan met de least squares benadering, maar levert een Bayesiaanse benadering ongeveer hetzelfde op of is dit heel anders?
Het idee van de Bayesiaanse benadering is dat er een prior wordt vastgelegd waaruit dan een posterior bepaald wordt. Als er niet een duidelijke prior te bepalen is, kan er gebruik gemaakt worden van de g-prior. Voor de oppervlaktespanning kan bijvoorbeeld een g-prior gebruikt worden met $g = n$ (het aantal rijen van X), $\nu_0 = 2$ en $\sigma_0^2 = 1$. Dit zijn vrij standaard beginwaarden en nu kan er worden gekeken of ook dan een redelijke waarde voor β gevonden kan worden. Hiertoe wordt een R-script gebruikt dat in de bijlage te vinden is. Er wordt een posterior bepaald door 5000 random waarden van β te samplen. Deze 5000 waarden vormen de posterior voor β. Een plot van de posterior voor β_0 en β_1 is hieronder weergegeven (figuur 2 respectievelijk 3).



Figuur 2: plot van de posterior van β_0



Figuur 3: plot van de posterior van β_1

De least squares oplossing voor β was (45.9629122, -0.1130157). Het is te zien dat deze waarden in de plotjes ook inderdaad op de plaatsen met de grootste kansdichtheid liggen. De least squares oplossing en de oplossing op de Bayesiaanse wijze leveren dus een redelijk eenduidig antwoord op.

2.3 Voorspelling

Als er met het lineaire regressiemodel een voorspelling gedaan moet worden, zal dit altijd weergegeven worden als

$$y = X_{voorspelling}\beta + \epsilon. \tag{10}$$

Hierbij is ϵ normaal verdeeld met gemiddelde 0 en variantie σ^2. De waarde van ϵ is niet bekend, dit is de fout. Daarnaast is $X_{voorspelling}$ de matrix met alle bekende waarden van de onafhankelijke variabelen die gebruikt worden voor de voorspelling (dus in het geval van de weersvoorspelling in 2012 zullen dit de waarden uit 2011 zijn) en β bestaat uit de regressie-coëfficiënten die gevonden worden bij dit model.
Het is van een voorspelling vaak handig om een variantie te kunnen schatten, om zo uiteindelijk ook wat te kunnen zeggen over de nauwkeurigheid van de voorspelling y. Dat betekent dat $\mathrm{Var}(X_{voorspelling}\beta + \epsilon)$ geschat moet worden. Dit kan ook geschreven worden als

$$\mathrm{Var}(X_{voorspelling}\beta) + \mathrm{Var}(\epsilon) + 2\mathrm{Cov}(X_{voorspelling}\beta, \epsilon).$$

Omdat ϵ onafhankelijk is van $X_{voorspelling}\beta$, is de covariantie gelijk aan 0. Verder is de variantie van ϵ gelijk aan σ^2 en $X_{voorspelling}$ is een constante matrix, waardoor nu volgt

$$\mathrm{Var}(X_{voorspelling}\beta) + \mathrm{Var}(\epsilon) + 2\mathrm{Cov}(X_{voorspelling}\beta, \epsilon) = X_{voorspelling}\mathrm{Var}(\beta)X^T_{voorspelling} + \sigma^2.$$

Als nu voor β de maximum likelihood estimator wordt genomen, volgt dat dit geschreven kan worden als

$$\sigma^2 X_{voorspelling}(X^T X)^{-1} X^T_{voorspelling} + \sigma^2.$$

Hierin is $X_{voorspelling}$ de matrix die bestaat uit de waarden van de onafhankelijke variabelen die gebruikt worden voor de voorspelling en X is de matrix met de waarden van de onafhankelijke

variabelen die gebruikt zijn voor de bepaling van β. Dit alles levert dus op:

$$\text{Var}(y) = \sigma^2(1 + X_{voorspelling}(X^T X)^{-1} X^T_{voorspelling}). \tag{11}$$

2.4 Regression diagnostics

2.4.1 De hat-matrix

Als er gekeken wordt naar het lineaire regressiemodel 1, dan kunnen de individuele observaties weergegeven worden met $(y_i, x^T_{[i]})$, waarbij $x^T_{[i]}$ de i'de rij van de matrix X voorstelt. De data kunnen nu één of meerdere observaties bevatten die het resultaat meer beïnvloeden dan andere datapunten dat doen. Het is nu zaak om uit te vinden welke punten dat zijn.
Normaal gesproken zal een kleine verandering in y_i geen grote veranderingen in $\hat{y}_i$ te weeg brengen, want $\hat{y}_i = x^T_{[i]}\hat{\beta} = x^T_{[i]}(X^T X)^{-1} X^T$ hangt af van de hele observatiematrix (y,X). Echter, als $x^T_{[i]}$ ver weg ligt van de rest van de data, zal de invloed veel groter zijn. Neem $\frac{\partial \hat{y}_i}{\partial y_i}$ als een maat voor de invloed van $x^T_{[i]}$. Nu is het bekend vanuit de vergelijking $\hat{y} = Py$ met $P = X(X^T X)^{-1} X^T$, dat

$$\frac{\partial \hat{y}_i}{\partial y_i} = p_{ii} \tag{12}$$

waarbij p_{ii} gegeven is door $x_i^T (X^T X)^{-1} x_i$. Deze waarde is het i'de hoofddiagonaalelement van de hat-matrix P [1]. Deze matrix heet de hat-matrix omdat het als het ware het dakje op de y zet. Het zorgt ervoor dat er een voorspelling uitkomt. De waarde van p_{ii} wordt ook wel de i'de hat-waarde genoemd. Het is bekend dat $0 \leq_L P \leq_L I_n$, waaruit volgt dat $0 \leq p_{ii} \leq 1$. Als het model een as-afsnede heeft, dan wordt dit $\frac{1}{n} \leq p_{ii} \leq 1$. In het lineaire regressiemodel $y_i = x_i^T \beta + \epsilon_i$, is de i'de observatie $(y_i, x^T_{[i]})$ een high-leverage punt (een punt met veel invloed) als de waarde van p_{ii} groot is vergeleken met de rest van de hat values van de observaties in de data.
Voor het voorbeeld van de oppervlaktespanning kan de hat-matrix P berekend worden door $P = X(X^T X)^{-1} X^T$. Dit levert de onderstaande matrix op (figuur 4).

0.23	0.19	0.16	0.16	0.14	0.14	0.09	0.09	0.07	0.04	0.	-0.05	-0.1	-0.14
0.19	0.16	0.14	0.14	0.12	0.12	0.08	0.08	0.07	0.05	0.02	-0.02	-0.05	-0.09
0.16	0.14	0.12	0.12	0.11	0.11	0.08	0.08	0.07	0.06	0.03	0.	-0.02	-0.05
0.16	0.14	0.12	0.12	0.11	0.11	0.08	0.08	0.07	0.06	0.03	0.	-0.02	-0.05
0.14	0.12	0.11	0.11	0.1	0.1	0.08	0.08	0.07	0.06	0.04	0.02	0.	-0.02
0.14	0.12	0.11	0.11	0.1	0.1	0.08	0.08	0.07	0.06	0.04	0.02	0.	-0.02
0.09	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.06	0.06	0.05	0.05
0.09	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.07	0.06	0.06	0.05	0.05
0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.08	0.08
0.04	0.05	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.09	0.09	0.1	0.11
0.	0.02	0.03	0.03	0.04	0.04	0.06	0.06	0.07	0.09	0.11	0.13	0.15	0.17
-0.05	-0.02	0.	0.	0.02	0.02	0.06	0.06	0.08	0.09	0.13	0.16	0.2	0.24
-0.1	-0.05	-0.02	-0.02	0.	0.	0.05	0.05	0.08	0.1	0.15	0.2	0.25	0.3
-0.14	-0.09	-0.05	-0.05	-0.02	-0.02	0.05	0.05	0.08	0.11	0.17	0.24	0.3	0.36

Figuur 4: De hat-matrix voor het voorbeeld van de yield.

Hieruit blijkt dat het eerste, dertiende en veertiende diagonaalelementen groot zijn in verhouding met de rest. Dat betekent dat de eerste, dertiende en veertiende meting high-leverage punten zijn voor de oppervlaktespanning.

2.4.2 Bayesiaanse benadering: predictive densities

Bij de Bayesiaanse benadering bepaalt men een prior en daarna een posterior. De bedoeling is dan dat de posterior beter bij de data past dan de prior dat deed. Maar wat nou als de prior en de posterior de data totaal niet op de juiste manier weergeven? Normaal gesproken komt men daar niet meteen achter, want men bepaalt alleen maar de posterior, zonder eerst te bekijken of de prior wel voldoet. Hoe kan dit probleem worden verholpen, met andere woorden, hoe kan er een beetje getest worden of de prior en posterior wel bij de data passen?
Een manier om dat te doen is met behulp van de predictive distributions. In deze verdelingen zijn alle bekende grootheden conditioneel vastgelegd en alle onbekende grootheden zijn eruit geïntegreerd. Als de verdeling niet conditioneel is op de meetwaarden, dan is er te spreken van een prior predictive distribution. Als er ook geconditioneerd is op de meetwaarden, dan is er te spreken van een posterior predictive distribution. Nu kan men, terwijl men de data heeft, proberen te kijken of er een predictive distribution te vinden is en deze kan dan vergeleken worden met de data zelf. Komen ze een beetje overeen of zitten er erg grote verschillen tussen? Als de verschillen groot zijn, moet men zich afvragen of de prior en posterior wel zo goed bij de data passen.
Dit kan bijvoorbeeld gedaan worden als er een prior is voor θ die gamma(θ,a,b) verdeeld is en als de data poisson(θ) verdeeld is. Met behulp van Baye's formule kan de posterior worden bepaald:

$$\begin{aligned} p(\theta|y_1,..,y_n) &= p(\theta) \times p(y_1,..,y_n|\theta)/p(y_1,..,y_n) \\ &= \theta^{a-1}e^{-b\theta} \times \theta^{\Sigma y_i}e^{-n\theta} \times c(y_1,..,y_n,a,b) \\ &= \theta^{a+\Sigma y_i-1}e^{-(b+n)\theta} \times c(y_1,..,y_n,a,b) \end{aligned}$$

Dit is een gamma-verdeling, dus de posterior is gamma$(a+\Sigma_{i=1}^n Y_i, b+n)$ verdeeld. De posterior predictive distribution kan nu bepaald worden door op de bekende waarden te conditioneren en de onbekende waarden uit te integreren.

$$\begin{aligned} p(\tilde{y}|y_1,...,y_n) &= \int_0^\infty p(\tilde{y}|y_1,...,y_n)p(\theta|y_1,...,y_n)\mathrm{d}\theta \\ &= \int p(\tilde{y}|\theta)p(\theta|y_1,...,y_n)\mathrm{d}\theta \\ &= \int \mathrm{dpois}(\tilde{y},\theta)\mathrm{dgamma}(\theta, a+\sum y_i, b+n)\mathrm{d}\theta \\ &\int \{\frac{1}{\tilde{y}!}\theta^{\tilde{y}}e^{-\theta}\}\{\frac{(b+n)^{a+\sum y_i}}{\Gamma(a+\sum y_i)}\theta^{a+\sum y_i-1}e^{-(b+n)\theta}\}\mathrm{d}\theta \\ &\frac{(b+n)^{a+\sum y_i}}{\Gamma(\tilde{y}+1)\Gamma(a+\sum y_i)}\int_0^\infty \theta^{a+\sum y_i+\tilde{y}-1}e^{-(b+n+1)\theta}\mathrm{d}\theta \end{aligned}$$

Deze integraal ziet er ingewikkeld uit, maar omdat bekend is dat een gamma-verdeling integreert tot 1:

$$1 = \int_0^\infty \frac{b^a}{\Gamma(a)}\theta^{a-1}e^{-b\theta}\mathrm{d}\theta$$

(voor $a, b > 0$), volgt dat

$$\int_0^\infty \theta^{a-1}e^{-b\theta}\mathrm{d}\theta = \frac{\Gamma(a)}{b^a}.$$

Als nu $a + \sum y_i + \tilde{y}$ in plaats van a wordt genomen en $b + n + 1$ in plaats van b, dan volgt

$$\int_0^\infty \theta^{a+\sum y_i+\tilde{y}-1} e^{-(b+n+1)\theta} \mathrm{d}\theta = \frac{\Gamma(a + \sum y_i + \tilde{y})}{(b+n+1)^{a+\sum y_i+\tilde{y}}}$$

Na enige algebraïsche versimpeling volgt uiteindelijk dat

$$p(\tilde{y}|y_1, .., y_n) = \frac{\Gamma(a + \sum y_i + \tilde{y})}{\Gamma(\tilde{y}+1)\Gamma(a+\sum y_i)} (\frac{b+n}{b+n+1})^{a+\sum y_i} (\frac{1}{b+n+1})^{\tilde{y}}.$$

Dit is een negatief binomiale verdeling met parameters $(a + \sum y_i, b + n)$ [2], met onderstaande verwachtingswaarde en variantie.

$$\mathrm{E}[\tilde{Y}|y_1, .., y_n] = \frac{a + \sum y_i}{b+n} = \mathrm{E}[\theta|y_1, ..., y_n]$$

$$\mathrm{Var}[\tilde{Y}|y_1, ...y_n] = \frac{a + \sum y_i}{b+n} \frac{b+n+1}{b+n} = \mathrm{Var}[\theta|y_1, ..., y_n] \times (b+n+1) = \mathrm{E}[\theta|_1, ..y_n] \times \frac{b+n+1}{b+n}$$

Dus als er extra punten aan dit model toegevoegd worden, is de voorspelling dat dit een negatief binomiale verdeling zal krijgen.
Om te controleren of de modellen voldoen, is het soms handig om wat data achter de hand te houden en dan de predictive distributions te berekenen. Als deze erg veel verschillen van de echte data die achtergehouden is, dan is het misschien wel de moeite waard om een ander model voor de prior te kiezen.

2.5 Model selectie

Stel dat er al bepaald is welke afhankelijke variabele y uitgedrukt moet worden in onafhankelijke variabelen uit de set $x_1, x_2, ..x_s$ door middel van een lineaire regressie. Welke variabelen uit deze set moeten nou eigenlijk gebruikt worden? Natuurlijk moeten dat alleen de belangrijke onafhankelijke variabelen zijn, zodat er een relatief klein aantal onafhankelijke variabelen zijn die de afhankelijke variabele voorspellen, maar toch met een zekere mate van precisie. In dit hoofdstuk zijn manieren opgenomen om te bepalen hoe "goedëen model is.

2.5.1 Mallows' C_p

Bekijk een lineair regressiemodel die gegeven wordt door $y = X\beta + \epsilon$ met $X = (X_1, X_2)$, waarin X_1 een $n \times p$-matrix is en X_2 een $n \times q$ matrix, zodat

$$y = X_1\beta_1 + X_2\beta_2 + \epsilon, \quad \epsilon \sim (0, \sigma^2 I_n),$$

met $\beta = (\beta_1^T, \beta_2^T)^T$. De voorspelling voor y in het gereduceerde model

$$y = X_1\beta_1 + u, \quad u \sim (0, \sigma^2 I_n)$$

kan gekozen worden als $\hat{y} = X_1\beta_1$ met $\beta_1 = (X_1^T X_1)^T y$. De voorspelling van y kan ook gezien worden als de benadering van $E(y) = X\beta$, waardoor het verwachte ongewogen gekwadrateerde foutverlies van $\hat{y}$ gegeven wordt door

$$\Gamma_p = E[(X_1\beta_1 - X\beta)^T (X_1\beta_1 - X\beta)].$$

Het gereduceerde model $y = X_1\beta_1 + u$ kan gezien worden als een "goed"model als Γ_p klein is. Noem nu $P_1 = X_1(X_1^T X_1)^{-1}X_1^T$, dan volgt

$$\Gamma_p = E(y^T P_1 y) - 2\beta^T X^T P_1 X\beta + \beta^T X^T X\beta$$

met

$$E(y^T P_1 y) = \sigma^2 \mathrm{tr}(P_1) + \beta^T X^T P_1 X\beta.$$

Verder is $\mathrm{tr}(P_1) = p$ en dus kan dit geschreven worden als

$$\Gamma_p = \sigma^2 p + \beta^T X^T M_1 X\beta, \quad M_1 = I_n - P_1.$$

Verder is $E(y^T M_1 y) = \sigma^2 \mathrm{tr}(M_1) + \beta^T X^T M_1 X\beta$ met $\mathrm{tr}(M_1) = n - p$, dus nu volgt

$$\Gamma_p = \sigma^2(2p - n) + E(y^T M_1 y).$$

Omdat Γ_p afhangt van de onbekenden β en σ^2, is nu nog niet te zeggen wanneer Γ_p klein is. Door nu Γ_p te benaderen met

$$\hat{\Gamma}_p = \hat{\sigma}^2(2p - n) + RSS_p,$$

met $\hat{\sigma}^2$ een benadering voor σ^2 en $RRS_p = y^T M_1 y$ is de residual sum of squares in het gereduceerde model. De herschaalde waarde

$$C_p = (2p - n) + \frac{RRS_p}{\hat{\sigma}^2}$$

is bekend als Mallows' C_p [1] voor het model $y = X_1\beta_1 + u$, waarbij de benadering voor σ^2 meestal gegeven wordt door

$$\hat{\sigma}^2 = \frac{1}{n - p - q}(y - X\hat{\beta})^T(y - X\hat{\beta}), \quad \hat{\beta} = (X^T X)^{-1}X^T y.$$

Als verschillende modellen vergeleken worden, is degene met de kleinste C_p "het beste".

Voor- en achterwaartse selectie
Bij stapsgewijze regressiemethodes wordt er stapsgewijs een variabele toegevoegd of juist weggehaald en dan wordt het zo verkregen model bekeken. Het model dat de "besteüitkomst heeft (bijvoorbeeld op basis van Mallows' C_p), wordt gekozen. Een algoritme voor achterwaartse selectie op basis van Mallows' C_p wordt als volgt gegeven:

- *Stap 1:* Bekijk het volledige regressiemodel met alle onafhankelijke variabelen die er zijn. Bereken de least squares variance estimate $\hat{\sigma}^2$ voor σ^2 van dit model. Bereken vervolgens Mallows' C_p (welke in dit geval gelijk zal zijn aan p).
- *Stap 2:* Verwijder één variabele en bereken C_p van dit gereduceerde model. Herhaal deze procedure voor alle onafhankelijke variabelen in het model en kies dan het model met de kleinste C_p.
- *Stap 3:* Ga verder met het model wat in stap 2 gekozen is. Bereken hiervan niet opnieuw de waarde van $\hat{\sigma}^2$, maar doe verder wel hetzelfde als in stap 2. Ga zo door totdat het nieuwe model waarmee begonnen wordt, niet meer verandert.

Deze methode kan ook op een voorwaartse manier worden uitgevoerd. Dan is het model waarmee begonnen wordt gelijk aan het intercept van het lineaire regressiemodel. Echter, bij deze methode valt de benadering $\hat{\sigma}^2$ samen met de variantie van het sample, wat een redelijk zwakke benadering is voor σ^2 als er meerdere (niet-constante) regressors zijn. Voorwaartse en achterwaartse selectie hoeven niet op hetzelfde model uit te komen.

2.5.2 AIC

AIC is ontworpen door Akaike en heet dan ook Akaike's Information Criterion. Het kan net als Mallows C_p gebruikt worden om onafhankelijke variabelen te selecteren uit een verzameling [1]. R en S-PLUS maken ook gebruik van dit criterium. Het wordt gegeven door

$$AIC = -2 \times \text{ maximized log-likelihood } + 2 \times \text{ (aantal parameters)} \tag{13}$$

Voor het lineaire regressie-model met de aannames die gemaakt zijn, wordt de likelihood gegeven door

$$L(\beta, \sigma^2; y) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp(-\frac{(y - X\beta)^T(y - X\beta)}{2\sigma^2}) \tag{14}$$

De log-likelihood is dan gegeven door

$$L^*(\beta, \sigma^2; y) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{(y - X\beta)^T(y - X\beta)}{2\sigma^2} \tag{15}$$

Deze functie bereikt zijn maximum voor β en σ^2 voor $\beta = \hat{\beta}$ en $\sigma^2 = \frac{n-p}{n}\hat{\sigma}^2$, met $\hat{\beta}$ en $\hat{\sigma^2}$ de ordinary least squares estimator en ordinary least squares variance estimator respectievelijk. Dan volgt

$$AIC = -2L^*(\hat{\beta}, \frac{n-p}{n}\hat{\sigma^2}; y) + 2p = n\ln(RSS/n) + 2p + n(\ln(2\pi) + 1) \tag{16}$$

Waarbij $RSS = y^T My$ met $M = I^n - X(X^TX)^{-1}X^T$.
De term die erbij opgeteld wordt $(n(\ln(2\pi) + 1))$ is alleen afhankelijk van de waarde van n. Deze blijft constant als er modellen met verschillende parameters, maar met dezelfde waarnemingen y getest worden. Daardoor is deze term irrelevant voor de vergelijking van de verschillende AIC-waarden. R rekent dan ook in de functie stepAIC (de functie die de achterwaartse selectie op basis van AIC uitvoert) met een aangepaste AIC-waarde, namelijk AIC_p [1], die gegeven wordt door

$$AIC_p = n\ln(RSS_p/n) + 2p \tag{17}$$

De waarde van AIC kan vanzelfsprekend ook worden gebruikt in voor- en achterwaartse modelselectiemethodes, net als Mallows' C_p hierin gebruikt wordt.
De AIC zoals hierboven beschreven is werkt bij modellen met weinig meetwaarden niet zo goed [11]. De modellen met meer parameters zullen dan altijd verkozen worden boven de andere. Als het aantal parameters ongeveer 30% bedraagt van de totale sample size, is de kans groot dat met AIC niet het beste model wordt geselecteerd. Als $p + q$ (het totaal aantal parameters) klein is, is gewone AIC goed genoeg, maar anders moet er een correctie-term worden aangebracht [11]. Hurvich en Tsai (1989) hebben deze gecorrigeerde versie aan de man gebracht [11]. De AIC met een correctie-term werd AICc genoemd, waarbij de laatste c staat voor corrected. De AICc is (in twee dimensies, met het totaal aantal parameters $p + q$) gedefinieerd als

$$AICc(p, q) = -2\log[\text{likelihood}(p, q)] + 2(p + q + 1)\frac{n}{n - p - q - 2} \tag{18}$$

Hierbij wordt eigenlijk een penalty gegeven die $2(p + q + 1)$ is en die wordt vermenigvuldigd met een correctiefactor, $\frac{n}{n-p-q-2}$. Akaike heeft uitgevonden dat de penalty twee keer het aantal parameters moest zijn (zie de definitie van de AIC hierboven). Er zijn $p + q + 1$ parameters (namelijk $p + q$ parameters in het model zelf en de innovation variance ϵ). Bij de gewone AIC maakt het niets uit of deze +1 meegenomen wordt of niet (dat is hierboven niet gedaan), omdat

dat toch alleen maar een constante scheelt. Bij de AICc is het wel van belang om de +1 mee te nemen, want er wordt vermenigvuldigd met de correctiefactor waardoor de AICc's niet meer op een constante zullen verschillen.
De keuze van de penalty is niet zomaar willekeurig. Het is namelijk zo dat het de bedoeling is om de werkelijkheid met een model weer te geven. Dit model heeft maar een beperkt aantal parameters, in het echte model zijn het er misschien wel oneindig. Er wordt met de AIC en de AICc eigenlijk een afstand berekend tussen het geschatte model en het werkelijke, ondanks dat er geen exacte vergelijking voor de werkelijke te vinden is. De penalty-term in de AICc is zo dat deze afstand altijd zonder bias kan worden bepaald. De AIC wordt voor kleine samples biased. Daarom zal er vaak voor de AICc gekozen worden. Dit kost niet veel extra ingewikkeld rekenwerk, maar kan wel de nadelen van AIC ondervangen. Het is aangetoond dat AIC en AICc de best mogelijke modellen opleveren als er aangenomen wordt dat de werkelijkheid oneindig complex is [11].
Als $p+q$ klein is in vergelijking met n, zullen AIC en AICc ongeveer gelijk zijn, omdat de correctiefactor dan ongeveer gelijk is aan 1. Als $p+q$ groter wordt, zal de penalty-term van de AICc meer invloed krijgen dan in de AIC het geval is. Daardoor zal de AICc bij modellen met minder meetwaarden en bij modellen waar het aantal parameters dichter in de buurt ligt van de sample size, eerder een model kiezen met minder parameters dan AIC zou doen. Voor grotere samples zullen de AICc en de AIC ongeveer hetzelfde antwoord geven.
AICc kan natuurlijk ook eenvoudiger geschreven worden voor rekenprogramma's zoals bijvoorbeeld R. Deze programma's werken niet met de likelihood, maar met de residual sum of squares. Net als voor de gewone AIC is gedaan, kan ook in de AICc de likelihood naar de SSR geschreven worden. Dit levert dan op

$$\text{AICc} = N\ln(\frac{SSR}{N}) + 2(p+q+1)\frac{N}{N-p-q-2} \tag{19}$$

als er geen constante term in het model zit en

$$\text{AICc} = N\ln(\frac{SSR}{N}) + 2(p+q+2)\frac{N}{N-p-q-3} \tag{20}$$

als er een constante term in het model zit. In één dimensie is dit natuurlijk

$$N\ln(\frac{SSR}{N}) + 2(p+1)\frac{N}{N-p-2}$$

of

$$N\ln(\frac{SSR}{N}) + 2(p+2)\frac{N}{N-p-3},$$

wat inderdaad gelijk is aan de AIC in één dimensie op de correctie-term na.
Het is erg veel werk om zelf handmatig alle modellen te proberen en dan daarna de AIC te berekenen. Gelukkig kan R dit ook, namelijk met de functie stepAIC. Hierbij is het in te geven welke variabelen gefit moeten worden en dan wordt er een model gemaakt waarvan de AIC bepaald wordt. Daarna wordt er een variabele weggehaald (degene met de grootste p-waarde, dit volgt in de volgende sectie) en dan wordt de AIC nogmaals bepaald. Als de AIC nu lager is, er nogmaals een variabele weggehaald en wordt de AIC opnieuw bepaald. Dit gaat door totdat er een zo laag mogelijke AIC gevonden is. Voor bijvoorbeeld de yield, kan de stepAIC-functie gebruikt worden. Dan wordt er ingevoerd fit=lm(Y $\sim$ P+T+P:T) gevolgd door stepAIC(fit), waarna de functie zal starten met het model met de waarden van de druk, temperatuur en de interactie hiertussen. Hierna zal worden gekeken of er variabelen weggelaten kunnen worden

om een lagere AIC te krijgen. In dit voorbeeld wordt er uitvoer verkregen waarin in eerste instantie een AIC van 97,881 gevonden is. Dit is voor het model waarin zowel P als T als de interactie hiertussen meegenomen is. Daarna wordt de interactie verwijderd en dan wordt een AIC verkregen van 109,801. Dit is duidelijk een verslechtering, dus het gefitte model moet ook de interactie meenemen en dus bestaat het beste model uit de onafhankelijke variabelen P en T en de interactie hiertussen. Een opmerking bij het gebruik van stepAIC is dat deze functie gebruik maakt van de AIC_p (zoals hierboven is uitgelegd), waardoor de AIC die verkregen wordt met stepAIC anders is dan wanneer in R gewoon de AIC van een model opgevraagd wordt. De AIC die vergelijkbaar is met ieder ander model kan opgevraagd worden door in te voeren AIC(..), met op de puntjes het model waarvan de AIC wordt gevraagd. Voor het voorbeeld van de yield wordt er een AIC verkregen van 156,64 respectievelijk 168,60.

2.5.3 Cross-validation

In statistiek en datamining is het vaak een kunst om een model uit een dataset te halen, bijvoorbeeld een regressiemodel of een classificatie. Een probleem wat vaak naar voren komt bij dit soort modellen is dat de modellen goede voorspellingen vertonen voor de data waarop ze zijn gefit, maar dat ze juist slechte voorspellingen vertonen voor toekomstige, onbekende data. Cross-validation is een manier om te bekijken hoe goed de modellen functioneren bij het voorspellen van onbekende data. Het idee van cross-validation is ontstaan in de dertiger jaren. In een paper van S. Larson wordt er een verzameling van data gebruikt voor de regressie en een andere verzameling wordt gebruikt voor de voorspelling [3]. Dit eerste idee van cross-validation werd uitgewerkt door Mosteller en Turkey en vele anderen tot het hedendaagse beeld van cross-validation.
Cross-validation is een statistische methode waarbij verschillende algoritmes worden vergeleken door de beschikbare data in tweeën te delen. Zo wordt er een verzameling gemaakt waarmee een model wordt bepaald, de zogenaamde learning of training set. Daarnaast wordt er een verzameling gemaakt waarmee het model wordt gevalideerd, de zogenaamde validating set. Bij cross-validation wordt de samenstelling van de training set en de validating set voortdurend doorgewisseld, waardoor uiteindelijk alle combinaties aan bod komen.
De meest voorkomende vorm van cross-validation is k-fold cross validation [3]. Dit houdt in dat de data in k gelijke verzamelingen wordt opgedeeld. Vervolgens wordt er telkens een andere verzameling buiten de training set gehouden. Daardoor wordt er een validating set verkregen bestaande uit één verzameling en de training set wordt gevormd door de $k-1$ andere verzamelingen. De meest gebruikte vorm van deze cross-validation maakt gebruik van 10 verzamelingen, dus $k = 10$ in dat geval.
Met cross-validation is het mogelijk om verschillende algoritmes te vergelijken. Dat gebeurt doordat er iedere iteratie één of meerdere learning sets gemaakt worden, welke ervoor zorgen dat er een model geleerd wordt. Dit model kan vervolgens gebruikt worden om de waarden uit de validating set te voorspellen. Om te kijken welk model de beste uitkomst geeft, wordt er gekeken naar een van te voren gedefinieerde functioneringsmetriek. Dan geeft elk algoritme aanleiding tot k samples uit de functioneringsmetriek. Deze k waarden kunnen op basis van allerlei methodes, zoals bijvoorbeeld het gemiddelde, gebruikt worden om de verschillende algoritmes te vergelijken. Er zijn eigenlijk twee verschillende doelen die bereikt kunnen worden met cross-validation. Er kan worden uitgerekend hoe goed één bepaald algoritme functioneert bij de data. Daarentegen kan er ook worden gekeken naar het functioneren van twee verschillende algoritmes om uiteindelijk te kunnen zeggen welke van de twee nauwkeuriger is. Deze twee doelen komen ongeveer op hetzelfde neer uiteindelijk. Als het namelijk maar eenmaal bekend is hoe het te berekenen is of een algoritme goed functioneert, kunnen ook de algoritmes onderling vergeleken worden.

Er zijn verschillende manieren om de modellen te valideren [3]:

- Resubstitution validation
  Bij resubstitution validation wordt er een model gefit bij een dataset en deze dataset wordt vervolgens ook gebruikt voor de validatie. De learning en validating sets zijn gelijk. Alle data wordt nu gebruikt voor de fit, maar er is een grote kans op over-fitting. Dit houdt in dat de datapunten binnen de dataset wel goed voorspeld worden, maar de toekomstige, onbekende datapunten juist niet.

- Hold-out validation
  Om over-fitting te voorkomen, is het goed om een onafhankelijke testset en een trainingset te hebben. De meest voor de hand liggende manier om dat te doen is door de data te splitsen in twee niet-overlappende verzamelingen. De test-data wordt niet gebruikt tijdens het maken van het model. Op deze manier is er geen overlap tussen de training set en de validating set. Een nadeel aan deze manier van werken is echter het feit dat niet alle data gebruikt wordt om een model te maken. Bovendien is het resultaat erg afhankelijk van hoe de training en de validating sets zijn gekozen. Om dit probleem te omzeilen, is het mogelijk om de hold-out validation meerdere malen uit te voeren en om dan te middelen. Dit dient echter op een systematische manier te gebeuren, anders kan het zijn dat bepaalde datapunten heel vaak in de training set voorkomen en andere punten juist helemaal niet.

- k-fold cross-validation
  Bij k-fold cross-validation wordt een dataset in k verzamelingen gesplitst. De validating set bestaat uit één van deze verzamelingen, de training set bestaat uit de andere $k-1$. De verzamelingen worden iedere keer op een systematische manier gemaakt om er zeker van te zijn dat elke verzameling een goede representatie is van de volledige dataset.

- Leave-one-out cross-validation (LOOCV)
  Eigenlijk is LOOCV een speciaal geval van de k-fold cross-validation met k gelijk aan het aantal datapunten. Dat betekent dat er iedere keer slechts één punt in de validating set komt, alle andere datapunten zitten in de training set. Een benadering op basis van LOOCV is bijna unbiased, maar de variantie is hoog. Daardoor zijn de benaderingen onbetrouwbaar. Deze manier van cross-validation wordt veel toegepast bij zeldzame gebeurtenissen, vooral in de bio-informatica. Een voordeel van deze vorm van cross-validation is het feit dat er een expliciete formule voor is:

$$\hat{V}(y_{n+1}|x_{n+1}, y_1, ..., y_n, x_1 ..., x_n) = \sum_{i=1}^{n} \frac{(y_i - (X\beta)_i)^2}{(I - X(X^T X)^{-1} X^T)_{ii}}. \tag{21}$$

  Hierin is y_i de echte waarde van de te voorspellen grootheid en $X\beta$ is de voorspelde waarde. De $n \times p$ matrix X bestaat uit de waarden van de voorspellers. I is de $n \times n$ identiteitsmatrix. Hoe lager deze waarde is, hoe beter het model is. Een lage waarde houdt namelijk in dat de afstand tussen de voorspelde waarde en de werkelijke waarde klein is.

- Repeated k-fold cross-validation
  Bij repeated k-fold cross-validation wordt een k-fold cross-validation meerdere malen herhaald. De data wordt herschikt voor elke ronde. Op deze manier worden er meerdere benaderingen verkregen en dus kan er een betrouwbaardere waarde gevonden worden voor de functionering van een algoritme of voor de vergelijking van twee of meer algoritmes.

Alle soorten validatie hebben ieder voor- en nadelen. Sommige hebben een overlappende training en validating set en daardoor kunnen ze aanleiding geven tot over-fitting. Anderen hebben weer

een hele kleine validating set en hebben daardoor een grote variantie. In het ideale geval is er een voldoende grote verzameling van onafhankelijke metingen. Het criterium om een voldoende grote verzameling te hebben, is door de meeste statistici gezet op een sample van 30 of meer [3]. Om onafhankelijke metingen te kunnen krijgen, moet het zo zijn dat de factoren die de metingen beïnvloeden onafhankelijk zijn gedurende iedere run van het algoritme. Deze factoren bestaan uit de training data en de test data. Als er dus datapunten zijn die voor meerdere rondes gebruikt worden om het model te testen, zijn de metingen niet meer onafhankelijk. De datasets moeten bovendien niet overlappen. Dus data die in de ene training set zit, kan niet ook in de andere training set terecht komen. Echter, in de praktijk mogen training sets overlappen, maar de validating sets moeten onafhankelijk blijven. Dit is omdat de meting van het functioneren beïnvloed wordt op twee manieren, namelijk direct door de validating set en indirect door de training set. De training set geeft namelijk aanleiding tot een model. Dit model wordt vervolgens weer gebruikt bij het schatten van de validating set. Het verschil tussen de validating set en deze schatting komt terecht in de metingen van de functioneringsmetriek.
k-fold cross-validation voldoet aan bovengestelde eisen die praktisch nodig zijn. De training sets overlappen, maar de validating sets zijn onafhankelijk. Het enige wat nu nog bepaald moet worden, is de grootte van k. In eerste instantie lijkt een grote waarde van k wenselijk, want in dat geval is de training set groot. Een grote training set betekent dat het model gebaseerd zal zijn op een verzameling die erg lijkt op de volledige dataset en dus zal het model de datapunten goed voorspellen. Bovendien geeft het meer benaderingen die aangeven hoe dicht de voorspelde waarden bij de echte waarden zitten. Meer waarden betekent dan ook dat de schatting betrouwbaarder wordt. Een groot nadeel is echter dat de overlap tussen de training sets ook groter wordt met een grotere k. Als er bijvoorbeeld vijfvoudige cross-validation gedaan wordt, is $3/4$ van alle training sets gedeeld met andere training sets. Voor $k = 10$ is dit al $8/9$. Daarbij komt dat de grootte van de validating set klein wordt als k groot wordt. Dit leidt tot een kleinere precisie. Als de validating set namelijk bestaat uit bijvoorbeeld 10 waarden, dan kan er gemeten worden met een nauwkeurigheid tot 10 procent, terwijl dit tot 5 procent had gekund als de set 20 waarden zou bevatten.
Alle voor- en nadelen zijn afgewogen en in de statistiek is het algemeen geaccepteerd om $k = 10$ te nemen [3]. Dan is het model namelijk gebaseerd op 90 procent van de data, wat een goede hoeveelheid is om te kunnen generaliseren naar de volledige dataset.
De cross-validation kan gedaan worden voor het voorbeeld van de yield. De LOOCV levert een directe methode op, welke in R kan worden gebruikt. De matrix X wordt gemaakt uit een kolom met enen (om het intercept mee te kunnen nemen) en de waarden voor T, P en de interactie tussen T en P. Deze interactie bestaat uit de elementsgewijze vermenigvuldiging van T en P. De vector β wordt gemaakt uit β_0, β_1, β_2 en β_3 uit het vorige hoofdstuk en y bestaat uit de waarden voor de yield. I wordt gedefinieerd als de 20×20 identiteitsmatrix. Als de benodigde matrices en vectoren gemaakt zijn, kan (21) in R worden ingevoerd:

$$\text{sum((y} - \text{X\%} * \text{\%beta)} \wedge 2/\text{diag(I} - \text{X\%} * \text{\%solve(t(X)\%} * \text{\%X)\%} * \text{\%t(X))).}$$

Dit levert voor dit model een waarde op van 68395760.

2.5.4 De p-waarde en een plot van de residuen

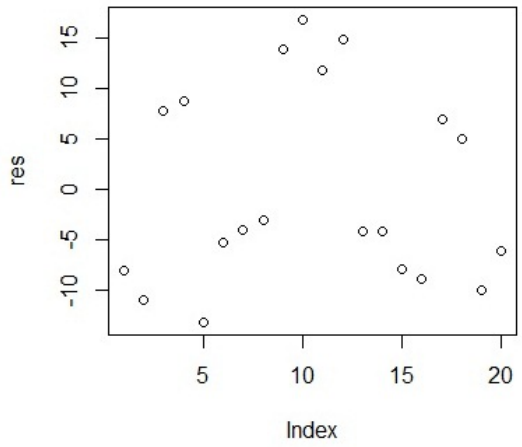
Er kan wel een beste waarde voor de gevraagde β_i bepaald worden, maar wanneer is deze waarde ook echt significant? Een maat hiervoor is de p-waarde. Deze waarde geeft eigenlijk aan of de parameter β_i ook echt significant van 0 verschilt. Er wordt eigenlijk een hypothese-toets gedaan met de hypotheses
$H_0 : \beta_i = 0$

$H_1 : \beta_i \neq 0$
Nu is het te hopen dat de hypothese H_0 verworpen mag worden. Dan is er namelijk een significante waarde voor β_i gevonden. Als H_0 niet verworpen mag worden, is β_i niet significant anders dan 0 en wordt gelijk gesteld aan 0. Hierdoor zal variabele i uit de regressievergelijking wegvallen. Om dit te kunnen testen, wordt gebruik gemaakt van de toetsingsgrootheid $T = \frac{\beta_i}{\mathrm{sd}(\beta_i)}$, met sd de standaarddeviatie. Deze toetsingsgrootheid heeft een student t-verdeling met in het geval van lineaire regressie $n - 2$ vrijheidsgraden, met n het totaal aantal meetpunten [7]. Op basis van de berekende toetsingsgrootheid T en het gegeven dat deze t_{n-2}-verdeeld is, berekent R een maat voor de waarschijnlijkheid van de nulhypothese. Deze waarschijnlijkheid heet de p-waarde en deze is te vinden als er gekeken wordt naar de summary van de fit in R en is makkelijk op te vragen door summary(fit) in te voeren. Als deze p-waarde kleiner is dan een bepaalde α (bijvoorbeeld 0,05 of 0,1), kan de nulhypothese verworpen worden. Dat betekent dat getest kan worden of alle onafhankelijke variabelen ook significant zijn. In het voorbeeld van de yield kan via de summary van de lineaire fit die gedaan is, gekeken worden of alle variabelen significant zijn. Onderstaande resultaten (tabel 3) laten zien dat met een waarde van $\alpha = 0.1$ alle variabelen significant zijn (alle p-waarden zijn kleiner dan 0.1). Als er voor $\alpha = 0.05$ gekozen wordt, is P niet meer helemaal significant. Echter, de p-waarde is wel dusdanig klein dat er wel een zekere invloed van P is in de lineaire regressie.

variabele	t-value	Pr(<| t |)
(intercept)	9.082	1.03 e-07
T	-4.261	0.000598
P	-1.861	0.081179
T:P	4.012	0.001007

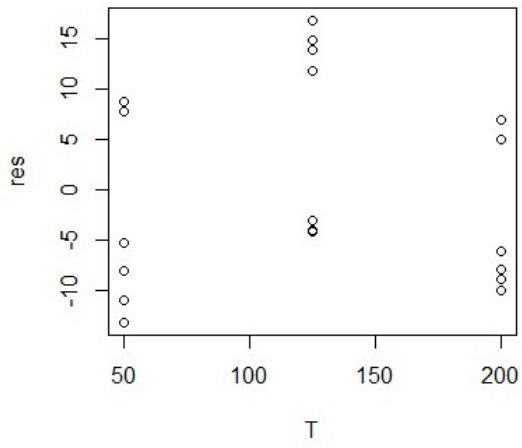
Tabel 3: De p-waarden voor P, T en de interactie hiertussen

Alleen het bestuderen van de p-waarden is echter niet genoeg om iets te kunnen zeggen over hoe goed het model de meetwaarden representeert. Ook het bekijken van de plotjes van de residuen kan veel informatie geven. Als hierin een random patroon zit, is het model waarschijnlijk goed. Als er nog een verband in de residuen zit, is er waarschijnlijk een onafhankelijke variabele toe te voegen waardoor dit verband verklaard wordt. Door het toevoegen van deze extra variabele zal de fit beter worden en bij de nieuwe fit zal er waarschijnlijk geen verband meer tussen de residuen zichtbaar zijn.
Allereerst kan er voor het model van de yield naar de plot van de residuen gekeken worden. In figuur 5 lijken de residuen redelijk random verdeeld te zijn.



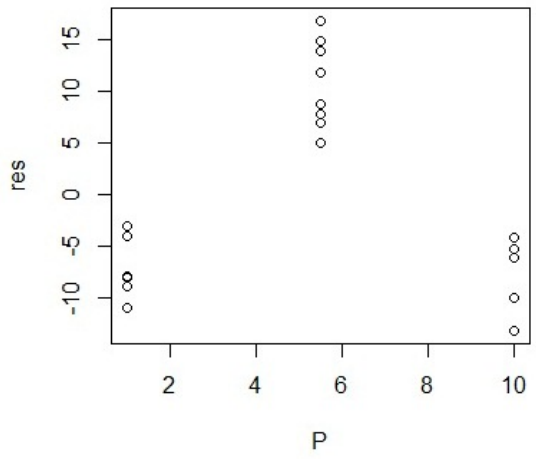
Figuur 5: Plot van de residuen

Echter, de residuen kunnen ook geplot worden tegen de temperatuur. Dit is gedaan in figuur 6.



Figuur 6: Plot van de residuen tegen de temperatuur

In figuur 6 is het te zien dat er geen random patroon in de residuen zit. Er zit nog een parabool in lijkt het. Daarom zou het geprobeerd kunnen worden om ook een kwadratische term van de temperatuur mee te fitten. Deze kwadratische term in de temperatuur zal dan het patroon eruit halen en dan zullen random residuen worden verkregen. Ditzelfde geldt voor de druk, zoals blijkt uit figuur 7. Ook hier zit nog een parabool in de residuen, dus ook de kwadratische term voor de druk zal meegenomen worden in de nieuwe fit.



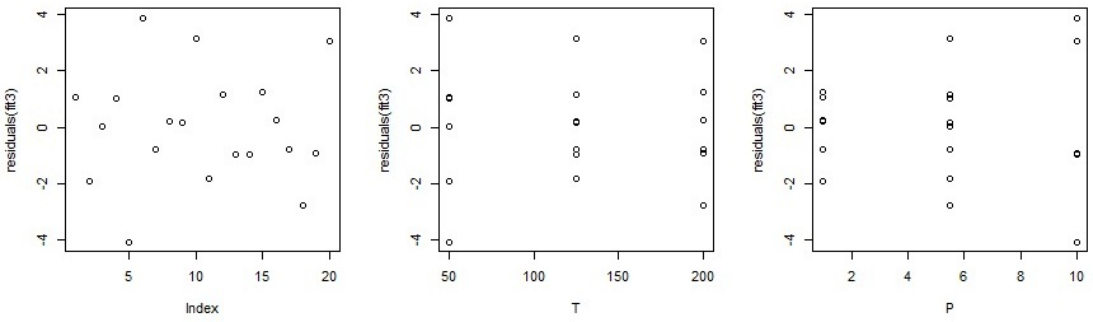
Figuur 7: Plot van de residuen tegen de druk

Nu kan er een nieuwe fit worden gedaan waarin de onafhankelijke variabelen P, T, de interactie tussen P en T, P^2 en T^2 meegenomen worden. Deze fit kan weer eenvoudig bepaald worden door in R in te voeren fit=lm(Y $\sim$ P+T+T:P+T2+P2), waarbij T2 en P2 staan voor de vectoren van de gekwadrateerde waarden van T en P (deze vectoren zijn apart ingevoerd). Door summary(fit) kunnen nu de p-waarden worden gevonden. In tabel 4 zijn de waarden van de bijbehorende β's en de p-waarden weergegeven.

variabele	waarde β	t-value	Pr(>| t |)
(Intercept)	64.2072310	19.668	1.35e-11
P	6.3447972	9.392	2.02e-07
T	-0.0471429	-0.934	0.366
P2	-0.8377425	-15.768	2.62e-10
T2	-0.0010603	-5.544	7.23e-05
P:T	0.0444444	18.255	3.69e-11

Tabel 4: De p-waarden voor P, T, T^2, P^2 en de interactie tussen P en T

Wat opvalt is dat de waarde voor T nu niet meer significant anders is van nul. Dit betekent dat T ook wel uit het model gehaald zou kunnen worden in dit geval. Het is echter de afspraak om als er hogere machten van een variabele in het model terechtkomen, de variabele zelf ook te laten staan. Dit is om een eventuele verschuiving die veroorzaakt kan worden door de eenheden aan te passen ook nog goed te laten fitten.
Natuurlijk moeten ook nu de plotjes van de residuen bekeken worden. Deze plotjes, de algemene residuen, de residuen versus temperatuur en de residuen versus druk, staan in figuur 8 weergegeven.



Figuur 8: Plot van de residuen van het model met kwadratische variabelen, links de overall plot, in het midden de plot tegen temperatuur, rechts de plot tegen druk

Deze residuen zijn nu inderdaad random verdeeld.
Er kan ook even gecheckt worden of de AIC van het model met de kwadratische termen lager is dan de AIC van het model zonder de kwadratische termen. Voor de berekening van de AIC-waarden wordt weer de functie AIC(...) gebruikt. Voor het model zonder de kwadratische termen is een AIC gevonden van 156,64. Door de toevoeging van de kwadratische termen is een AIC van 97,36 gevonden. Het is nu dus ook met de AIC's te zien dat de toevoeging van de kwadratische termen gunstig is voor het model.

2.5.5 Bayesiaanse benadering: Bayes factor

In de Bayesiaanse benadering is het selecteren van een model betrekkelijk simpel. Als men denkt dat bepaalde regressiecoëfficiënten gelijk zijn aan nul, dan kan er een prior gevonden worden waarin dit het geval is. Een veelgebruikte manier om dit weer te geven is door de regressiecoëfficiënt van variabele j te definiëren als $\beta_j = z_j \times b_j$, waar $z_j \in 0, 1$ en b_j is een reëel getal. Met deze parameterisatie wordt de regressievergelijking gegeven door:

$$y_i = z_1 b_1 x_{i,1} + ... + z_p b_p x_{i,p} + \epsilon_i.$$

De z_j's geven dus aan welke coëfficiënten gelijk zijn aan nul [2]. Elke waarde van $z = (z_1, ...z_p)$ correspondeert met een ander model, oftewel met een verschillende verzameling van variabelen die een non-zero regressiecoëfficiënt hebben.
In de Bayesiaanse modelselectie wordt dan voor ieder model (dus voor iedere z) een posterior distribution bepaald. Hiervoor is een joint prior nodig op z, β, σ^2. Het is gebleken dat een bepaalde versie van de g-prior ervoor zorgt dat het berekenen van $p(y|X, z)$ mogelijk is voor iedere z [2]. Als de prior voor de modellen gegeven is door $p(z)$, is de posterior te bepalen via

$$p(z|y, X) = \frac{p(z)p(y|X, z)}{\Sigma_{\tilde{z}} p(\tilde{z})p(y|X, \tilde{z})}.$$

Twee modellen kunnen worden vergeleken door te kijken naar de posterior odds:

$$\text{odds}(z_a, z_b|y, X) = \frac{p(z_a|y, X)}{p(z_b|y, X)} = \frac{p(z_a)}{p(z_b)} \times \frac{p(y|X, z_a)}{p(y|X, z_b)}$$

Dit kan gezien worden als posterior odds = prior odds × "Bayes factor". De Bayes factor geeft aan hoe erg de data model z_a prefereren boven model z_b. Een uitdrukking voor de Bayes

factor kan gevonden worden door de marginale kans te berekenen. Dat kan via het oplossen van onderstaande integraal:

$$\int\int p(y,\beta,\sigma^2|X,z)d\beta d\sigma^2 = \int\int p(y|\beta,X)p(\beta|X,z,\sigma 2)p(\sigma^2)d\beta d\sigma^2$$

Uiteindelijk wordt dan onderstaande uitdrukking voor de Bayes factor gevonden [2]:

$$\frac{p(y|X,z_a)}{p(y|X,z_b)} = (1+n)^{(p_{z_b}-p_{z_a})/2}(\frac{s^2_{z_a}}{s^2_{z_b}})^{1/2} \times (\frac{s^2_{z_b}+SSR^{z_b}_g}{s^2_{z_a}+SSR^{z_a}_g})^{(n+1)/2}. \tag{22}$$

2.6 Gemengde lineaire modellen

Bij het maken van een model wordt altijd een fout gemaakt. In de voorgaande lineaire modellen is er aangenomen dat de fout normaal verdeeld is. Dit komt er dan eigenlijk op neer dat de gemaakte fout in meting 2 onafhankelijk is van de fout in meting 1. Echter, voor de weersvoorspelling hoeft dit niet te gelden. Er kan daar een correlatie zijn tussen de fout in meting 2 en die in meting 1. Als er bijvoorbeeld een koude dag is voorspeld voor de maandag, maar het wordt dan juist warm, zal er op dinsdag niet meteen een veel koudere dag volgen.
Deze correlatie komt tot uitdrukking in de zogenaamde gemengde lineaire modellen (linear mixed models). Dit model kan op twee manieren geschreven worden, namelijk als een subject specifiek model of als een marginaal model. In het eerste geval wordt dit weergeven als

$$y = X\beta + Z\gamma + \epsilon. \tag{23}$$

Hierin bestaat X uit de onafhankelijke variabelen en β wordt weer gevormd door de regressiecoëfficiënten, net als in het gewone lineaire model het geval is. Ook ϵ is net zo gedefinieerd als in het gewone lineaire model, namelijk als de fout die normaal verdeeld is met verwachtingswaarde 0 en covariantie $\sigma^2 I$. De extra term $Z\gamma$ bestaat uit de matrix Z die bekend is en γ is een onbekende random vector die onafhankelijk is van ϵ.
De tweede manier om het model weer te geven is als een marginaal model:

$$y = X\beta + \epsilon. \tag{24}$$

Hierin zijn X en β gelijk aan het gewone model, maar ϵ is nu normaal verdeeld met verwachtingswaarde 0 en covariantie Σ.
Beide modellen komen op hetzelfde neer, maar de subject specifieke weergave is iets specifieker. Dat wil zeggen dat de subject specifieke manier omgeschreven kan worden naar een marginaal model, maar andersom hoeft dat niet te kunnen. Om dit aan te tonen kan er gekeken worden naar de varianties. Voor het subject specifieke model is voor de variantie van y gevonden (omdat ϵ en γ onafhankelijk zijn):

$$\begin{aligned}\text{Var(y)} &= \text{Var}(X\beta + Z\gamma + \epsilon) \\ &= \text{Var}(Z\gamma) + \text{Var}(\epsilon) = Z\text{Var}(\gamma)Z^T + \sigma^2 I\end{aligned}$$

Als nu $\Sigma = Z\text{Var}(\gamma)Z^T + \sigma^2 I$, is dit model omgeschreven naar een marginaal model. Andersom, als er een Σ bekend is, hoeft het niet weer in deze specifieke vorm geschreven te kunnen worden, dus een marginaal model hoeft niet per se naar een subject specifiek model omgeschreven te kunnen worden.
Met de library(nlme) kan in R een mixed model gefit worden. Dit gaat op soortgelijke wijze als bij de gewone lineaire regressie, alleen moet er nu ook aangegeven worden welke correlatiestructuur er gebruikt wordt.

2.7 Ridge regressie

Het kan voorkomen dat er hele grote waarden voor β gevonden worden. Dit zou kunnen komen doordat het model instabiel is. De grote $\beta's$ kunnen namelijk een gevolg zijn van multicollineariteit. Daardoor kan het model slechter worden als het gebruikt wordt voor voorspellingen. Hier is wat op gevonden, namelijk penalized regression. Dat betekent dat grote waarden van β een soort straffactor meekrijgen en daardoor zullen ze minder snel in het model terechtkomen. Bij Ridge regressie wordt de RSS veranderd door er een term met een straffactor bij te zetten. De RSS wordt nu $RSS_{Ridge} = RSS + \lambda\tilde{\beta}^T\tilde{\beta}$, waarbij $\tilde{\beta}$ de vector is van de regressiecoëfficiënten. Op deze manier wordt als de RSS_{Ridge} geminimaliseerd wordt, de optie met de kleinste waarden van β genomen.
Ridge regressie komt dus neer op het minimaliseren van RSS_{Ridge}. Op die manier wordt $\hat{\beta}_{Ridge}$ bepaald [17]. Dit kan weergegeven worden als:

$$\hat{\beta}_{Ridge} = \text{Argmin}_\beta(\sum_{i=1}^{N}(Y_i - \beta_0 - \sum_{j=1}^{p} x_{i,j}\beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2).$$

Het intercept wordt niet verkleind, deze valt namelijk buiten het deel wat te maken heeft met de "straf"λ. De uitdrukking voor $\hat{\beta}_{Ridge}$ kan ook geschreven worden als

$$\hat{\beta}_{Ridge} = \text{Argmin}_\beta||y - X\beta||^2 + \lambda||\beta||^2.$$

Hoe groter λ wordt, hoe kleiner β uiteindelijk wordt. Met Ridge regressie gaat voor zeer grote λ, β richting 0.
Ridge regressie is in eerste instantie ontworpen om te voorkomen dat X^TX singulier wordt. De ridge-coëfficiënt λ fungeert als een perturbatie op de diagonaalelementen van X^TX.
Een andere manier om naar Ridge regressie te kijken is op de Bayesiaanse wijze. Als er vanuit gegaan wordt dat β klein zal zijn, kan dat in de prior distribution worden meegenomen. In dat geval kunnen $\beta_1, ..., \beta_M$ gezien worden als individueel normaal verdeelde random variabelen met gemiddelde 0 en variantie τ^2 [18]. De uitkomstenvector y bestaat uit de voorspellingsfunctie ingevuld voor x en β, met een onbekende fout ϵ, dus $y = f(x, \beta) + \epsilon$. Deze fout is normaal verdeeld met gemiddelde 0 en onbekende variantie σ^2. Als nu wordt aangenomen dat de variantie σ^2 bekend is, kan de prior geformuleerd worden. De prior kan dan de likelihood $L(y, X, \hat{\beta}, \sigma^2)$ van de data wegen aan de kans op $\hat{\beta}$. Nadat deze gewogen likelihood is genormaliseerd, is een posterior verkregen [18]. Deze posterior distribution is:

$$f_{posterior}(\beta|y, X, \sigma^2, \tau^2) = (y - \hat{\beta}^T X)^T(y - \hat{\beta}^T X) + \frac{\sigma^2}{\tau^2}\hat{\beta}^T\hat{\beta}.$$

De Bayesiaanse schatter zal de $\hat{\beta}$ zijn met de grootste kans in de posterior distribution. Echter, er kan herkend worden dat $f_{posterior}(\beta|y, X, \sigma^2, \tau^2) = -RSS_{Ridge}$, als $\lambda = \frac{\sigma^2}{\tau^2}$. Dat betekent dat de Bayesiaanse schatter gelijk zal zijn aan de least squares schatter voor RSS_{Ridge} en een juist gekozen prior variantie τ^2 [18].

3 Interessante artikelen

Het doel van dit onderzoek is om een voorspelling te doen voor het weer van januari 2012 (met name qua temperatuur). Deze voorspelling zal gebaseerd zijn op lineaire regressie. In de literatuur zijn een aantal onderzoeken te vinden waarbij ook gebruik gemaakt wordt van lineaire regressie. In dit hoofdstuk worden twee artikelen besproken met als doel hier wat van te leren voor de regressie die in het volgende hoofdstuk uitgevoerd zal worden.

3.1 Linear Short-term Climate Predictive Skill in the Northern Hemisphere [4]

In dit artikel wordt er een correlatie analyse uitgevoerd. De voorspellers zijn de temperatuur van het zee-oppervlak, de circulatie op het noordelijke halfrond op 700-mb hoogte (Northern Hemisphere 700-mb height, NH 700-mb) en de voorgaande data met betrekking tot de temperatuur en de neerslag. Door gebruik te maken van een multivariaat lineair statistisch model, worden voorspellingen gedaan voor de temperatuur en de neerslag in Europa en de Verenigde Staten. Hierbij wordt gebruik gemaakt van cross-validation om een juiste benadering te verkrijgen.
De resultaten die verkregen worden, zijn niet door het hele jaar gelijke goed. In de maanden januari tot april is de voorspelling relatief goed. De invloed van de temperatuur van het zee-oppervlak (See Surface Temperature, SST) op de mate waarin de voorspelling juist is, is gunstig. Dit kan komen doordat de lower boundary condition een consistente invloed heeft op het klimaat als je kijkt naar termijnen van één tot drie maanden. Verder is SST de enige factor waarin de invloed vanuit de tropen meegenomen wordt.
Dit artikel maakt gebruik van de SST in de voorspelling op basis van eerdere onderzoeken. De SST wordt namelijk beïnvloed door het klimaat in de tropen, bijvoorbeeld door de stroming El Niño/Southern Occilation (ENSO) zoals Opsteegh en Van den Dool (1980) en Hoskins en Karoly (1981) uitleggen. Ook zijn er nog andere stromingen die voor regionale invloeden zorgen (Namias 1982, Lazante 1984, Iwasaka et al. 1987). Er zijn zelfs lokale effecten te vinden die samenhangen met de SST. Er is ook informatie beschikbaar over de quasi-globale luchtdruk op zeeniveau, de data tot 1950 is terug te vinden. Deze zou gebruikt kunnen worden om de voorspelling te verbeteren. Echter, deze data is vooral rondom de tropen niet zo betrouwbaar.
Het klimaat op een bepaalde locatie wordt meestal bepaald door de samengestelde effecten van verschillende systematische relaties. Deze relaties zijn vaak op grote schaal. Dit betekent dat een bepaalde locatie onderdeel is van een gebied dat onder ongeveer dezelfde invloeden staat. Nu kan er multipele regressie worden gedaan voor één bepaalde locatie. Dit zou betekenen dat de coherente patronen waarin gebieden onder dezelfde invloeden bestaan, verloren gaan. Het kan daarom verstandig zijn om te kijken naar voorspellingen voor hele gebieden. Hierdoor kan de voorspelling op een bepaalde locatie uiteindelijk ook beter worden, omdat de informatie over hele gebieden niet vertroebeld zal worden door verkeerde metingen op individuele punten. De fouten in metingen zullen er namelijk snel uitgehaald kunnen worden, omdat deze verbleken bij de andere metingen. Een nadeel van deze werkwijze kan zijn dat de variatie in metingen in een bepaald punt wat te verklaren is door bepaalde fysische invloeden bijvoorbeeld ook als zijnde meetfouten niet meer meegenomen wordt. In dit artikel wordt ervoor gekozen om naar grotere gebieden te kijken. Deze aanpak wordt patern-to-patern strategy genoemd, omdat er naar patronen in gebieden gekeken wordt. Er wordt vanuit gegaan dat de lokale invloeden gering zullen zijn voor de voorspellingen.
Datasets voor 1955-1991 worden gebruikt in dit artikel waarin de quasi-globale zee-oppervlakte temperatuur (SST), de NH700-mb, de temperaturen en neerslagcijfers in Europa en de Verenigde Staten worden meegenomen. De tijd tussen de voorspellers en de voorspelde waarden is één of

drie maanden. De data voor de SST wordt verdeeld in vierkantjes van 10 graden bij 10 graden. De data kan verkregen worden voor kleinere vierkantjes (twee bij twee), maar dan zijn er punten die ontbreken. Door de vierkantjes groter te nemen met een gemiddelde, wordt de hoeveelheid missende data aanzienlijk beperkt. De data voor de vierkantjes die mist, wordt bepaald via interpolatie. De grote vierkantjes die meer dan 2 procent van de data missen, worden uitgesloten (met uitzondering van de Tropical Pacific Belt).
Voor de voorspelling van de temperatuur in Europa en de VS worden de voorgaande waarden van temperatuur gebruikt, samen met de waarden van SST en NH 700-mb. Echter, voor de voorspelling van de neerslag worden alleen de waarden van SST en NH 700-mb meegenomen. Deze keuze is gebaseerd op een studie van Huang en Van den Dool in 1993, waarin aangetoond wordt dat er geen lineaire relatie bestaat tussen de komende neerslag en de voorbije neerslag. Er zijn drie methoden die gebruikt kunnen worden voor de multivariate regressie. Dit zijn canonical correlation analysis (CCA), singular value decomposition (SVD) en principal oscillation pattern (POP) analysis. In dit artikel wordt gebruik gemaakt van CCA. Dat is eigenlijk een multipele lineaire regressie waarbij gebruik wordt gemaakt van een specifieke toepassing van empirical orthogonal function (EOF) analyse. De data wordt gestandaardiseerd voordat er CCA gedaan wordt. De CCA wordt uitgevoerd. Er wordt een correlatiematrix voor elke voorspelde waarde en de voorspellers gemaakt. Er wordt daarna cross-validation uitgevoerd om overfitting te reduceren.
In de voorspelling van de driemaandelijkse gemiddelde temperaturen, blijkt de winter, het vroege voorjaar en de late zomer het best voorspelbaar te zijn. Ook valt het op dat het vergroten van de lead time tot 10 maanden slechts een kleine invloed heeft op de precisie van de voorspelling. Voor de driemaandelijke gemiddelde temperatuur, wordt er gekeken naar de voorspelling die alleen gebaseerd is op de SST, de 700-mb height en alleen op de U.S. surface temperature. Het blijkt dat de SST de meeste voorspellende informatie bevat. Vooral in de winter is de SST zeer voorspellend, wat samenhangt met de ENSO. Op basis van deze bevindingen, is er gekozen voor een model waarin de SST twee keer zoveel gewicht krijgt. Voor de maanden januari-februari-maart is de skill van dit model 0,35, terwijl voor het model zonder het dubbele gewicht dit 0,31 is. De skill voor de voorspelling voor juni-september is 0,29.
De voorspelbaarheid van de maandgemiddelden is lager dan de voorspelbaarheid van de driemaandelijkse gemiddelden. Dit is voor het gehele jaar het geval. Voor de maandelijkse gemiddelden is het weer zo dat de winterperiode beter te voorspellen is dan de zomerperiode.
De beste voorspelbaarheid lijkt samen te hangen met de ENSO. Om dit te testen zijn er lag correlation fields gemaakt. De gemiddelde temperatuur in de late zomer (juli-augustus) in veel van de staten van de VS vertoont een positieve correlatie met de tropische SST in de Stille Oceaan in de voorgaande winter en lente, de SST in de Indische Oceaan tegen het einde van de winter en het vroege voorjaar en de SST in de Atlantische Oceaan in de vroege zomer.
Op gelijke wijze als bij de temperatuur, wordt er ook een voorspelling gedaan voor de neerslag. De totale neerslag gedurende drie maanden wordt voorspeld als functie van de SST en de 700-mb height. De SST heeft weer dubbel gewicht in het model. De voorspellende waarde van het model is voor de neerslag beduidend kleiner dan voor de temperatuur het geval is. Net als bij de temperatuur is er in de late winter en in de vroege lente een piek te zien waarin de voorspelbaarheid groot is. De piek die de temperatuur in de zomer ook nog had, heeft de neerslag niet.
Er wordt ook geprobeerd om een voorspelling te doen voor de 300-mb height in het Pacific/Noord-Amerikaanse gebied. Dit wordt gedaan omdat er vaak een aanname wordt gedaan dat het klimaat op het land beïnvloed wordt door de hogere luchtstromingen. Er worden modellen geprobeerd die de NH 700 mb-height en de SST meenemen (waarbij de SST een dubbele weging heeft), alleen de 700 mb-height of alleen de SST. Het blijkt dat het eerste model het beste werkt. Net als bij de temperatuur is de fit het beste tegen het einde van het koude seizoen en de fit is ook goed in de

late zomer. De verschillen gedurende het jaar zijn echter minder groot dan bij de temperatuur het geval is. De modellen zijn ook voor het noordelijke halfrond geprobeerd. Hier blijkt weer dat het model met de 300-mb height en de SST het beste werkt. De voorspelling voor het weer in Europa met hetzelfde model toont niet zo veel overeenkomsten. De voorspelbaarheid is laag als de datasets van de Europese weerstations gebruikt worden. Geografisch zijn er wel verschillen merkbaar in hoe goed het model werkt.

3.2 Predicting Summertime Caribbean Pressure in Early April [5]

In dit artikel wordt met behulp van een lineaire regressie de luchtdruk op zeeniveau bepaald. Variaties in de luchtdruk in de zomer boven de Caribische Zee zijn gerelateerd aan de activiteiten van de Atlantische tropische cyclonen (TC) en aan de variaties in regionale regenval. Diverse studies hebben aangetoond dat een lage druk op zeeniveau in het Caribische gebied, leidt tot toename van de cycloonactiviteit. Ook zit er een relatie in de druk en de regenval in het zomerseizoen. De drukvariaties in het Caribische gebied zijn gerelateerd aan de variaties in de luchtvochtigheid en tropische winden in het Atlantisch gebied. Deze variaties in de druk in het zomerseizoen zijn echter heel lastig te voorspellen. De gemiddelde druk in april-mei werd altijd als voorspeller genomen, maar dit verklaart slechts een deel van de variatie in de druk in de zomer. Onderzoekers hebben de luchtdruk en de regenval gerelateerd aan variaties in El Niño-Southern Oscillation (ENSO). Naast de ENSO-gerelateerde variaties binnen de luchtdruk op zeeniveau, is het ook aangetoond dat de subtropische ridge in de Atlantische Oceaan in de winter en in het voorjaar gerelateerd is aan de druk in het Caribische gebied in de zomer. Deze informatie is beschikbaar voordat het zomer is, dus op basis hiervan zou een voorspelling gedaan kunnen worden.
In dit artikel wordt gebruik gemaakt van datasets met betrekking tot de druk, wind en temperaturen van het zeewater aan het oppervlak (sea surface temperatures, SST) van the Comprehensive Ocean Atmosphere Data Sets (COADS). De indices van de druk op zeeniveau (sea level pressure, SLP) zijn berekend met behulp van data van the Global Historical Climatology Network en aangepast met behulp van data die verkregen is van Monthly Climatological Data of the World (U.S. Department of Commerce 1995a). Met behulp van de data van COADS is er een Atlantische ridge vastgesteld tussen twintig en dertig graden west. Dit is gevonden door het grid van de data af te zoeken naar de hoogste waarde voor de druk. De zes punten die gevonden zijn, worden gemiddeld en zo wordt er een ridge gevonden. Deze waarden van de COADS data worden daarna naar 1995 doorgetrokken door gebruik te maken van de data uit the Climate Diagnostics Bulletins (U.S. department of Commerce 1995b) en door gebruik te maken van de waarden uit de dagelijkse analyse van het oppervlak van de Atlantische Oceaan. Zo wordt er een continue dataset voor de periode 1950-1995 gecreëerd. Deze data wordt gebruikt als de training set.
De Noord-Atlantische (NATL) SST's zijn bepaald door gebruik te maken van de gemiddelde SST's in het gebied van de gereconstrueerde SST-regio's zoals gedaan is door Smith et al (1996). Daarbij zijn er Niño 3.4 SSTA's gemaakt met behulp van dezelfde globale datasets. Bovendien zijn er indices geconstrueerd voor de druk op zeeniveau in de maanden april-mei (AM), juni-september (JJAS), juni-juli (JJ) en augustus-september (AS). Alle zo verkregen data kan worden gebruikt om cross-correlation en autocorrelation coëfficiënten te bepalen. Met behulp hiervan kan de hoeveelheid vrijheidsgraden worden gereduceerd op de manier als in Leith (1973) is gedaan. Uit de coëfficiënten blijkt onder meer dat er een relatie is tussen de subtropische ridge en de temperaturen van het zee-oppervlak in de Noord-Atlantische gebieden. Als de temperaturen van het zeewater aan het oppervlak lager zijn, is de druk die samenhangt met de subtropische ridge groter. Deze link is vaker genoemd in de literatuur, maar nog bijna nooit teruggevonden

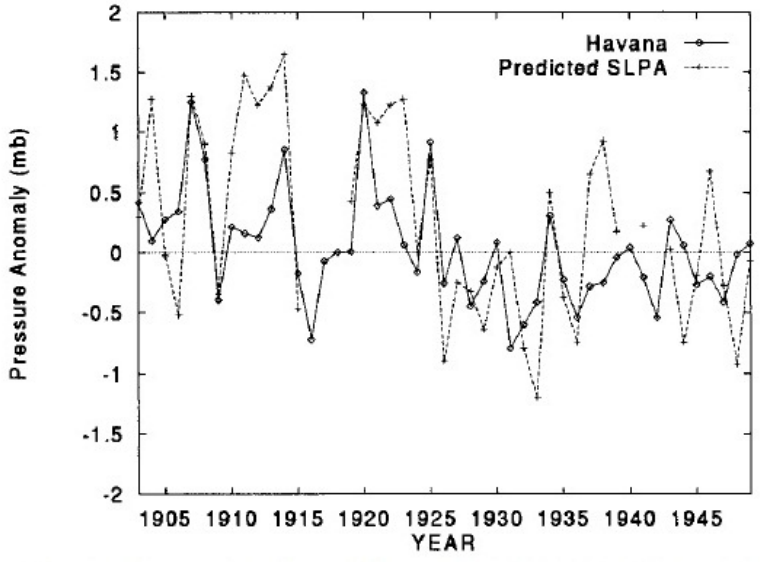
in een voorspelling. Ook blijkt het dat er een correlatie bestaat tussen de Noord-Atlantische SST's, de ENSO-invloeden en de druk op zeeniveau. Dit betekent dat vanuit deze analyse een voorspelling gedaan kan worden waarin de combinatie van de ridge, de Noord-Atlantische SST's en de informatie over de ENSO (in januari-maart) wordt gemaakt. Deze informatie is allemaal beschikbaar begin april en dus kan er op deze manier een verbetering van de voorspelling van de druk op zeeniveau bewerkstelligd worden.
Met behulp van multipele lineaire regressie en met behulp van de method of leaps and bounds (Furnival en Wilson, 1974) kan een voorspelling voor de luchtdruk op zeeniveau in het Caribische gebied geconstrueerd worden. Deze methode kan begin april uitgevoerd worden. Hierbij wordt de data van 1950 tot 1995 als training set gebruikt en de data van 1903 tot 1949 wordt gebruikt als verificatie. De voorspelde waarde is de luchtdruk op zeeniveau in juni-september in het Caribische gebied. De voorspellers zijn de ridge sterkte (MR), de Niño 3.4 van januari-maart (NINO) en de temperatuur van het zee-oppervlak in het Noord-Atlantische gebied in januari-maart (NATL). Deze drie voorspellers zijn verkregen uit data als eerder beschreven en het is gebleken dat vijftig procent van de variantie van de luchtdruk op zeeniveau in juni-september verklaard kan worden door de variatie in deze voorspellers. De regressievergelijking die gevonden wordt is gegeven door:

$$SLPA = \beta_0 + \beta_1(MR - 1000\,mb) + \beta_2(NATL) + \beta_3(NINO) \tag{25}$$

waarbij de waarden van β gegeven worden door

$$\beta_0 = -2.163, \beta_1 = 0.082, \beta_2 = -0.491, \beta_3 = -0.127.$$

Uit vergelijking 25 kan nu de relatie tussen de voorspellers en de voorspelde waarden worden gehaald. Zwakke ridge condities, warme zee-oppervlakken in januari-maart en warme Niño 3.4 condities bijvoorbeeld, zorgen voor een lagere druk in het Caribische gebied dan normaal.
Om nu te kunnen testen of het model juist is, wordt gebruik gemaakt van de onafhankelijke test-set voor de periode 1903-1949. Voor deze dataset (Parker et al. SST-dataset en drukmetingen uit Havana, Cuba) wordt op gelijke wijze als bij de training set waarden voor de ridge verkregen. Voor de jaren 1916-1918, 1940 en 1942 zijn geen geschikte drukmetingen, dus uiteindelijk wordt er een sample met data van 42 jaar gemaakt om mee te testen. In onderstaande figuur (figuur 9) is te zien dat de voorspelde waarden redelijk goed overeenkomen met de werkelijke waarden.



Figuur 9: De gemeten waarden in Havana en de voorspelde waarden [5]

Er is een correlatie van $r = 0.63$, wat veertig procent van de variantie verklaart. Er is dus een lichte degradatie (in de training set was er vijftig procent gevonden), maar de voorspelling is toch redelijk goed. Ook is er met behulp van test-samples gekeken of het meenemen van de voorspeller El Niño 3.4 zinvol is en of er ook voorspellingen gedaan kunnen worden voor de periode juni-juli en augustus-september. Het meenemen van de voorspeller El Niño 3.4 blijkt wel toegevoegde waarde te hebben en de voorspellingen van juni-juli en augustus-september zijn minder betrouwbaar dan de voorspellingen over de gehele periode juni-september.

4 Weersvoorspelling

4.1 Factoren die van invloed zijn op het weer

Het weer is een heel veranderlijk fenomeen. Hoe vaak gebeurt het niet dat er op een mooie zomerse dag zo opeens een plensbui langskomt? En afgelopen zomer wilde het maar niet warm worden. De droge, zonnige dagen waren ver te zoeken. Bovendien, waarom is er het ene jaar wel sneeuw en het volgende niet? Heeft de temperatuur van de zomermaanden invloed op de wintermaanden?
Om al dit soort vragen te beantwoorden, is het goed om eens te bekijken wat voor factoren er allemaal invloed hebben op het weer en het klimaat. Deze factoren zijn te verdelen in twee soorten: interne en externe factoren. Interne factoren zijn dingen als de wisselwerking tussen de atmosfeer, de oceaan, de ijsbedekking en de biosfeer. Een goed voorbeeld hiervan is El Niño, waarover later nog meer informatie zal volgen. Externe factoren zijn dingen als de zonneactiviteit en de vulkaanuitbarstingen [13]. Dit duidt al aan dat er een heleboel factoren zijn die gezamenlijk zorgen voor het weerbeeld. Een aantal van die factoren zijn hieronder kort toegelicht:

- Oceanen
  De grote waterpartijen warmen langzaam op en koelen ook langzaam weer af. Doordat de luchtstromingen over deze oceanen lopen, kan het zijn dat deze luchtstromingen warmere of juist koelere lucht naar het land toe brengen. Op deze manier heeft de temperatuur van het zeeoppervlak invloed op de temperatuur aan land. De temperatuur van het oppervlaktewater in zee wordt aangeduid met de sea surface temperature (SST). In de voorspelling die verderop gedaan zal worden, wordt de SST-data ook meegenomen.
  Er is een bijzondere zeestroming in de Stille Oceaan, El Niño. Dit is een soort periodieke stroming met een periode van twee tot zeven jaar. El Niño is een warme stroming. Hierdoor komt er warmer water aan de oppervlakte. De tegenhanger hiervan is La Niña, hiermee wordt koud water naar de oppervlakte gebracht. Ook de oppervlaktedruk verschilt tijdens deze oscillaties, wat de Southern Oscillation genoemd wordt. De veranderingen door El Niño, La Niña en de Southern Oscillation wordt de ENSO genoemd [12]. Deze stromingen kunnen extreem weer veroorzaken en natuurlijk ook andere temperaturen dan normaal zijn. Meestal treedt El Niño zo rond kerst op, waardoor de winter warmer zal worden in grote delen van de wereld. De deelstromingen van El Niño zijn genummerd. Niño 1.2 loopt van 10 graden Zuiderbreedte tot de nulmeridiaan en van 90 naar 80 graden Westerlengte [16]. Niño 3 loopt van 5 graden Noorderbreedte tot 5 graden Zuiderbreedte en van 150 naar 90 graden Westerlengte [16]. Niño 4 bevindt zich van 5 graden Noorderbreedte tot 5 graden Zuiderbreedte en 160 graden Oosterlengte tot 150 graden Westerlengte [16]. Niño 3.4 loopt van 5 graden Noorderbreedte tot 5 graden Zuiderbreedte en van 170 tot 120 graden Westerlengte [16]. De meeste invloed op het weerbeeld wordt vaak na drie tot zes maanden na het beginnen van El Niño zichtbaar.

- Arctische Oscillatie
  De Arctische Oscillatie (AO) is een index die gerelateerd is aan de zeeoppervlaktedruk in het gebied boven de 20 graden Noorderbreedte [14]. De druk in het Arctische gebied wordt met een index uitgedrukt waarvan het teken tegengesteld is aan het teken van de index die de druk rond de 37-45 graden Noorderbreedte uitdrukt. Deze indices hebben invloed op het weer in het noordelijke halfrond. Als namelijk de AO positief is in het poolgebied, is de oppervlaktedruk laag in het poolgebied. Dit betekent dat de lucht vanuit de polen niet makkelijk naar de gebieden rond de 37-45 graden Noorderbreedte stroomt, waardoor de koude lucht rond de Noordpool blijft hangen. Vooral in de wintermaanden (december-maart) is de fluctuatie in de luchtdruk rond de Noordpool interessant. Deze luchtdruk

verandert niet volgens een bepaald patroon, maar heeft wel een duidelijke invloed op de temperatuur tijdens de Nederlandse winters. Zoals het KNMI [13] zegt, is het meest duidelijke effect van de AO de positie van de straalstroom, een band met hoge windsnelheden op tien kilometer hoogte op de gematigde breedte. Onder de straalstroom ontwikkelen zich depressies. Hiermee bepaalt de AO dus de positie van hoge en lagedrukgebieden.]]
De NAO (North Atlantic Oscillation) is een soortgelijke index. De NAO kijkt naar het verschil in luchtdruk boven Ijsland en de Azoren. Dit verschil in luchtdruk beïnvloedt de grootte en richting van de Westenwind boven Europa. Als het verschil in luchtdruk groot is (en daardoor de NAO ook groot), zullen de westenwinden sterk zijn en daardoor zullen er koude zomers en milde winters ontstaan. Voor een kleine NAO (dus een negatieve NAO) geldt juist het omgekeerde.
Welke van de twee (de AO of de NAO) beter is voor de voorspelling van de temperatuur in de winter, is niet duidelijk in de literatuur. In dit onderzoek wordt gebruik gemaakt van de NAO, omdat daar waarden van beschikbaar zijn tot 1872. Voor de AO is dit niet het geval.
Een soortgelijke index is er ook voor het zuidelijk halfrond, maar in dit onderzoek wordt naar Nederland gekeken en daarmee is deze index dus niet interessant.

- Zon
  De grootste invloed van de zon op het klimaat is de absorptie van zonnestralen door de atmosfeer en het aardoppervlak. Zoals ook in een rapport over zon en klimaat wordt uitgelegd [15]: de stroming in de atmosfeer wordt voor een groot deel bepaald door de verdeling van de geabsorbeerde zonnestraling en de uitgaande IR-straling. Ook als de zonneactiviteit gelijk blijft, heeft de zon invloed op de klimaatprocessen. Als nu echter de hoeveelheid inkomende zonnestraling varieert, kan de temperatuur op aarde ook variëren. Bijvoorbeeld het bestaan van de zonnevlekkencyclus kan ervoor zorgen dat de temperaturen op aarde jaarlijks wat variëren. In 2010 is er weer een nieuwe zonnevlekkencyclus op gang gekomen. Deze lijkt rustig te verlopen en zal dan een effect hebben van 0,05 graden Celsius ten opzichte van het zonnevlekkenmaximum in 2001. Bij zeer weinig zonne-activiteit kan de temperatuur dalen met zo'n 0,2 graden Celsius [13]. Een publicatie van Haigh et al. toont echter aan dat er over het effect van de zon nog veel onduidelijk is [13]. Bovendien gaat het om zo'n marginale bijdrage dat het in dit onderzoek buiten beschouwing gelaten zal worden.

- Vulkanen, stratosferische waterdamp, koolstofdioxide en wolkenontwikkeling
  Wat de temperatuur zal worden op aarde is afhankelijk van de hoeveelheid inkomende zonneschijn, waarop bijvoorbeeld de zonnevlekken invloed hebben. Echter, de hoeveelheid teruggekaatste IR-straling is ook erg belangrijk. De deeltjes in de atmosfeer bepalen of de IR-straling teruggekaatst kan worden of niet. Als er bijvoorbeeld veel CO_2 en waterdamp in de lucht zit, zal er uiteindelijk minder IR-straling de aarde verlaten. Ook de ontwikkeling van de wolken is van belang.
  Al deze invloeden zijn nog niet helemaal eenduidig vastgelegd in de wetenschap. Ook de invloed van de mens en de onderlinge invloed van temperatuurstijging en bijvoorbeeld wolkenontwikkeling is nog niet helemaal duidelijk. In rapporten als "De staat van het klimaat 2010"[13], uitgegeven door het KNMI, zijn al dit soort eventuele invloeden beschreven. Ook in tal van andere onderzoeken (bijvoorbeeld van het IPCC - Intergovernmental Panel on Climate Change), valt de wetenschappelijke discussie te lezen.
  Omdat de wetenschap er nog niet helemaal over uit is hoe de verhouding van bijvoorbeeld CO_2 in de lucht, de wolkenontwikkeling of de waterdamp in de stratosfeer van invloed is op het weer en het klimaat, zal dit in dit onderzoek achterwege gelaten worden. In dit

onderzoek wordt namelijk alleen gewerkt met erg eenvoudige modellen. Eventueel zouden dergelijke invloeden in een later stadium meegenomen kunnen worden.

4.2 Het model

Evenals in de eerder beschreven artikelen (Linear Short-term Climate Predictive Skill in the Northern Hemisphere [4] en Summertime Caribbean Pressure in Early April [5]) is het de bedoeling om een lineaire regressie te doen om zo te proberen om het weer (met name de temperatuur) in januari 2012 te voorspellen. Hiertoe zijn een aantal datasets opgezocht en bestudeerd. Allereerst is er natuurlijk gekeken naar de temperaturen. Deze data zijn via het KNMI te verkrijgen. Er is een dataset met alle maandelijkse gemiddelde temperaturen en een dataset met de dagelijkse temperaturen, neerslag, windsterkte etc. In eerste instantie is de dataset met de maandelijkse gemiddelde temperaturen gebruikt, omdat er anders te veel variabelen zijn om een regressie mee te doen. Eventueel zou het model uitgebreid kunnen worden met behulp van de tweede file. Ook zijn temperaturen van de zee opgezocht. Hiervoor is een handige 1-dimensionale dataset gevonden met de zeetemperaturen voor de ENSO. Ook is er een grid gevonden met allemaal waarden voor de zeetemperaturen. Hieruit zijn de waarden voor een aantal plaatsen gebruikt. Tenslotte zijn er ook waarden voor de Arctische Oscillatie gevonden. Met al deze data zijn geleidelijk allemaal fits ontstaan en aangepast zoals in de komende hoofdstukken beschreven zal worden. De herkomst van de data is in het hoofdstuk Literatuur terug te vinden. Hier zijn tevens de R-scripten voor de berekeningen uit het volgende hoofdstuk te vinden.

5 Standaard lineaire regressie en selectie met AIC

5.1 Regressie over de temperaturen

Er gaan allemaal speculaties in de rondte; het was een koude zomer, dan wordt het vast ook een koude winter. Of is het juist omgekeerd? Deze uitspraken komen erop neer dat er een verband zou bestaan tussen de temperatuur van een aantal maanden eerder (de zomermaanden als er een voorspelling gedaan wordt over de wintermaanden) en de temperatuur van de te voorspellen maand.
Om te kijken of er inderdaad een verband is tussen de maandtemperaturen kan er een lineaire regressie gedaan worden voor de maand januari waarbij de temperaturen in de maanden april, mei, juni, juli, augustus en september als onafhankelijke variabelen worden gezien. In dit geval wordt er dus gezocht naar een model waarbij drie maanden van te voren een voorspelling gedaan kan worden voor de temperatuur waarbij de afgelopen zes maanden meegenomen worden in de voorspelling. Het allereerste model kan gevonden worden door de ingebouwde functie lm in R te gebruiken. Voordat dat kan, moet natuurlijk de data worden ingelezen. Hiertoe wordt het bestand met de gemiddelde maandtemperatuur bestudeerd. Dit bestand heeft titels boven de kolommen staan welke makkelijk gebruikt kunnen worden in de regressie. Door de data in te lezen in R en dan de juiste kolommen te selecteren (kolom 6 tot en met 11), kunnen de maanden makkelijk worden aangemaakt. De eerste 105 rijen worden geselecteerd. Dit zijn de waarden vanaf 1906 tot 2010. Voor januari wordt de data van de eerste kolom geselecteerd. Hierbij moet er wel aan worden gedacht dat de data een jaar later moeten beginnen en een jaar langer moeten doorlopen. Voor januari wordt daarom element 2 tot en met 106 van de eerste kolom genomen. Op die manier wordt namelijk januari 1907 voorspeld aan de hand van april, mei, juni, juli, augustus en september 1906 etc.
Nu de variabelen allemaal bekend zijn, kan in R met behulp van de functie lm($jan \sim APR + MAY + JUN + JUL + AUG + SEP, data = Data1$) een fit gedaan worden. De uitkomst, inclusief de p-waarden, is hieronder (tabel 5) weergegeven.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	-111.6219	48.0928	-2.32	0.0224
APR	0.1195	0.1856	0.64	0.5214
MAY	0.2786	0.2157	1.29	0.1995
JUN	0.2017	0.2225	0.91	0.3668
JUL	0.0545	0.2036	0.27	0.7896
AUG	-0.0543	0.2268	-0.24	0.8113
SEP	0.4068	0.2276	1.79	0.0770

Tabel 5: De coëfficiënten en de p-waarden voor de regressie van de maandtemperaturen

Hieruit blijkt dat alleen het intercept een significante bijdrage levert en bij een α van 0,1 levert ook de maand september een significante bijdrage. Op dit model kan nu ook de functie stepAIC worden losgelaten. Op die manier wordt het model met de kleinste AIC verkregen. Het resultaat staat hieronder (tabel 6) , alleen de maanden mei en september worden nog meegenomen en een as-afsnede. De AIC van dit model is 985.82. De AIC van het model zonder de stepAIC is 992.38.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	-87.6710	35.2162	-2.49	0.0144
MAY	0.3573	0.1984	1.80	0.0748
SEP	0.4483	0.2087	2.15	0.0341

Tabel 6: Resultaten van de stepAIC voor de regressie van de maandtemperaturen

Nu kan er natuurlijk ook gekeken worden wat er gebeurt als de interacties meegenomen worden. Dit wordt in R ingevoerd en ook de stepAIC wordt gedaan. Dit levert het onderstaande resultaat op (tabel 7). De AIC van dit model is 982,27.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	-632.4506	430.0343	-1.47	0.1447
APR	8.1920	3.5582	2.30	0.0235
MAY	10.0115	3.1021	3.23	0.0017
JUN	-7.7659	3.0299	-2.56	0.0120
AUG	0.9977	2.5178	0.40	0.6928
SEP	2.2282	1.2786	1.74	0.0847
APR:MAY	-0.0436	0.0186	-2.34	0.0213
APR:JUN	0.0282	0.0149	1.89	0.0622
APR:AUG	-0.0235	0.0153	-1.54	0.1278
APR:SEP	-0.0225	0.0155	-1.45	0.1508
MAY:AUG	-0.0388	0.0172	-2.25	0.0265
JUN:AUG	0.0362	0.0190	1.91	0.0595

Tabel 7: De resultaten voor de regressie van de maandtemperaturen waarbij de correlaties meegenomen zijn

Het valt op dat de maanden april, mei en juni nu een significante bijdrage leveren. Ook de interacties van april en mei en mei en augustus zijn significant.

5.2 Regressie over de SST

Naast de regressie waarbij de gemiddelde maandtemperatuur meegenomen wordt, kan er ook een regressie gedaan worden waarbij de temperatuur van het oppervlaktewater van de zee als onafhankelijke variabele wordt meegenomen.

5.2.1 El Niño

De eerste regressie die gedaan wordt, is de regressie waarbij de oppervlaktetemperaturen van de zeestromingen van El Niño worden meegenomen. Deze dataset is makkelijk geordend en er kan, op dezelfde manier als bij de temperaturen is gedaan, data geselecteerd worden die nodig is. Allereerst wordt er (ook weer met stepAIC) een model gevonden voor de regressie met de stroming Niño 4. Hierin zijn de temperaturen van juni en juli van de Niño 4 significant bij een α van 0,1, zoals hieronder te zien is (tabel 8). De AIC van dit model is 990,31.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	-58.2979	98.0158	-0.59	0.5533
nino4jun	17.3698	8.5499	2.03	0.0448
nino4jul	-14.9661	8.1904	-1.83	0.0706

Tabel 8: Resultaten van de regressie met de stroming Niño 4

Het is nu te zien dat alleen de temperaturen van de zee voor de stroming nino4 in juni en juli meegenomen worden.
De interacties kunnen ook voor dit model weer meegenomen worden. Dit levert een model op met een AIC van 988,53 met onderstaande onafhankelijke variabelen met de p-waarden (tabel 9).

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	4182.2739	1837.4590	2.28	0.0250
nino4jun	18.7837	8.4674	2.22	0.0288
nino4jul	-175.9667	72.3338	-2.43	0.0168
nino4sep	-177.4904	75.3625	-2.36	0.0205
nino4jul:nino4sep	6.6810	2.8872	2.31	0.0227

Tabel 9: Resultaten voor de regressie van Niño 4 met de interacties

Het is te zien dat alle variabelen die in dit model voorkomen, significant zijn. De interactie van juli en september levert een gunstige bijdrage voor het model.
Ook voor de Niño 3.4 kan een soortgelijke regressie gedaan worden. Als hier echter een stepAIC van wordt gedaan, worden alle variabelen geschrapt. Uiteindelijk blijft er alleen een as-afsnede over en heeft het model een AIC van 990,48. Dit betekent dat de Niño 3.4 geen directe invloed heeft op de temperatuur in januari in Nederland.
Voor de Niño 3 wordt ook een regressie gedaan. Dit levert een AIC op van 990,96 na stepAIC en de resultaten staan in tabel 10.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	-30.7403	119.8362	-0.26	0.7981
nino3jun	18.7229	10.2492	1.83	0.0707
nino3jul	-17.2651	9.5334	-1.81	0.0731

Tabel 10: De resultaten van de regressie voor Niño 3

Opvallend is dat hier weer de waarden van juni en juli overblijven in het model. Bij de Niño 4 gebeurt dat ook.
Ook kunnen de interacties meegenomen worden. Dat levert na stepAIC een model op met AIC = 997,63. Deze waarde is hoger dan de AIC van het model zonder interacties. Dit betekent dat er een lokaal minimum van de AIC gevonden is. De variabelen die nog in het model zitten en hun p-waarden staan in 11.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	7567.6476	5295.0076	1.43	0.1563
nino3apr	286.6554	450.6494	0.64	0.5263
nino3may	-95.2731	565.7343	-0.17	0.8666
nino3jun	-284.3973	382.2185	-0.74	0.4587
nino3jul	-505.1218	215.4651	-2.34	0.0212
nino3aug	-349.3153	861.2045	-0.41	0.6860
nino3sep	393.5581	679.4063	0.58	0.5638
nino3apr:nino3jun	-44.5111	29.1641	-1.53	0.1304
nino3apr:nino3aug	67.0019	38.3637	1.75	0.0841
nino3apr:nino3sep	-32.1606	23.7254	-1.36	0.1786
nino3may:nino3jun	56.0401	35.3552	1.59	0.1164
nino3may:nino3aug	-54.6980	37.3819	-1.46	0.1468
nino3jul:nino3sep	18.4828	8.1354	2.27	0.0254

Tabel 11: De resultaten voor de regressie voor Niño 3 met de interacties

Rest nu natuurlijk nog de regressie voor de stroming Niño 1.2. Dit levert een model op met een AIC van 988,47 met alleen nog de waarden voor augustus en september in de fit, zoals te zien is in tabel 12.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	38.5580	59.2413	0.65	0.5166
nino1.2aug	15.5232	6.4100	2.42	0.0172
nino1.2sep	-16.7727	7.0892	-2.37	0.0199

Tabel 12: De resultaten voor de regressie van Niño 1.2

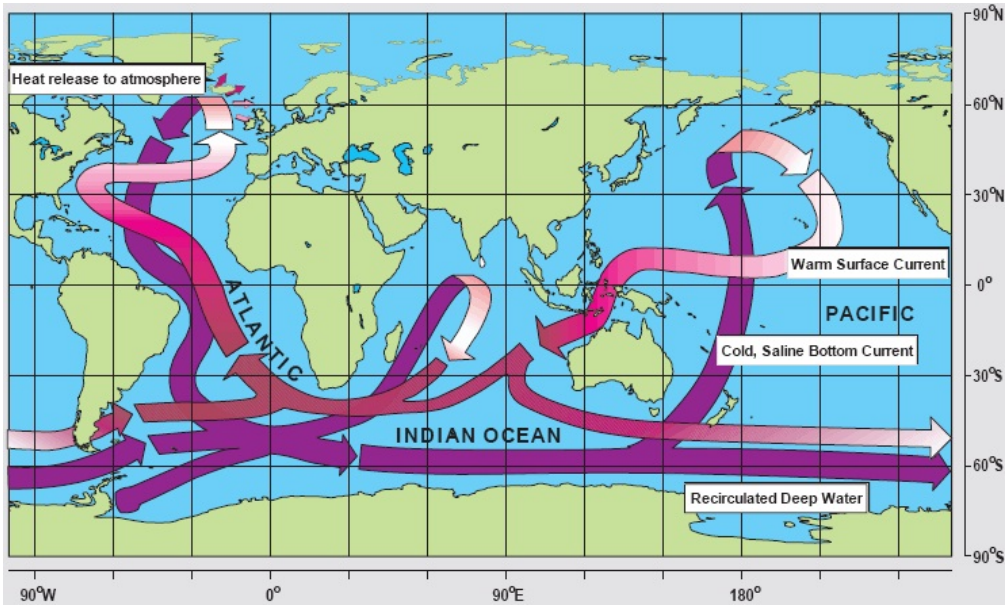
Ook nu kan er weer gekeken worden naar de interacties. De resultaten zijn te vinden in tabel 13, de AIC van dit model is 986,36.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	-1025.9182	1148.3043	-0.89	0.3739
nino1.2apr	195.7897	152.4625	1.28	0.2022
nino1.2may	-154.5561	125.7592	-1.23	0.2221
nino1.2jun	-488.5024	212.1563	-2.30	0.0235
nino1.2jul	556.4477	234.5400	2.37	0.0197
nino1.2aug	17.6635	10.5654	1.67	0.0979
nino1.2sep	-20.5693	7.1390	-2.88	0.0049
nino1.2apr:nino1.2jun	72.1451	25.1522	2.87	0.0051
nino1.2apr:nino1.2jul	-84.1936	24.7979	-3.40	0.0010
nino1.2may:nino1.2jun	-56.1250	21.5589	-2.60	0.0107
nino1.2may:nino1.2jul	65.7326	20.6258	3.19	0.0020

Tabel 13: De resultaten voor de regressie van Niño 1.2 met interacties

5.2.2 Noordzee, Oostzee en Atlantische Oceaan

Hierboven is gewerkt met maandtemperaturen van de Nino's. Hierdoor is het mogelijk om voor bepaalde maanden de temperatuur van de zee mee te nemen om zo de temperatuur van januari boven land te voorspellen. Deze aanpak lijkt heel logisch, maar waarom dan de zeestromingen zo ver weg? Er kan ook gekeken worden naar temperaturen in de Noordzee, Oostzee en de Atlantische Oceaan. Hiertoe zijn veel kaarten beschikbaar met kleuren die de temperatuur weergeven van bepaalde plekken in de zee. Deze kaarten zijn gemaakt van een grid met meetwaarden. Voor dit onderzoek is ook zo'n grid met meetwaarden gebruikt. Het is even puzzelen hoe het grid precies werkt, maar uiteindelijk blijkt dat de kolommen bestaan uit de breedtegraden die lopen van -88 tot 88 graden, waarbij -88 graden helemaal in het zuiden is en 88 graden is helemaal in het noorden. Het grid loopt per twee graden omhoog. De rijen lopen van 0 tot 358 graden, waarbij 0 graden de 0-meridiaan is die door Greenwich loopt en dan volgen de andere meridianen met de klok mee, dus eerst komen de meridianen met Oosterlengte en dan loopt het door naar de meridianen met Westerlengte. Dat betekent dat 358 graden de meridiaan is die op 2 graden Westerlengte loopt. Het grid loopt ook deze kant op per twee graden omhoog.
In één file staan waarden voor 10 jaren. De waarden lopen gewoon aan de onderkant door. Dat wil zeggen dat na de rij die de 358 graden van bijvoorbeeld 1900 weergeeft, de rij komt met de 0-meridiaan van het jaar 1901. Voor de stukken land zijn natuurlijk geen meetwaarden voor de SST. Hier wordt dan ook het cijfer -9999 neergezet. Door te kijken naar het verloop van de gebieden met -9999, kan worden gecontroleerd dat het grid inderdaad op de zojuist beschreven wijze is opgebouwd. De stukken land zijn te herkennen in het grid.
Natuurlijk kan er geen regressie gedaan worden op alle waarden uit het grid. Er zullen dus een paar waarden uitgekozen moeten worden om een regressie te kunnen doen. Om een verantwoorde keuze te kunnen maken, is er gekeken naar de kaart met alle zeestromingen in de wereld. Deze kaart is hieronder weergegeven (figuur 10).



Figuur 10: Een kaart met een schematische weergave van de zeestromingen over de wereld [9]

Het is te zien dat alle zeestromingen met elkaar verbonden zijn. In de Atlantische Oceaan loopt een stroming die het warme water vanuit de Indische en zelfs vanuit de Stille Oceaan aanvoert. Uiteindelijk lopen er ook nog kleine warme stroompjes richting de Noordzee. Voor het weer in Nederland is het voor de hand liggend om de temperaturen van de Noordzee, Oostzee en Atlantische Oceaan te nemen. Er is één punt in de Noordzee en één punt in de Oostzee gekozen. Dit is gedaan omdat in het grid deze zeeën maar klein zijn en dus is het nemen van meerdere punten niet zo zinvol. Er is gekozen voor ongeveer het middelste punt in de Noord- en Oostzee. Daarbij zijn er vier punten in de Atlantische Oceaan gekozen. Deze punten zijn zodanig gekozen dat het hele Noordelijke deel van de Atlantische Oceaan vertegenwoordigd wordt. Ook zal zo de stroming die in de Atlantische Oceaan loopt naar voren komen.
De punten die gekozen zijn:

- Voor de Noordzee: 58 graden Noorderbreedte en 4 graden Westerlengte. Dit komt overeen met het punt (2,74) in het grid (en hetzelfde punt een jaar later, daartoe moet er 180 graden bij het eerste getal opgeteld, dus (182,74), gevolgd door (362,74) enzovoorts).
- Voor de Oostzee: 58 graden Noorderbreedte en 22 graden Westerlengte. Dit komt overeen met het punt (11,74) in het grid.
- Voor de Atlantische Oceaan: 22° NB 14° OL, 30° NB 30° OL, 20° NB 50° OL en 10° ZB 20° OL. Deze punten komen overeen met de punten (173,66), (165,59), (155,54) en (170,39) in het grid.

Deze punten worden in R geselecteerd en dan kan er een regressie gedaan worden waarbij de punten voor de SST van de Noordzee, Oostzee en de Atlantische Oceaan meegenomen worden. Na stepAIC is het resultaat dat alleen de as-afsnede significant bijdraagt aan de fit, de AIC is dan 990,48. Dit betekent dus dat de jaarwaarden voor de sea surface temperature van de Noordzee, Oostzee en de Atlantische Oceaan niet bijdragen aan een goede fit voor de temperatuur in januari. Dat kan komen doordat er alleen waarden bekend zijn van de SST op jaarbasis. Waarschijnlijk zal een grid met maandwaarden voor de SST wel een goede fit opleveren.
Er kan worden gekeken of het meenemen van interacties wel een model oplevert, om zo toch in het overall-model een aantal waarden voor de zeestromingen dichterbij Nederland mee te nemen. Het meenemen van de interacties geeft inderdaad een model. De AIC van dit model is 997,49 en de resultaten zijn te zien in tabel 14. De AIC is groter dan voor het model met alleen het intercept, dus hier is weer sprake van een lokaal minimum in AIC.

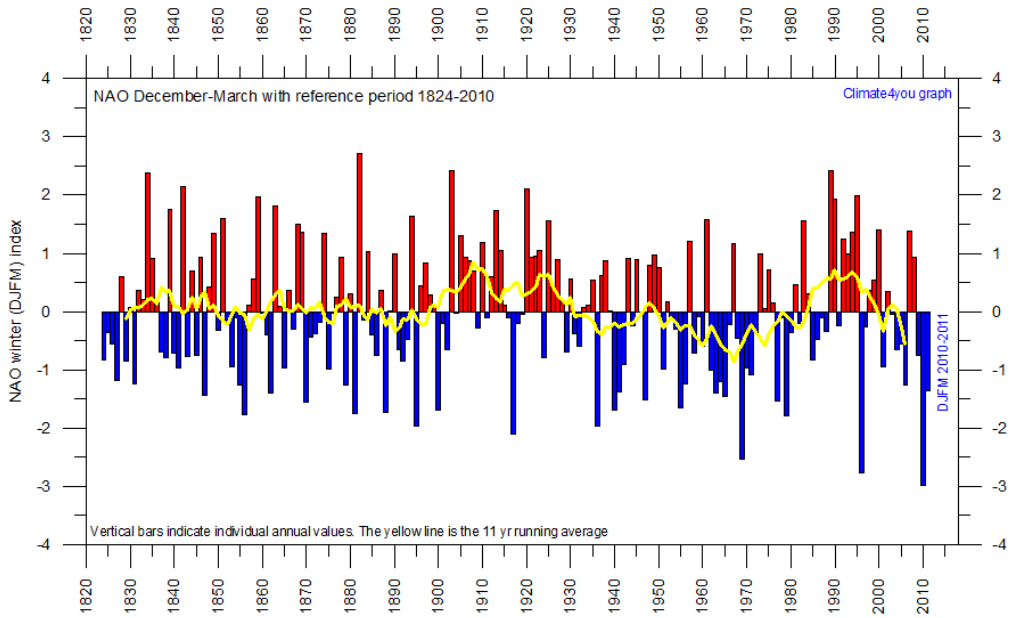
	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	452.2114	287.7255	1.57	0.1194
oostzee	0.1028	0.0379	2.71	0.0080
atlantisch1	-0.1795	0.1119	-1.60	0.1119
atlantisch2	-0.5971	0.3167	-1.89	0.0625
atlantisch3	0.3542	0.2930	1.21	0.2298
atlantisch4	-0.0896	0.0661	-1.36	0.1786
oostzee:atlantisch2	-0.0000	0.0000	-2.10	0.0386
atlantisch1:atlantisch2	0.0003	0.0002	1.98	0.0506
atlantisch1:atlantisch3	-0.0002	0.0002	-1.51	0.1353
atlantisch2:atlantisch4	0.0000	0.0000	2.12	0.0366

Tabel 14: De resultaten voor de fit met de SST's en hun interacties

Dit model is redelijk groot en alleen de waarde voor de Oostzee en de interactie tussen de temperaturen voor het tweede en vierde punt in de Atlantische Oceaan zijn significant.

5.3 Arctische Oscillatie

Voor de Arctische Oscillatie is een grafiek beschikbaar met de waarden van de gemiddelde NAO van december-maart. Deze grafiek is hieronder weergegeven (figuur 11).



Figuur 11: De meetwaarden voor de NAO in de periode december-maart

Helaas is er geen document beschikbaar met de getallen die gebruikt zijn om de grafiek te maken. Daarom is de grafiek vanaf 1906 afgelezen en ingevoerd in R. Dan is er weer een regressie mogelijk voor de temperatuur in januari als functie van de NAO-waarden. Deze fit levert een AIC op van 986,63. De resultaten staan in tabel 15. Het blijkt dat het intercept en de NAO beide significant zijn, de p-waarden zijn heel klein.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	16.1896	2.5433	6.37	0.0000
NAO	5.6831	2.3397	2.43	0.0169

Tabel 15: De resultaten van de regressie van januari als functie van de NAO-waarden

5.4 Modellen samenvoegen

Het idee is om een model te fitten dat afhankelijk is van alle mogelijke factoren die hierboven beschreven zijn. Natuurlijk kan het niet om alle waarden maar klakkeloos in R in te voeren en dan een model te fitten, daarvoor zijn er te veel variabelen. Het kan dan een oplossing zijn

om alle variabelen die in de voorgaande secties na stepAIC overgebleven zijn, in een model te stoppen en dan een fit te maken. Dit resulteert in een model met een AIC van 985,59. Hierop kan ook weer stepAIC worden toegepast. Dat geeft een model met een AIC van 954,22 en de variabelen die nog over zijn met hun regressiecoëfficiënten zijn te zien in tabel 16.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	7563.5926	2961.1651	2.55	0.0126
APR	4.5556	2.7914	1.63	0.1067
MAY	5.4836	2.5896	2.12	0.0374
JUN	-2.2128	0.9624	-2.30	0.0242
AUG	1.9049	1.6332	1.17	0.2470
SEP	3.3548	1.0373	3.23	0.0018
nino4jun	38.3796	10.8944	3.52	0.0007
nino3apr	18.4534	9.6800	1.91	0.0603
nino3may	-20.2679	14.1637	-1.43	0.1564
nino3jul	-388.4057	107.6659	-3.61	0.0005
nino3sep	-356.7092	111.1693	-3.21	0.0019
nino1.2apr	171.3989	133.9288	1.28	0.2044
nino1.2may	-129.0687	109.9223	-1.17	0.2439
nino1.2jun	-274.2007	191.0971	-1.43	0.1553
nino1.2jul	346.9482	208.2571	1.67	0.0997
nino1.2aug	13.7556	9.4181	1.46	0.1482
nino1.2sep	-26.6277	6.4708	-4.12	0.0001
NAO	4.7141	2.0947	2.25	0.0272
APR:MAY	-0.0296	0.0162	-1.82	0.0718
APR:JUN	0.0310	0.0129	2.41	0.0184
APR:SEP	-0.0387	0.0126	-3.08	0.0029
MAY:AUG	-0.0174	0.0138	-1.26	0.2114
nino3jul:nino3sep	13.5992	4.0758	3.34	0.0013
nino1.2apr:nino1.2jun	59.2087	22.9003	2.59	0.0116
nino1.2apr:nino1.2jul	-69.4333	22.5841	-3.07	0.0029
nino1.2may:nino1.2jun	-50.9662	19.7165	-2.58	0.0116
nino1.2may:nino1.2jul	58.6063	19.2210	3.05	0.0031

Tabel 16: De resultaten voor de regressie waarbij alle onafhankelijke variabelen die in voorgaande secties na stepAIC zijn overgebleven zijn meegenomen

Het mooiste zou nu natuurlijk zijn om ook de interacties mee te nemen. Helaas kan R hierop geen goede fit en stepAIC doen, het zijn te veel variabelen. Een fit is dan wel te maken, maar dan is het niet meer duidelijk wat de p-waarden zijn. Dat is niet wenselijk en dus is er besloten om de interacties te bepalen van het model wat hierboven gefit is. Ook dat bleek niet mogelijk, het zijn nog steeds te veel variabelen. Daarom is er gekeken naar bovenstaand model en daarvan zijn handmatig alle waarden die niet voldoen met een significantie voor $\alpha = 0,05$ verwijderd. Dat betekent dat er nog een fit gedaan zal worden met als onafhankelijke variabelen may, jun, sep, nino4jun, nino3jul, nino3sep, nino1.2sep, NAO, apr:jun, apr:sep, nino3jul:nino3sep, nino1.2apr:nino1.2jun, nino1.2apr:nino1.2jul, nino1.2may:nino1.2jun en nino1.2may:nino1.2jul. Ook dit zijn te veel variabelen en dus is degene met de grootste p-waarde geschrapt (dat is may) en daarna is uiteindelijk inderdaad een fit gedaan. Hierbij worden alle interacties meege-

nomen. Na stepAIC is er een model over met een AIC van 788,80. De resultaten van de fit staan weergegeven in tabel 17 en 18.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	-535727.9331	291923.3382	-1.84	0.0851
JUN	6164.4801	3519.1523	1.75	0.0990
SEP	-4184.6945	3342.6848	-1.25	0.2286
nino4jun	34121.3734	16376.2796	2.08	0.0536
nino3jul	16232.3402	10675.9996	1.52	0.1479
nino3sep	22462.4065	11314.5570	1.99	0.0645
nino1.2sep	-24433.7534	13416.8554	-1.82	0.0873
NAO	8248.3781	6157.1422	1.34	0.1991
JUN:APR	-79.2116	42.3862	-1.87	0.0801
JUN:nino4jun	20.6493	9.1296	2.26	0.0380
JUN:nino3jul	-241.4231	135.3760	-1.78	0.0935
JUN:nino3sep	-242.4762	136.1270	-1.78	0.0939
JUN:nino1.2sep	-7.3035	4.4747	-1.63	0.1222
JUN:NAO	-0.8798	1.2020	-0.73	0.4747
SEP:APR	64.8324	42.6764	1.52	0.1482
SEP:nino4jun	-23.6058	9.8422	-2.40	0.0290
SEP:nino3jul	170.8105	128.7858	1.33	0.2034
SEP:nino3sep	166.6643	129.4560	1.29	0.2163
SEP:nino1.2sep	7.9725	4.5025	1.77	0.0957
SEP:NAO	0.1655	1.2881	0.13	0.8994
nino4jun:nino3jul	-1115.3738	598.0770	-1.86	0.0806
nino4jun:nino3sep	-1376.7404	632.5206	-2.18	0.0448
nino4jun:NAO	28.1848	15.7282	1.79	0.0921
nino3jul:nino3sep	-727.8029	416.7833	-1.75	0.0999
nino3jul:nino1.2sep	913.3525	504.8127	1.81	0.0892
nino3jul:NAO	-337.7709	234.3343	-1.44	0.1688
nino3sep:nino1.2sep	931.1467	509.4168	1.83	0.0863
nino3sep:NAO	-290.1421	236.4337	-1.23	0.2375
nino1.2sep:NAO	-34.1814	12.6245	-2.71	0.0155
nino1.2apr:nino1.2jun	1474.5118	1480.8368	1.00	0.3342
nino1.2apr:nino1.2jul	-1457.7101	1602.3708	-0.91	0.3765
nino1.2jun:nino1.2may	800.2466	2011.6062	0.40	0.6960
nino1.2jul:nino1.2may	-712.9051	2096.7105	-0.34	0.7383
JUN:nino3jul:nino3sep	9.0021	5.0761	1.77	0.0952
JUN:nino1.2apr:nino1.2jun	-1.9030	1.9936	-0.95	0.3540
JUN:nino1.2apr:nino1.2jul	1.6852	2.0997	0.80	0.4340
JUN:nino1.2jun:nino1.2may	2.1314	2.1743	0.98	0.3416
JUN:nino1.2jul:nino1.2may	-2.0149	2.3425	-0.86	0.4024
SEP:nino3jul:nino3sep	-6.1785	4.8125	-1.28	0.2175
SEP:nino1.2apr:nino1.2jun	0.3198	0.2230	1.43	0.1708
SEP:nino1.2jun:nino1.2may	-0.4952	0.3075	-1.61	0.1268
SEP:nino1.2jul:nino1.2may	0.2320	0.2693	0.86	0.4016

Tabel 17: Regressie met interacties

	Estimate	Std. Error	t value	Pr(>| t |)
JUN:nino4jun:APR	-0.2734	0.1187	-2.30	0.0350
JUN:nino3jul:APR	3.1110	1.6298	1.91	0.0744
JUN:nino3sep:APR	3.1189	1.6435	1.90	0.0759
JUN:nino1.2sep:APR	0.0852	0.0548	1.55	0.1397
JUN:NAO:APR	0.0126	0.0151	0.84	0.4148
SEP:nino4jun:APR	0.2686	0.1237	2.17	0.0453
SEP:nino3jul:APR	-2.5689	1.6448	-1.56	0.1379
SEP:nino3sep:APR	-2.5660	1.6520	-1.55	0.1399
SEP:nino1.2sep:APR	-0.0964	0.0563	-1.71	0.1060
SEP:NAO:APR	-0.0096	0.0162	-0.59	0.5626
nino4jun:nino3jul:nino3sep	45.9724	23.1895	1.98	0.0649
nino4jun:nino1.2apr:nino1.2jun	-487.6351	110.4569	-4.41	0.0004
nino4jun:nino1.2apr:nino1.2jul	509.2861	115.0464	4.43	0.0004
nino4jun:nino1.2jun:nino1.2may	514.8151	117.4771	4.38	0.0005
nino4jun:nino1.2jul:nino1.2may	-537.5591	122.4082	-4.39	0.0005
nino3jul:nino1.2apr:nino1.2jun	199.4812	105.6115	1.89	0.0772
nino3jul:nino1.2apr:nino1.2jul	-208.2414	112.0005	-1.86	0.0815
nino3jul:nino1.2jun:nino1.2may	-295.9350	128.2789	-2.31	0.0348
nino3jul:nino1.2jul:nino1.2may	301.2658	133.1578	2.26	0.0379
nino3sep:nino1.2apr:nino1.2jun	66.4246	103.5657	0.64	0.5304
nino3sep:nino1.2apr:nino1.2jul	-76.2728	112.3430	-0.68	0.5069
nino3sep:nino1.2jun:nino1.2may	-159.7089	104.5555	-1.53	0.1462
nino3sep:nino1.2jul:nino1.2may	167.9253	109.5214	1.53	0.1447
nino3jul:nino3sep:nino1.2sep	-34.2131	19.1215	-1.79	0.0925
nino1.2sep:nino1.2apr:nino1.2jun	113.2435	53.6229	2.11	0.0508
nino1.2sep:nino1.2apr:nino1.2jul	-119.6536	55.9616	-2.14	0.0483
nino1.2sep:nino1.2jun:nino1.2may	-116.8207	56.4115	-2.07	0.0549
nino1.2sep:nino1.2jul:nino1.2may	122.7333	58.8275	2.09	0.0533
nino3jul:nino3sep:NAO	11.8015	8.7951	1.34	0.1984
nino1.2apr:nino1.2jun:nino1.2jul	-1.9124	0.6344	-3.01	0.0082
nino1.2apr:nino1.2jun:nino1.2may	-1.1395	0.7915	-1.44	0.1692
NAO:nino1.2apr:nino1.2jun	98.6008	26.8278	3.68	0.0020
nino1.2apr:nino1.2jul:nino1.2may	0.9200	0.8082	1.14	0.2717
NAO:nino1.2apr:nino1.2jul	-102.9783	28.2351	-3.65	0.0022
nino1.2jun:nino1.2jul:nino1.2may	1.7291	0.6601	2.62	0.0186
NAO:nino1.2jun:nino1.2may	-103.7875	28.1761	-3.68	0.0020
NAO:nino1.2jul:nino1.2may	108.6869	29.7093	3.66	0.0021
JUN:nino3jul:nino3sep:APR	-0.1157	0.0611	-1.89	0.0767
JUN:APR:nino1.2apr:nino1.2jun	0.0025	0.0026	0.95	0.3571
JUN:APR:nino1.2jun:nino1.2may	-0.0045	0.0035	-1.28	0.2172
JUN:APR:nino1.2jul:nino1.2may	0.0035	0.0031	1.13	0.2733
SEP:nino3jul:nino3sep:APR	0.0950	0.0615	1.54	0.1420
SEP:APR:nino1.2apr:nino1.2jun	-0.0034	0.0028	-1.21	0.2436
SEP:APR:nino1.2jun:nino1.2may	0.0052	0.0039	1.35	0.1968
SEP:APR:nino1.2jul:nino1.2may	-0.0028	0.0033	-0.85	0.4103
nino3jul:nino3sep:nino1.2apr:nino1.2jun	3.0537	1.1740	2.60	0.0193
nino3jul:nino3sep:nino1.2apr:nino1.2jul	-2.9332	1.2600	-2.33	0.0334

Tabel 18: Regressie met interacties (vervolg)

Dit is een hele lange lijst met variabelen. Als nu $\alpha = 0,05$ wordt gekozen, kan er een kleiner model worden gemaakt. Er kan worden gekeken wat de AIC wordt als er een model wordt gefit van alle variabelen die voldoen aan $\alpha = 0,05$. Dit model heeft een AIC van 981,24. Er kan echter ook weer gewoon stepAIC worden gedaan. Dat levert een model op met een AIC van 968,99, met de regressors die in tabel 19 staan.

	Estimate	Std. Error	t value	Pr(>| t |)
(Intercept)	-127.9898	35.5131	-3.60	0.0005
JUN:nino4jun	-0.0487	0.0246	-1.98	0.0506
nino4jun:SEP	0.0810	0.0283	2.86	0.0052
JUN:nino4jun:APR	0.0008	0.0003	2.54	0.0129
nino4jun:SEP:APR	-0.0009	0.0003	-2.52	0.0133
nino4jun:nino1.2apr:nino1.2jun	2.2745	0.5808	3.92	0.0002
nino4jun:nino1.2apr:nino1.2jul	-2.3430	0.6046	-3.88	0.0002
nino4jun:nino1.2jun:nino1.2may	-2.4089	0.6092	-3.95	0.0001
nino4jun:nino1.2jul:nino1.2may	2.5108	0.6368	3.94	0.0002
nino1.2apr:nino1.2jul:nino1.2sep	-0.0314	0.0093	-3.38	0.0010
nino1.2apr:nino1.2jun:NAO	0.0087	0.0038	2.32	0.0225

Tabel 19: Model na stepAIC met de meest significante waarden ($\alpha = 0,05$) uit tabel 17 en 18

Dit model is erg klein, maar heeft toch nog een redelijk lage AIC. Ook is het opvallend dat er alleen nog maar interacties in voorkomen, vooral tussen de Niño's.

5.5 LOOCV

In afgelopen paragrafen zijn modellen gemaakt door te selecteren op AIC. Het is eenzijdig om alleen naar de AIC te kijken en dus zal er ook nog een andere maat van nauwkeurigheid berekend worden voor alle modellen uit afgelopen paragrafen. Een relatief makkelijk te berekenen waarde is de Leave-One-Out cross-validation (LOOCV). Bij alle eerder gefitte modellen is een LOOCV-waarde berekend. Hiertoe is er voor iedere fit een matrix (X) gemaakt met de waarden voor de voorspellers erin en een rij enen om het intercept mee te nemen. Voor de interacties is weer de elementsgewijze vermenigvuldiging genomen. De matrix I bestaat uit de 105×105 identiteitsmatrix en de vector Jan is de vector met alle waarden voor de temperatuur in januari (deze vector is al eerder aangemaakt). De vector β bestaat uit de waarden van de regressiecoëfficiënten zoals die in voorgaande paragrafen zijn bepaald. In R kan nu de LOOCV-waarde worden berekend met de formule:

$$\text{sum}((\text{Jan} - \text{X\%} * \%\text{beta}) \wedge 2/\text{diag}(\text{I} - \text{X\%} * \%\text{solve}(\text{t(X)\%} * \%\text{X)\%} * \%\text{t(X)})).$$

De precieze R-scripts voor de afzonderlijke modellen zijn terug te vinden in de bijlage. Heel enkel gaf R aan dat de hat-matrix ($X(X^TX)^{-1}X^T$) niet berekend kon worden, omdat de matrix X^TX singulier was voor de berekeningen. Als die melding verschijnt, kan door gebruik te maken van de Cholesky decompositie toch een inverse worden berekend van X^TX en dus kan dan ook een waarde voor de LOOCV worden gevonden. In tabel 20 staan de waarden voor de AIC en de LOOCV van alle fits op een rijtje.

fit	AIC	LOOCV
eerste fit, temperatuur	985.82	108393.4
tweede fit, temperatuur met interacties	982.27	61228.75
derde fit, nino4	990.31	72863.37
vierde fit, nino4 met interacties	988.53	69999.19
vijfde fit, nino3.4	990.48	74658.02
zesde fit, nino3	990.96	73209.09
zevende fit, nino3 met interacties	997.63	70171.15
achtste fit, nino1.2	988.47	72103.78
negende fit, nino1.2 met interacties	986.36	64626.63
tiende fit, SST Noordzee, Oostzee en Atlantische Oceaan	990.48	74658.02
elfde fit, SST met interacties	997.49	72162,97
twaalfde fit, NAO	986.63	71217.8
overall-fit	954.22	42588.91
overall-fit met interacties	788.80	10653.19
ingekorte overall-fit met interacties	968.99	55207.40

Tabel 20: Alle gefitte modellen en hun AIC en LOOCV

Het valt op dat de overall-fit met interacties een erg lage AIC en een erg lage LOOCV heeft. Daarna volgt de overall-fit zonder interacties en dan de ingekorte overall-fit met interacties. Ook is het te zien dat over het algemeen de LOOCV laag is voor de modellen waarvoor de AIC ook laag is. Op basis van de LOOCV en de AIC zou de overall-fit met interacties het beste model zijn, gevolgd door de overall-fit zonder interacties.

5.6 Voorspellingen voor de gemiddelde temperatuur in januari 2012

Met alle modellen die in de voorgaande paragrafen zijn bepaald, kan een voorspelling gedaan worden voor de gemiddelde temperatuur in januari 2012. Hiertoe wordt er een vector gemaakt met alle waarden die van belang zijn voor de voorspelling van januari 2012. Zo worden de gemiddelde temperaturen van de voorgaande maanden in 2011 bepaald, de SST's van 2011 komen erin te staan en ook de NAO van 2011. Verder kunnen de interacties die nodig zijn voor de diverse modellen eenvoudig gemaakt worden door een vermenigvuldiging. Daarna wordt deze vector getransponeerd en vermenigvuldigd met de vector β die bestaat uit de regressiecoëfficieënten van de verschillende modellen. De resultaten van de voorspelling met de verschillende modellen staan in tabel 21. Daarbij is ook de variantie van de voorspelling geschat met behulp van (11).

fit	voorspelling voor jan 2012	SD (°C)
eerste fit, temperatuur	2.55 °C	2.65
tweede fit, temperatuur met interacties	1.67 °C	2.81
derde fit, nino4	2.17 °C	2.70
vierde fit, nino4 met interacties	2.17 °C	2.81
vijfde fit, nino3.4	1.63 °C	2.70
zesde fit, nino3	1.91 °C	2.71
zevende fit, nino3 met interacties	1.86 °C	2.76
achtste fit, nino1.2	1.32 °C	3.23
negende fit, nino1.2 met interacties	2.17 °C	2.64
tiende fit, SST Noordzee, Oostzee en Atlantische Oceaan	1.63 °C	2.71
elfde fit, SST met interacties	1.70 °C	2.80
twaalfde fit, NAO	0.77 °C	2.64
overall-fit	1.91 °C	2.73
overall-fit met interacties	4013.26 °C	139343.8
ingekorte overall-fit met interacties	3.09 °C	2.62

Tabel 21: De voorspelling voor de gemiddelde temperatuur in januari 2012 en de standaarddeviatie voor de verschillende fits

Het is duidelijk te zien dat de overall-fit met interactie een onredelijk antwoord geeft, meer dan 4000 graden in januari in Nederland kan natuurlijk niet. Het model van de overall-fit met interacties is erg groot. Het onredelijke antwoord zal dan ook kunnen komen door over-fitting. De rest van de voorspellingen zijn allemaal wel redelijk. Ter vergelijking; het gemiddelde van de gemiddelde temperaturen in januari van 1907 tot 2011 is 1,63 graden Celsius.

5.7 Conclusie; welk model is het beste op basis van standaard lineaire regressie?

Met behulp van lineaire regressie zijn diverse fits gedaan. Er zijn fits gemaakt waarbij alleen de temperatuur, alleen de SST en alleen de NAO is meegenomen. Ook zijn er fits gedaan waarin interacties meegenomen zijn. Met behulp van de AIC en de LOOCV is gekeken welke fit het beste is. De overall-fit waarbij ook de interacties meegenomen zijn, heeft de laagste waarden voor AIC en LOOCV, gevolgd door het overall-model waarbij de interacties niet zijn meegenomen. Daarna is er met ieder model een voorspelling gedaan voor de temperatuur in januari 2012. Hierbij kwam naar voren dat bijna alle modellen een redelijke voorspelling geven, met een standaarddeviatie rond de 3°C. Het overall-model met interacties heeft echter een hele rare voorspelling opgeleverd. Dat kan komen door over-fitting. Het ingekorte model met interacties (waarbij alle significante waarden uit het grote model zijn gebruikt om een model te fitten), geeft namelijk wel een redelijke voorspelling. Gezien het feit dat het overall-model met interacties de laagste AIC en LOOCV heeft, maar geen goede voorspelling oplevert, wordt er gekozen voor het overall-model zonder interacties als hoofdmodel. Dit model heeft na het model met interacties de laagste AIC en LOOCV. Daarbij vertoont het geen over-fitting in de voorspelling.

5.8 Wat zegt het eindmodel?

Het model waarvoor in voorgaande sessie is gekozen (de overall-fit), wordt gegeven door:
Jan = 7564 + 4.6 APR + 5.5 MAY - 2.2 JUN + 1.9 AUG + 3.4 SEP + 38.4 nino4jun + 18.5 nino3apr - 20.3 nino3may - 388.4 nino3jul - 356.7 nino3sep + 171.4 nino1.2apr - 129.1 nino1.2may

- 274.2 nino1.2jun + 346.9 nino1.2jul + 13.8 nino1.2aug - 26.6 nino1.2sep + 4.7 NAO - 0.03 APR*MAY + 0.03 APR*JUN - 0.04 APR*SEP - 0.02 MAY*AUG + 13.6 nino3jul:nino3sep + 59.2 nino1.2apr*nino1.2jun - 69.4 nino1.2apr*nino1.2jul - 51 nino1.2may*nino1.2jun + 58.6 nino1.2may*nino1.2jul.
Nu is het de vraag wat dit model zegt. Wat zou er gebeuren als bijvoorbeeld de temperatuur in april met een graad omhoog zou gaan? Dat zou te zien zijn aan het teken van de waarde van de β die bij de temperatuur in april hoort. Echter, april komt ook nog voor in allerlei interacties. Deze interacties hebben veelal een ander teken dan het teken van de β horend bij de temperatuur in april.
Om vrij eenvoudig toch iets te kunnen zeggen over het gevolg van een temperatuursverhoging in april, kan het gewoon worden uitgerekend wat de voorspelling is als april 1 graad warmer geweest zou zijn. Zo is dat voor alle andere maanden, de zeestromingen en de NAO (die met 1 eenheid omhoog gaat i.p.v. 1 graad) te doen. Dus door alle andere waarden gelijk te houden en alleen de waarde van één variabele aan te passen naar 1 graad warmer, kan de nieuwe voorspelling berekend worden. De resultaten hiervan zijn in onderstaande tabel weergegeven (tabel 5.8). Ter vergelijking; de voorspelling die het model geeft als er niets wordt veranderd is 1.91 ° Celsius.

1°C verandering in...	nieuwe voorspelling (°C)
APR	1.69
MAY	1.02
JUN	3.41
AUG	1.54
SEP	0.62
nino4jun	5.74
nino3apr	3.76
nino3may	-0.12
nino3jul	-1.36
nino3sep	3.08
nino1.2apr	2.52
nino1.2may	0.66
nino1.2jun	-0.24
nino1.2jul	4.14
nino1.2aug	3.29
nino1.2sep	-0.75
NAO	2.38

Tabel 22: De nieuwe voorspellingen als de variabelen los van elkaar 1°C (of 1 eenheid) omhoog gaan.

Het valt nu op dat als het warmer wordt in april, mei, augustus of september, dat het dan kouder zal worden in januari volgens het model. Als het warmer wordt in juni, dan voorspelt het model een warmere januari. Juni komt niet voor, dus een verandering in de temperatuur van juni heeft geen invloed op de temperatuur van januari.
Daarnaast is het te zien dat als de stroming Niño 3 in mei of juli of de Niño 1.2 in mei, juni of september warmer wordt, januari kouder zal worden. Echter, als de Niño 4 in juni, de Niño 3 in april of september of de Niño 1.2 in april, juli of augustus warmer wordt, zal januari ook warmer worden. Daarnaast is het te zien dat een verandering in de temperatuur van de stroming Niño 4 in juni een grote verandering in de voorspelling veroorzaakt. In het algemeen blijkt het ook

dat veranderingen in de SST met 1°C meer invloed hebben dan veranderingen in de maandtemperaturen met 1°C. Tenslotte zegt het model dat een verhoging van de NAO zal leiden tot een warmere januari.
Al met al valt het op dat als juni koud is, dat dan ook januari koud zal zijn. Voor de andere maanden geldt dit echter niet. Ook hebben de zeestromingen een grote invloed in de uiteindelijke voorspelling voor de temperatuur in januari. Dit zou dus betekenen dat de kranten wat kort door de bocht zijn geweest. Een koude juni is inderdaad het geval, maar er zijn meer koude wintermaanden geweest. Dit kan het effect van de koude juni weer opheffen.

6 Verbeteringen in het model door andere regressie?

6.1 Mixed models

Tot nog toe is er alleen gewerkt met gewone lineaire regressie. Dit betekent dat de fouten tot nog toe onafhankelijk werden gezien. Er kan echter ook een correlatie meegenomen worden in de fout van het ene jaar, die doorwerkt in het volgende jaar. Op welke manier dat gebeurt en hoe lang de fout doorwerkt, hangt af van het type mixed model dat je neemt. In dit onderzoek zal er voor drie types mixed models een fit gedaan worden. De bedoeling is om te kijken of het beste model wat in afgelopen paragrafen gevonden is, verbeterd kan worden door het meenemen van de correlatie tussen de fouten.
Het eerste type correlatie dat bekeken wordt, is de AR1 (autoregressive order 1). Voor de voorspelling van y_t wordt gewerkt met het model $y_t = f(t) + \epsilon_t$, met $f(t)$ een functie en ϵ_t de fout. Bij AR1 wordt er gewerkt met een correlatie tussen de fout ϵ op tijdstip t en de fout ϵ op tijdstip $t+1$. De correlatie hiertussen wordt weergegeven met $\text{Cor}(\epsilon_t, \epsilon_{t+1}) = \phi_1$.
Het tweede type is de MA1 (moving average order 1). Hierbij wordt het model voor de voorspelling gegeven door $y_t = f(t) + \epsilon_t + \theta_1\epsilon_{t-1}$, met nu ϵ_t de onbekende fout uit het model voor tijdstip t en ϵ_{t-1} de onbekende fout uit een tijdstip eerder. Via de grootte van θ wordt de invloed van de fout uit eerdere tijdstippen meegenomen. De correlatie tussen de voorspellingen y_t en y_{t-1} wordt gegeven door

$$\text{Cor}(y_t, y_{t-1}) = \text{Cor}(\epsilon_t + \theta_1\epsilon_{t-1}, \epsilon_{t-1} + \theta_1\epsilon_{t-2}) = \theta_1.$$

Het laatste model wat bekeken wordt, is het ARMA(1,1) (autoregressive moving average process order 1,1) model. Hierbij worden eigenlijk bovenstaande twee modellen gecombineerd. De voorspelling voor y_t wordt weer gegeven door $y_t = f(t) + \epsilon_t + \theta_1\epsilon_{t-1}$, maar nu zijn ϵ_t en ϵ_{t-1} gecorreleerd volgens $\text{Cor}(\epsilon_t, \epsilon_{t-1}) = \phi_1$. De correlatie tussen y_t en y_{t-1} is nu gegeven door

$$\text{Cor}(y_t, y_{t-1}) = \text{Cor}(\epsilon_t + \theta_1\epsilon_{t-1}, \epsilon_{t-1} + \theta_1\epsilon_{t-2}) = \phi_1 + \phi_1^2 + \theta_1 + \theta_1 + \theta_1^2\phi_1.$$

Uit voorgaande paragrafen is gebleken dat het overall-model zonder de interacties het beste is om een voorspelling mee te doen. Voor dit model zal dan ook een eventuele verbetering met behulp van de mixed models gezocht worden. De resultaten van de verschillende regressiemethodes staan in tabel 23. Hierbij dient de opmerking gemaakt te worden dat er twee varianten van AIC te vinden zijn. Dat komt doordat er bij het fitten van een mixed model wordt gewerkt met de methode REML, wat bij de modellen met alleen fixed effects niet het geval is. Om deze verschillende modellen toch te kunnen vergelijken, wordt er handmatig ingesteld om niet met REML te werken (maar met ML) en dan is de AIC van de mixed models direct te vergelijken

met de fixed models. Het beste model zal gekozen worden door te kijken naar de AIC op basis van REML en om dit model te kunnen vergelijken met de modellen met alleen fixed effects, zal de bijbehorende AIC met ML bepaald worden.

fit	resultaat	AIC met REML	AIC met ML
AR1	$\phi_1 = 0.06482108$	908.48	958.18
MA1	$\theta_1 = 0.07442376$	908.45	958.15
ARMA1	$\phi_1 = -0.4392836$, $\theta_1 = 0.5063045$	910.40	959.90

Tabel 23: De mixed model fits voor het overall-model uit voorgaande paragrafen

Uit bovenstaande tabel blijkt dat het model met de MA1-structuur het beste resultaat oplevert, de AIC op basis van REML is het laagst. Om nu het mixed model te kunnen vergelijken met het model met alleen fixed effects, wordt gekeken naar de AIC op basis van ML. Dit is 958,15. De AIC met alleen de fixed effects, dus de AIC die is verkregen na alleen stepAIC, is 954,22. De AIC van het mixed model is dus hoger. Dat betekent dat het toevoegen van de random effecten geen positieve invloed op het model heeft. Het model met alleen fixed effects is beter dan het mixed model.

6.2 Ridge regressie

Ridge regressie kan in R uitgevoerd worden met de functie lm.ridge onder de library MASS. Voor het overall-model kan worden gekeken of er met Ridge regressie een beter model gemaakt kan worden dan met stepAIC gedaan is.
Bij Ridge regressie is het de truc om te bepalen welke waarde voor λ de meest geschikte is. Dat kan door voor een heel aantal waarden van λ de AIC te berekenen. Op die manier kan de meest optimale waarde voor λ gevonden worden en hopelijk levert dat een model op met een lagere AIC dan voor het simpele model is verkregen.
Om de waarde van λ te kunnen bepalen, moet er wel een AIC bepaald kunnen worden. Dat kan door de gemaximaliseerde log-likelihood te bepalen en het aantal vrijheidsgraden. Dan kan de AIC berekend worden door -2 $\times$ maximized log-likelihood + 2 $\times$ (aantal vrijheidsgraden + 2) voor het model met een intercept, zoals hier het geval is. Voor het gewone lineaire model, dus $\lambda = 0$, is het aantal vrijheidsgraden gelijk aan het aantal parameters in het model, zoals zo meteen ook zal worden bewezen.
Om de log-likelihood te maximaliseren, moet er een waarde voor $\hat{\beta}$ bepaald worden waarbij

$$||y - X\beta||^2 + \lambda||\beta||^2$$

minimaal is. Dat is hetzelfde als het minimaliseren van

$$(y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta.$$

Dit is niets anders dan

$$y^Ty - 2\beta^TX^Ty + \beta^T(X^TX + \lambda I)\beta.$$

Hiervan kan een afgeleide genomen worden en deze kan gelijk gesteld worden aan nul. Dat levert dan op

$$-2X^Ty + 2(X^TX + \lambda I)\beta = 0$$
$$\hat{\beta} = (X^TX + \lambda I)^{-1}X^Ty.$$

Vervolgens kan er gekeken worden naar de log-likelihood bij een vastgestelde λ. Deze is maximaal voor $\hat{\beta}$ bij die λ en dus kan dat gebruikt worden in de bepaling van de AIC. Voor een vastgestelde waarde van λ is de log-likelihood als functie van die λ gegeven door:

$$l_\lambda = \log(\prod_{i=1}^{n} p(y_i|\hat{\beta_\lambda}, \sigma)$$

$$= \sum_{i=1}^{n} \log(\frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{1}{2\sigma^2}(y_i - x_i^T\hat{\beta_\lambda})^2})$$

$$= -\frac{n}{2}\log(2\pi) - n\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x_i^T\hat{\beta_\lambda})^2.$$

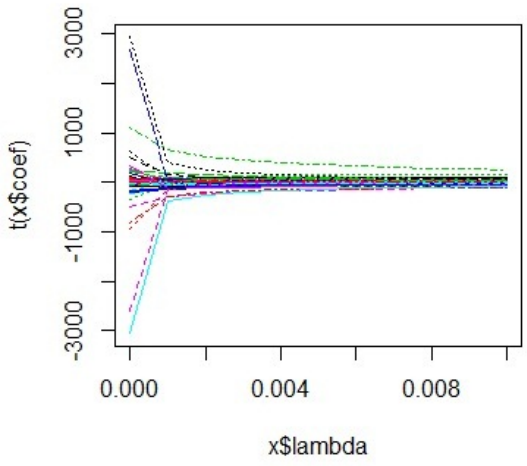
Als dit in vectornotatie wordt weergegeven en als dan $\hat{\beta}_\lambda$ wordt ingevuld, is dit gelijk aan

$$-\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}(y - X(X^TX + \lambda I)^{-1}X^Ty)^T(y - X(X^TX + \lambda I)^{-1}X^Ty).$$

Nu is er alleen nog de vraag wat σ zou moeten zijn. In hoofdstuk 2.5.2 (het deel over de AIC voor het gewone lineaire model) is genomen $\hat{\sigma}^2 = \frac{n-p}{n}\sigma^2$, waarbij σ^2 gelijk is aan de ordinary least squares variance estimator, dus $\sigma^2 = \frac{1}{n-p}(y - X\hat{\beta})^T(y - X\hat{\beta})$. Voor de Ridge regressie wordt dezelfde uitdrukking voor $\hat{\sigma}^2$ gebruikt, maar dan natuurlijk met $\hat{\beta} = (X^TX + \lambda I)^{-1}X^Ty$. Het aantal vrijheidsgraden voor een lineaire afbeelding van een variabele op zichzelf via $\tilde{y} = S_\lambda y$ is gelijk aan de trace van de matrix S_λ. In dit onderzoek wordt er gekeken naar een lineaire afbeelding van y op zichzelf via $\tilde{y} = X(X^TX + \lambda I)^{-1}X^Ty$. Dat betekent dat het aantal vrijheidsgraden in dit geval gegeven wordt door de trace van de matrix $X(X^TX + \lambda I)^{-1}X^T$. Dat kan even gecontroleerd worden voor het geval dat $\lambda = 0$. In dat geval is de verwachting dat het aantal vrijheidsgraden gelijk is aan het aantal parameters p uit het model. De rekenregels zeggen Tr(AB)=Tr(BA) als deze vermenigvuldiging mogelijk is. Dan volgt dus heel simpel:

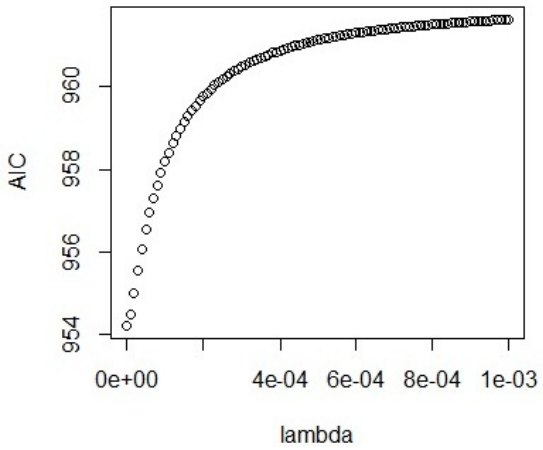
$$Tr(X(X^TX + \lambda I)^{-1}X^T) = Tr((X^TX)^{-1}X^TX) = Tr(I_{p\times p}) = p.$$

Nu is het mogelijk om de AIC voor verschillende waarden van λ te bepalen door -2 $\times$ maximized log-likelihood + 2 $\times$ (aantal vrijheidsgraden + 2) te berekenen. Dit kan gedaan worden in een for-loop in R. Om een beetje een inschatting te kunnen maken van de grootte van λ is in figuur 12 een plotje gemaakt voor de regressie-coëfficiënten als functie van λ.



Figuur 12: Plot van de regressiecoëfficiënten bij verschillende waarden voor λ

Het is in deze plot te zien dat een kleine waarde van λ al voldoende is om de waarden van β kleiner te maken en uiteindelijk richting 0 te brengen. De waarden van λ tussen 0 en 0,001 zullen al voldoende zijn om in de for-loop te berekenen.
In R kan alles worden uitgevoerd, met als resultaat de grafiek uit figuur 13.



Figuur 13: Plot van de AIC bij verschillende waarden voor λ

In bovenstaande grafiek is duidelijk te zien dat de AIC voor het geval $\lambda = 0$ het kleinste is. Dat betekent dat het doen van Ridge-regressie met een waarde voor $\lambda > 0$ geen positief effect heeft op het model. Het model zal niet verbeteren met Ridge regressie. Het zal alleen maar slechter worden. Het model wat verkregen is na stepAIC is dus nog steeds het beste model.

7 Conclusie

De bedoeling van deze scriptie is om een voorspelling voor minimaal drie maanden vooruit te doen voor de gemiddelde temperatuur van januari 2012. Dat betekent dat alle datasets tot en met september gebruikt mogen worden in de voorspelling. Om uiteindelijk een model te kunnen krijgen, is er gekeken naar de theorie achter lineaire regressie.
Daarna is er een simpel model gefit op basis van lineaire regressie waarin alleen de temperaturen van de maanden april tot en met september zijn meegenomen. Dit model is uitgebreid met interacties tussen de maanden. Ook is er op gelijke wijze een model gefit voor de Sea Surface Temperature van de diverse stromingen van El Niño en later van de Noordzee, Oostzee en Atlantische Oceaan. Ook is er een fit gedaan voor de temperatuur in januari 2012 als functie van de NAO, een index die de luchtstroming van de polen naar Nederland en andersom weergeeft.
Al deze simpele modelletjes zijn gecombineerd tot één model. De AIC van dit model is 954,22. De AIC geeft eigenlijk aan hoe goed een model is, hoe lager hoe beter. Aan dit gecombineerde model zijn ook de interacties nog toegevoegd. Dat leverde een AIC op van 788,80. Echter, dit model bleek last te hebben van over-fitting. De voorspelling van dit model was waardeloos. Daarom is er gekozen voor het eerste gecombineerde model. Dat model heeft een redelijk lage AIC en LOOCV (ook een maat van hoe goed het model is) en dus is daar een voorspelling mee te doen. De voorspelling op basis van het beste model is dat het 1,91 graden Celsius wordt in januari 2012. De standaarddeviatie is 2,73 graden. Verder bleek uit dit model dat een koude juni ook een koude januari tot gevolg zou hebben, maar voor de andere maanden geldt een omgekeerd effect. Daarnaast hebben de zeestromingen een grote invloed op de temperatuur in januari.
Er is geprobeerd om het model beter te maken door random effecten mee te nemen en daarmee de correlatie tussen de fouten in verschillende jaren. Dit leverde geen verbetering op. Ook is er geprobeerd om Ridge regressie te doen. Dat leverde ook geen verbetering voor het model. Daarom is dus op basis van het eerste gecombineerde model op basis van lineaire regressie de voorspelling voor de gemiddelde temperatuur in januari 2012 gezet op 1,91 graden Celsius. Dat is ongeveer gelijk aan het gemiddelde van 1907-2011 (1,63 graden). De standaarddeviatie is 2,73 graden.

8 Nawoord

Het is inmiddels begin februari, wat betekent dat de echte waarden van de maand januari ook te vinden zijn. De gemiddelde maandtemperatuur was 3,7 graden Celsius. Dat ligt welliswaar iets boven de voorspelling en ook iets boven het gemiddelde van de maandtemperaturen, maar toch nog binnen de foutmarge. Het dus lang niet zo'n horrorwinter als over gespeculeerd werd. Het is zelfs boven gemiddeld warm geworden.
Een nadeel van mijn voorspelling is natuurlijk dat de variantie groot is. Daardoor is er geen betrouwbare voorspelling te doen. Dat heeft de Telegraaf echter ook al gezegd. Langetermijnverwachtingen zijn lastig. Als de onzekerheid groot is, is de voorspelling ook minder zinvol. Zo dus ook mijn voorspelling. Maar aan de andere kant, als het weer met een simpele lineaire regressie goed te voorspellen was, dan zou het ook geen leuk gespreksonderwerp meer zijn. Bovendien zouden de weerkundigen niet zo veel aandacht krijgen en zou de "Oant Moarn"van Piet Paulusma misschien wel helemaal niet bekend zijn in Nederland.
De voorspelling is dan misschien niet helemaal geweldig, dat betekent niet dat ik er niets van heb geleerd. De lineaire regressie, het werken met R, het zelf onderzoek doen en de problemen en leuke dingen die je daarbij tegenkomt heb ik allemaal mogen ondervinden. Vooral in R kwam het nog wel eens voor dat ik grote delen met de hand had uitgerekend, terwijl ik dat gewoon in R had kunnen opvragen. In het script uit de bijlage is het waarschijnlijk ook wel te zien dat ik

efficiënter ben gaan rekenen, ik laat R meer doen.
Verder wil ik natuurlijk Ernst Wit hartelijk bedanken voor zijn hulp! Hij was altijd bereid om te helpen. Een mailtje als ik vastliep en bijna iedere week een afspraak om even naar de stand van zaken te kijken. Ook was hij altijd meedenkend en positief. Hartelijk dank daarvoor, het heeft me enorm geholpen.

9 Literatuur

1 Linear Regression, Lecture notes in statistics (voornamelijk hoofdstuk 1,2 en 6), Jürgen Groß, Springer-Verlag Berlin Heidelberg 2003

2 A first course in Bayesian statistical methods, hoofdstuk 9, Peter D. Hoff, Springer Science+Business Media, LLC 2009

3 C, Cross-Validation, Payam Refaeilzadeh, Lei Tang, Huan Lu, Arizona State University (artikel)

4 Anthony G. Barnston, Linear Statistical Short-Term Climate Predictive Skill in the Northern Hemisphere, Journal of Climate, Vol. 7, p. 1513-1564

5 John A. Knaff, Predicting Summertime Carribean Pressure in Early April, Weather and forecasting, Vol. 13, p. 740-752

6 John E. Freuds' Mathematical Statistics with applications, hoofdstuk 10, Miller and Miller, 7th edition, Pearson Prentice Hall

7 Hoofdstuk 4, meervoudige lineaire regressie, collegedictaat van de Technische Universiteit Eindhoven

8 Hoofdstuk 3, meervoudige lineaire regressie, collegedictaat van de Technische Universiteit Eindhoven

9 een onderwijsproject voor aardrijkskunde in België, online bekeken op aardrijkskunde.dbz.be op 7 januari 2012

10 cran.r-project.org, appendix mixed models

11 een handout over AICc van de New York University
online gelezen op http://people.stern.nyu.edu/churvich/Forecasting/Handouts/AICC.pdf op 12 december 2011

12 site van de Climate Prediction Centre van de noaa
www.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostuff, bezocht op 3 januari 2011

13 Brochure van het KNMI: De staat van het klimaat 2010
online gelezen op knmi.nl/bibliotheek/klimaatbrochures op 3 januari 2011

14 site over onder andere AO, site van de noaa
www.arctic.noaa.gov/essay_bond.html, bezocht op 8 januari 2011

15 WAB Zon en Klimaat, rapport van het KNMI en NIOZ, rapport 500102001, online gelezen via klimaatportaal.nl op 4 januari 2012

16 site van University Corporation for Atmospheric Research (UCAR)
www.cgd.ucar.edu/cas/cataLog/climind/TNI_N34/index.html, bezocht op 7 januari 2012

17 college-dictaat van de universiteit van Purdue, online bekeken op 25 januari 2012,
http://www.stat.purdue.edu/ boli/stat512/lectures/topic5a.pdf

18 Thesis van Paul Komarek, online bekeken op 25 januari 2012,
http://komarix.org/ac/papers/thesis/thesis_html

19 Telegraaf online, bezocht op 27 oktober 2011,
http://www.telegraaf.nl/binnenland/10802168/_Horrorwinter_niet_te_voorspellen_.html?sn=binnenland

Herkomst van de data

- *Gemiddelde maandtemperaturen:* KNMI, de waarden voor het meetstation Eelde zijn gebruikt
- *SST-data:* NOAA, de waarden voor El Niño zijn te vinden op
ftp://ftp.ncdc.noaa.gov/pub/data/cmb/ersst/v3b/pdo/, bezocht op 22 december 2011
de datagrid is te vinden op
ftp://ftp.ncdc.noaa.gov/pub/data/cmb/ersst/v3b/ascii/, bezocht op 22 december 2011
- *NAO:* Climate4you,
http://www.climate4you.com/AirPressureVariations.htm# NAO_uncorrected, bezocht op 6 januari 2012

10 Bijlagen

10.1 Bijlage 1: R-script van de voorbeelden uit hoofdstuk 2

10.2 Bijlage 2: R-script van de berekeningen uit hoofdstuk 5 en 6

Bijlage 1: Het R-script voor de voorbeelden uit hoofdstuk 2

```
## MLE bepalen van het voorbeeld van de oppervlaktespanning (hst 2.2.2)
data=c(1,1,1,1,1,1,1,1,1,1,1,1,1,40,57,69.7,70,80,80,100,100,110,120,140,160,180,200)
X=matrix(data,nrow=14,ncol=2)
y=c(41.6,39.5,38.3,37.8,37.2,36.6,35,34.4,33.2,32.3,30.4,27.9,25.7,23.3)
leastsquares=solve(t(X)%*%X)%*%t(X)%*%y
x=c(40,57,69.7,70,80,80,100,100,110,120,140,160,180,200)
fit=lm(y~x)
plot(x,y)
abline(45.9629122,-0.1130157,col='red')

## Beta bepalen voor het voorbeeld van de yield (hst 2.2.2)
T=c(50,50,50,50,50,50,125,125,125,125,125,125,125,125,200,200,200,200,200,200)
P=c(1,1,5.5,5.5,10,10,1,1,5.5,5.5,5.5,5.5,10,10,1,1,5.5,5.5,10,10)
Y=c(68,65,81,82,57,65,52,53,82,85,80,83,76,76,28,27,70,68,80,84)
fit2=lm(Y~T+P+T:P)

## Bayesiaanse benadering: voorbeeld van de oppervlaktespanning Bayesiaans benaderd
## met de g-prior (herkomst van de g-prior: [2]) (hst 2.2.3)
library(mvtnorm)
# g-prior uit [2]
lm.gprior<-function(y,X,g=dim(X)[1],nu0=1,s20=try(summary(lm(y~-
1+X))$sigma^2,silent=TRUE),S=1000)
{
  n<-dim(X)[1] ; p<-dim(X)[2]
  Hg<- (g/(g+1)) * X%*%solve(t(X)%*%X)%*%t(X)
  SSRg<- t(y)%*%( diag(1,nrow=n)  - Hg ) %*%y

  s2<-1/rgamma(S, (nu0+n)/2, (nu0*s20+SSRg)/2 )

  Vb<- g*solve(t(X)%*%X)/(g+1)
  Eb<- Vb%*%t(X)%*%y

  beta<-rmvnorm(S, Eb,Vb)

  list(beta=beta,s2=s2)
}

# prior parameters
g=n; nu.0=2; sigma2.0=1

XtX<-t(X)%*%X
n=nrow(X)
p=ncol(X)
```

```
nu.n<- nu.0+n

# store mcmc samples in these objects
S=5000
beta.post<-matrix(nrow=S,ncol=p)
sigma2.post<-rep(NA,S)

# Monte Carlo for posterior of beta and sigma2
l=lm.gprior(y,X,nu0=2,s20=1,S=1000)
beta.post=l$beta
sigma2.post=l$s2

mean(beta.post);
mean(sigma2.post);

# Plot van de beta's
library(KernSmooth)
beta0=beta.post[,1]
beta1=beta.post[,2]
plot(bkde(beta0))
plot(bkde(beta1))

## hat-matrix voor voorbeeld van oppervlaktespanning:
p<-X%*%solve(t(X)%*%X)%*%t(X)

## AIC's van het voorbeeld van de yield (hst 2.5.2)
stepAIC(lm(Y~P+T+P:T))
AIC(lm(Y~P+T+P:T))
AIC(stepAIC(lm(Y~P+T+P:T)))
AIC(lm(Y~P+T))
AIC(stepAIC(lm(Y~P+T)))

## LOOCV voor het voorbeeld van de yield (hst 2.5.3)
data2=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,50,50,50,50,50,50,125,125,125,125,125,125,125,125,2
00,200,200,200,200,200,200,1,1,5.5,5.5,10,10,
1,1,5.5,5.5,5.5,5.5,10,10,1,1,5.5,5.5,10,10,50,50,275,275,500,500,125,125,687.5,
687.5,687.5,1250,1250,200,200,1100,1100,2000,2000)
X=matrix(data2,nrow=20,ncol=4)
colnames(X)<-c("intercept","T","P","T:P")
beta<-c(92.35926,-0.31222,-2.87037,0.04444)
y<-Y
I<-diag(1,20)
sum((y-X%*%beta)^2/diag(I-X%*%solve(t(X)%*%X)%*%t(X)))
```

```
## p-waarden en plot van de residuen voor het voorbeeld van de yield (hst 2.5.4)
summary(fit2)
res<-residuals(fit2)
plot(res)
plot(T,res)
plot(P,res)

## p-waarden en plot van residuen met kwadratische termen
## voor het voorbeeld van de yield (hst 2.5.4)
T2=c(50^2,50^2,50^2,50^2,50^2,50^2,125^2,125^2,125^2,125^2,125^2,125^2,125^2,125^2,200^2,
200^2,200^2,200^2,200^2,200^2)
P2=c(1,1,30.25,30.25,100,100,1,1,30.25,30.25,30.25,30.25,100,100,1,1,30.25,30.25,100,100)

fit3=lm(Y~P+T+P:T+P2+T2)
summary(fit3)

par(mfrow=c(1,3))
plot(residuals(fit3))
plot(T,residuals(fit3))
plot(P,residuals(fit3))

## AIC's vergelijken
AIC(fit2)
AIC(fit3)
```

Bijlage 2: R-script voor de berekeningen van hoofdstuk 5 en 6

```
E<-read.table('sstelnino',col.names=c("year","month","nino4","nino3.4","nino3","nino1.2"))
F<-E$nino4
nino4<-F[73:1343]
nino4apr=nino4[seq(from=4,to=1260,by=12)]
nino4may=nino4[seq(from=5,to=1260,by=12)]
nino4jun=nino4[seq(from=6,to=1260,by=12)]
nino4jul=nino4[seq(from=7,to=1260,by=12)]
nino4aug=nino4[seq(from=8,to=1260,by=12)]
nino4sep=nino4[seq(from=9,to=1260,by=12)]
G<-E$nino3.4
nino3.4<-G[73:1343]
nino3.4apr=nino3.4[seq(from=4,to=1260,by=12)]
nino3.4may=nino3.4[seq(from=5,to=1260,by=12)]
nino3.4jun=nino3.4[seq(from=6,to=1260,by=12)]
nino3.4jul=nino3.4[seq(from=7,to=1260,by=12)]
nino3.4aug=nino3.4[seq(from=8,to=1260,by=12)]
nino3.4sep=nino3.4[seq(from=9,to=1260,by=12)]
H<-E$nino3
nino3<-H[73:1343]
nino3apr=nino3[seq(from=4,to=1260,by=12)]
nino3may=nino3[seq(from=5,to=1260,by=12)]
nino3jun=nino3[seq(from=6,to=1260,by=12)]
nino3jul=nino3[seq(from=7,to=1260,by=12)]
nino3aug=nino3[seq(from=8,to=1260,by=12)]
nino3sep=nino3[seq(from=9,to=1260,by=12)]
J<-E$nino1.2
nino1.2<-J[73:1343]
nino1.2apr=nino1.2[seq(from=4,to=1260,by=12)]
nino1.2may=nino1.2[seq(from=5,to=1260,by=12)]
nino1.2jun=nino1.2[seq(from=6,to=1260,by=12)]
nino1.2jul=nino1.2[seq(from=7,to=1260,by=12)]
nino1.2aug=nino1.2[seq(from=8,to=1260,by=12)]
nino1.2sep=nino1.2[seq(from=9,to=1260,by=12)]

D<-read.csv("maanddata1")
Data1<-D[0:105,6:11]
Jan<-D[2:106,3]

Data1$nino4apr<-with(Data1,nino4apr)
Data1$nino4may<-with(Data1,nino4may)
Data1$nino4jun<-with(Data1,nino4jun)
Data1$nino4jul<-with(Data1,nino4jul)
Data1$nino4aug<-with(Data1,nino4aug)
```

```
Data1$nino4sep<-with(Data1,nino4sep)

Data1$nino3.4apr<-with(Data1,nino3.4apr)
Data1$nino3.4may<-with(Data1,nino3.4may)
Data1$nino3.4jun<-with(Data1,nino3.4jun)
Data1$nino3.4jul<-with(Data1,nino3.4jul)
Data1$nino3.4aug<-with(Data1,nino3.4aug)
Data1$nino3.4sep<-with(Data1,nino3.4sep)

Data1$nino3apr<-with(Data1,nino3apr)
Data1$nino3may<-with(Data1,nino3may)
Data1$nino3jun<-with(Data1,nino3jun)
Data1$nino3jul<-with(Data1,nino3jul)
Data1$nino3aug<-with(Data1,nino3aug)
Data1$nino3sep<-with(Data1,nino3sep)

Data1$nino1.2apr<-with(Data1,nino1.2apr)
Data1$nino1.2may<-with(Data1,nino1.2may)
Data1$nino1.2jun<-with(Data1,nino1.2jun)
Data1$nino1.2jul<-with(Data1,nino1.2jul)
Data1$nino1.2aug<-with(Data1,nino1.2aug)
Data1$nino1.2sep<-with(Data1,nino1.2sep)

K<-read.table("19001909")
L<-read.table("19101919")
M<-read.table("19201929")
N<-read.table("19301939")
O<-read.table("19401949")
P<-read.table("19501959")
Q<-read.table("19601969")
R<-read.table("19701979")
S<-read.table("19801989")
T<-read.table("19901999")
U<-read.table("20002009")
V<-read.table("20102019")

## Noordzee 1900: K[2,74]
## Noordzee 1910: +180 K[182,74]
## Noordzee data:
noordzee<-
c(K[seq(from=1082,to=1630,by=180),74],L[seq(from=2,to=1630,by=180),74],M[seq(from=2,to=1630,
by=180),74],
N[seq(from=2,to=1630,by=180),74],O[seq(from=2,to=1630,by=180),74],P[seq(from=2,to=1630,by=1
80),74],
```

```
Q[seq(from=2,to=1630,by=180),74],R[seq(from=2,to=1630,by=180),74],S[seq(from=2,to=1630,by=180),74],
T[seq(from=2,to=1630,by=180),74],U[seq(from=2,to=1630,by=180),74],V[2,74])
Data1$noordzee<-with(Data1,noordzee)

## Oostzee 1900: K[11,74]
oostzee<-
c(K[seq(from=1091,to=1640,by=180),74],L[seq(from=11,to=1640,by=180),74],M[seq(from=11,to=1640,by=180),74],
N[seq(from=11,to=1640,by=180),74],O[seq(from=11,to=1640,by=180),74],P[seq(from=11,to=1640,by=180),74],
Q[seq(from=11,to=1640,by=180),74],R[seq(from=11,to=1640,by=180),74],S[seq(from=11,to=1640,by=180),74],
T[seq(from=11,to=1640,by=180),74],U[seq(from=11,to=1640,by=180),74],V[11,74])
Data1$oostzee<-with(Data1,oostzee)

## Atlantische Oceaan 1900: K[173,66], K[165,59], K[155,54],K[170,39]
atlantisch1<-
c(K[seq(from=1253,to=1800,by=180),66],L[seq(from=173,to=1800,by=180),66],M[seq(from=173,to=1800,by=180),66],
N[seq(from=173,to=1800,by=180),66],O[seq(from=173,to=1800,by=180),66],P[seq(from=173,to=1800,by=180),66],
Q[seq(from=173,to=1800,by=180),66],R[seq(from=173,to=1800,by=180),66],S[seq(from=173,to=1800,by=180),66],
T[seq(from=173,to=1800,by=180),66],U[seq(from=173,to=1800,by=180),66],V[173,66])
Data1$atlantisch1<-with(Data1,atlantisch1)

atlantisch2<-
c(K[seq(from=1245,to=1790,by=180),59],L[seq(from=165,to=1790,by=180),59],M[seq(from=11,to=1790,by=180),59],
N[seq(from=165,to=1790,by=180),59],O[seq(from=165,to=1790,by=180),59],P[seq(from=165,to=1790,by=180),59],
Q[seq(from=165,to=1790,by=180),59],R[seq(from=165,to=1790,by=180),59],S[seq(from=165,to=1790,by=180),59],
T[seq(from=165,to=1790,by=180),59],U[seq(from=165,to=1790,by=180),59],V[165,59])
Data1$atlantisch2<-with(Data1,atlantisch2)

atlantisch3<-
c(K[seq(from=1235,to=1780,by=180),54],L[seq(from=155,to=1780,by=180),54],M[seq(from=11,to=1780,by=180),54],
N[seq(from=155,to=1780,by=180),54],O[seq(from=155,to=1780,by=180),54],P[seq(from=155,to=1780,by=180),54],
Q[seq(from=155,to=1780,by=180),54],R[seq(from=155,to=1780,by=180),54],S[seq(from=155,to=1780,by=180),54],
T[seq(from=155,to=1780,by=180),54],U[seq(from=155,to=1780,by=180),54],V[155,54])
```

```
Data1$atlantisch3<-with(Data1,atlantisch3)

atlantisch4<-
c(K[seq(from=1250,to=1800,by=180),39],L[seq(from=170,to=1800,by=180),39],M[seq(from=170,to=1
800,by=180),39],
N[seq(from=170,to=1800,by=180),39],O[seq(from=170,to=1800,by=180),39],P[seq(from=170,to=180
0,by=180),39],
Q[seq(from=170,to=1800,by=180),39],R[seq(from=170,to=1800,by=180),39],S[seq(from=170,to=180
0,by=180),39],
T[seq(from=170,to=1800,by=180),39],U[seq(from=170,to=1800,by=180),39],V[170,39])
Data1$atlantisch4<-with(Data1,atlantisch4)

A<-read.table("AO")
NAO<-A[seq(from=1, to=105, by=1),1]
Data1$NAO<-with(Data1,NAO)

intercept<-c(rep(1,105))
Data1$intercept<-with(Data1,intercept)

namesvoorspelling<-c("APR","MAY","JUN","JUL","AUG","SEP","nino4apr","nino4may",
"nino4jun","nino4jul","nino4aug","nino4sep","nino3.4apr","nino3.4may",
"nino3.4jun","nino3.4jul","nino3.4aug","nino3.4sep","nino3apr","nino3may",
"nino3jun","nino3jul","nino3aug","nino3sep","nino1.2apr","nino1.2may",
"nino1.2jun","nino1.2jul","nino1.2aug","nino1.2sep","noordzee","oostzee",
"atlantisch1","atlantisch2","atlantisch3","atlantisch4","NAO","intercept")
Voorspelling<-t(c(120,131,156,153,163,148,27.25, 27.06, 26.79, 25.74, 24.79,
 24.35,28.22, 28.58, 28.62, 28.64, 28.48, 28.24,27.21, 27.66, 27.64, 27.09,
 26.53, 26.16,25.28, 24.41, 23.23, 22.19, 21.03, 20.30,622,222,1371,2107,2608,
2711,-1.5,1))
colnames(Voorspelling)<-namesvoorspelling

library(MASS)

fit<-lm(Jan~APR+MAY+JUN+JUL+AUG+SEP,data=Data1)
summary(stepAIC(fit))
AIC((stepAIC(fit)))
AIC(lm(Jan~APR+MAY+JUN+JUL+AUG+SEP,data=Data1))
fit1<-lm(Jan~(APR+MAY+JUN+JUL+AUG+SEP)^2,data=Data1)
summary(stepAIC(fit1)
AIC((stepAIC(fit1)))
fit2<-lm(Jan~nino4apr+nino4may+nino4jun+nino4jul+nino4aug+nino4sep,data=Data1)
summary(stepAIC(fit2))
AIC(stepAIC(fit2))
fit3<-lm(Jan~(nino4apr+nino4may+nino4jun+nino4jul+nino4aug+nino4sep)^2,data=Data1)
summary(stepAIC(fit3))
```

```
AIC(stepAIC(fit3))
fit4<-lm(Jan~nino3.4apr+nino3.4may+nino3.4jun+nino3.4jul+nino3.4aug+nino3.4sep,data=Data1)
summary(stepAIC(fit4))
AIC(stepAIC(fit4))
fit5<-lm(Jan~nino3apr+nino3may+nino3jun+nino3jul+nino3aug+nino3sep,data=Data1)
summary(stepAIC(fit5))
AIC(stepAIC(fit5))
fit6<-lm(Jan~(nino3apr+nino3may+nino3jun+nino3jul+nino3aug+nino3sep)^2,data=Data1)
summary(stepAIC(fit6))
AIC(stepAIC(fit6))
fit7<-lm(Jan~nino1.2apr+nino1.2may+nino1.2jun+nino1.2jul+nino1.2aug+nino1.2sep,data=Data1)
summary(stepAIC(fit7))
AIC(stepAIC(fit7))
fit8<-
lm(Jan~(nino1.2apr+nino1.2may+nino1.2jun+nino1.2jul+nino1.2aug+nino1.2sep)^2,data=Data1)
summary(stepAIC(fit8))
AIC(stepAIC(fit8))
fit9<-lm(Jan~noordzee+oostzee+atlantisch1+atlantisch2+atlantisch3+atlantisch4,data=Data1)
summary(stepAIC(fit9))
AIC(stepAIC(fit9))
fit10<-lm(Jan~(noordzee+oostzee+atlantisch1+atlantisch2+atlantisch3+atlantisch4)^2,data=Data1)
summary(stepAIC(fit10))
AIC(stepAIC(fit10))
fit11<-lm(Jan~NAO,data=Data1)
summary(stepAIC(fit11))
AIC(stepAIC(fit11))

fitoverall<-
lm(Jan~APR+MAY+JUN+AUG+SEP+APR:MAY+APR:JUN+APR:AUG+APR:SEP+MAY:AUG+JUN:AUG+
nino4jun+nino4jul+nino4sep+nino4jul:nino4sep+
nino3apr+nino3may+nino3jun+nino3jul+nino3aug+nino3sep+nino3apr:nino3jun+nino3apr:nino3aug
+nino3apr:nino3sep+nino3may:nino3jun+nino3may:nino3aug+nino3jul:nino3sep
+nino1.2apr+nino1.2may+nino1.2jun+nino1.2jul+nino1.2aug+nino1.2sep+nino1.2apr:nino1.2jun+nin
o1.2apr:nino1.2jul+
nino1.2may:nino1.2jun+nino1.2may:nino1.2jul+oostzee+atlantisch1+atlantisch2+atlantisch3+
atlantisch4+oostzee:atlantisch2+atlantisch1:atlantisch2+atlantisch1:atlantisch3+
atlantisch2:atlantisch4+NAO,data=Data1)
AIC(fitoverall)
summary(stepAIC(fitoverall))
AIC(stepAIC(fitoverall))

fitoverall2<-
lm(Jan~(JUN+SEP+APR:JUN+APR:SEP+nino4jun+nino3jul+nino3sep+nino1.2sep+nino3jul:nino3sep+
nino1.2apr:nino1.2jun+nino1.2apr:nino1.2jul+nino1.2may:nino1.2jun+nino1.2may:nino1.2jul+NAO)^
2,data=Data1)
```

```
summary(stepAIC(fitoverall2))
AIC(stepAIC(fitoverall2))

fitoverall3<-lm(Jan~
 JUN:nino4jun+
 SEP:nino4jun+
 nino4jun:nino3sep+
 nino1.2sep:NAO+
 JUN:nino4jun:APR+
 SEP:nino4jun:APR
 +nino4jun:nino1.2apr:nino1.2jun
 +nino4jun:nino1.2apr:nino1.2jul
 +nino4jun:nino1.2jun:nino1.2may
 +nino4jun:nino1.2jul:nino1.2may
 +nino1.2sep:nino1.2apr:nino1.2jul
 +nino1.2apr:nino1.2jun:nino1.2jul
 +NAO:nino1.2apr:nino1.2jun
 +NAO:nino1.2apr:nino1.2jul
 +nino1.2jun:nino1.2jul:nino1.2may
 +NAO:nino1.2jun:nino1.2may
 +NAO:nino1.2jul:nino1.2may
 +nino3jul:nino3sep:nino1.2apr:nino1.2jun
 +nino3jul:nino3sep:nino1.2apr:nino1.2jul, data=Data1)
AIC(fitoverall3)
summary(stepAIC(fitoverall3))
AIC(stepAIC(fitoverall3))

## LOOCV
## voor fit
A<-as.matrix(Data1)
beta<-c(0,0.3573,0,0,0,0.4483,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-87.6710)
sum((Jan-A%*%beta)^2/diag(diag(1,105)-A%*%solve(t(A)%*%A)%*%t(A)))

## voor fit1
namesB<-c("APR","MAY","JUN","AUG","SEP","APR:MAY","APR:JUN","APR:AUG","APR:SEP",
"MAY:AUG","JUN:AUG","intercept")
B<-matrix(0,nrow=105,ncol=12)
colnames(B)<-namesB
B[,1]<-Data1$APR
B[,2]<-Data1$MAY
B[,3]<-Data1$JUN
B[,4]<-Data1$AUG
B[,5]<-Data1$SEP
B[,6]<-Data1$APR*Data1$MAY
```

```
B[,7]<-Data1$APR*Data1$JUN
B[,8]<-Data1$APR*Data1$AUG
B[,9]<-Data1$APR*Data1$SEP
B[,10]<-Data1$MAY*Data1$AUG
B[,11]<-Data1$JUN*Data1$AUG
B[,12]<-Data1$intercept
beta1<-c(8.19198,10.01153,-7.76585,0.99771,2.22820,-0.04365,0.02815,-0.02349,
-0.02252,-0.03881,0.03622,-632.45060)
sum((Jan-B%*%beta1)^2/diag(diag(1,105)-B%*%solve(t(B)%*%B)%*%t(B)))

## voor fit2
C<-matrix(0,nrow=105,ncol=3)
namesC<-c("intercept","nino4jun","nino4jul")
C[,1]<-Data1$intercept
C[,2]<-Data1$nino4jun
C[,3]<-Data1$nino4jul
colnames(C)<-namesC
beta2<-c(-58.30,17.37,-14.97)
sum((Jan-C%*%beta2)^2/diag(diag(1,105)-C%*%solve(t(C)%*%C)%*%t(C)))

## voor fit3
D<-matrix(0,nrow=105,ncol=5)
namesD<-c("intercept","nino4jun","nino4jul","nino4sep","nino4jul:nino4sep")
D[,1]<-Data1$intercept
D[,2]<-Data1$nino4jun
D[,3]<-Data1$nino4jul
D[,4]<-Data1$nino4sep
D[,5]<-Data1$nino4jul*Data1$nino4sep
colnames(D)<-namesD
beta3<-c(4182.274,18.784,-175.967,-177.490,6.681)
sum((Jan-D%*%beta3)^2/diag(diag(1,105)-D%*%solve(t(D)%*%D)%*%t(D)))

## voor fit4
E<-matrix(0,nrow=105,ncol=1)
E[,1]<-Data1$intercept
beta4<-16.276
sum((Jan-E%*%beta4)^2/diag(diag(1,105)-E%*%solve(t(E)%*%E)%*%t(E)))

## voor fit5
G<-matrix(0,nrow=105,ncol=3)
G[,1]<-Data1$intercept
G[,2]<-Data1$nino3jun
G[,3]<-Data1$nino3jul
beta6<-c(-30.740,18.723,-17.265)
sum((Jan-G%*%beta6)^2/diag(diag(1,105)-G%*%solve(t(G)%*%G)%*%t(G)))
```

```
## voor fit6
a<-matrix(0,nrow=105,ncol=13)
a[,1]<-Data1$intercept
a[,2]<-Data1$nino3apr
a[,3]<-Data1$nino3may
a[,4]<-Data1$nino3jun
a[,5]<-Data1$nino3jul
a[,6]<-Data1$nino3aug
a[,7]<-Data1$nino3sep
a[,8]<-Data1$nino3apr*Data1$nino3jun
a[,9]<-Data1$nino3apr*Data1$nino3aug
a[,10]<-Data1$nino3apr*Data1$nino3sep
a[,11]<-Data1$nino3may*Data1$nino3jun
a[,12]<-Data1$nino3may*Data1$nino3aug
a[,13]<-Data1$nino3jul*Data1$nino3sep
beta7<-c(7567.648,286.655,-95.273,-284.397,-505.122,-349.315,393.558,-44.511
,67.002,-32.161,56.040,-54.698,18.483)
sum((Jan-a%*%beta7)^2/diag(diag(1,105)-a%*%solve(t(a)%*%a)%*%t(a)))

## voor fit7
b<-matrix(0,nrow=105,ncol=3)
b[,1]<-Data1$intercept
b[,2]<-Data1$nino1.2aug
b[,3]<-Data1$nino1.2sep
beta8<-c(38.558,15.523,-16.773)
sum((Jan-b%*%beta8)^2/diag(diag(1,105)-b%*%solve(t(b)%*%b)%*%t(b)))

## voor fit8
c<-matrix(0,nrow=105,ncol=11)
c[,1]<-Data1$intercept
c[,2]<-Data1$nino1.2apr
c[,3]<-Data1$nino1.2may
c[,4]<-Data1$nino1.2jun
c[,5]<-Data1$nino1.2jul
c[,6]<-Data1$nino1.2aug
c[,7]<-Data1$nino1.2sep
c[,8]<-Data1$nino1.2apr*Data1$nino1.2jun
c[,9]<-Data1$nino1.2apr*Data1$nino1.2jul
c[,10]<-Data1$nino1.2may*Data1$nino1.2jun
c[,11]<-Data1$nino1.2may*Data1$nino1.2jul
beta9<-c(-1025.918,195.790,-154.556,-488.502,556.448,17.664,-20.569,72.145
,-84.194,-56.125,65.733)
sum((Jan-c%*%beta9)^2/diag(diag(1,105)-c%*%solve(t(c)%*%c)%*%t(c)))
```

```
## voor fit9
d<-matrix(0,nrow=105,ncol=1)
d[,1]<-Data1$intercept
beta10<-16.276
sum((Jan-d%*%beta10)^2/diag(diag(1,105)-d%*%solve(t(d)%*%d)%*%t(d)))

## voor fit10
f<-matrix(0,nrow=105,ncol=10)
f[,1]<-Data1$intercept
f[,2]<-Data1$oostzee
f[,3]<-Data1$atlantisch1
f[,4]<-Data1$atlantisch2
f[,5]<-Data1$atlantisch3
f[,6]<-Data1$atlantisch4
f[,7]<-Data1$oostzee*Data1$atlantisch2
f[,8]<-Data1$atlantisch1*Data1$atlantisch2
f[,9]<-Data1$atlantisch1*Data1$atlantisch3
f[,10]<-Data1$atlantisch2*Data1$atlantisch4
beta11<-c(4.522e+02,1.028e-01,-1.795e-01,-5.971e-01,3.542e-01,-8.958e-02,-3.202e-05
,3.144e-04,-2.275e-04,4.998e-05)
sum((Jan-f%*%beta11)^2/diag(diag(1,105)-f%*%chol2inv(chol(t(f)%*%f))%*%t(f)))

## voor fit11
g<-matrix(0,nrow=105,ncol=2)
g[,1]<-Data1$intercept
g[,2]<-Data1$NAO
beta12<-c(16.190,5.683)
sum((Jan-g%*%beta12)^2/diag(diag(1,105)-g%*%solve(t(g)%*%g)%*%t(g)))

## voor fitoverall
h<-matrix(0,nrow=105,ncol=27)
h[,1]<-Data1$intercept
h[,2]<-Data1$APR
h[,3]<-Data1$MAY
h[,4]<-Data1$JUN
h[,5]<-Data1$AUG
h[,6]<-Data1$SEP
h[,7]<-Data1$nino4jun
h[,8]<-Data1$nino3apr
h[,9]<-Data1$nino3may
h[,10]<-Data1$nino3jul
h[,11]<-Data1$nino3sep
h[,12]<-Data1$nino1.2apr
h[,13]<-Data1$nino1.2may
h[,14]<-Data1$nino1.2jun
```

```
h[,15]<-Data1$nino1.2jul
h[,16]<-Data1$nino1.2aug
h[,17]<-Data1$nino1.2sep
h[,18]<-Data1$NAO
h[,19]<-Data1$APR*Data1$MAY
h[,20]<-Data1$APR*Data1$JUN
h[,21]<-Data1$APR*Data1$SEP
h[,22]<-Data1$MAY*Data1$AUG
h[,23]<-Data1$nino3jul*Data1$nino3sep
h[,24]<-Data1$nino1.2apr*Data1$nino1.2jun
h[,25]<-Data1$nino1.2apr*Data1$nino1.2jul
h[,26]<-Data1$nino1.2may*Data1$nino1.2jun
h[,27]<-Data1$nino1.2may*Data1$nino1.2jul
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
sum((Jan-h%*%betaoverall)^2/diag(diag(1,105)-h%*%solve(t(h)%*%h)%*%t(h)))

## voor fitoverall2
H<-lm(formula = Jan ~ JUN + SEP + nino4jun + nino3jul + nino3sep +
   nino1.2sep + NAO + JUN:APR + JUN:nino4jun + JUN:nino3jul +
   JUN:nino3sep + JUN:nino1.2sep + JUN:NAO + SEP:APR + SEP:nino4jun +
   SEP:nino3jul + SEP:nino3sep + SEP:nino1.2sep + SEP:NAO +
   nino4jun:nino3jul + nino4jun:nino3sep + nino4jun:NAO + nino3jul:nino3sep +
   nino3jul:nino1.2sep + nino3jul:NAO + nino3sep:nino1.2sep +
   nino3sep:NAO + nino1.2sep:NAO + nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul +
   nino1.2jun:nino1.2may + nino1.2jul:nino1.2may + JUN:nino3jul:nino3sep +
   JUN:nino1.2apr:nino1.2jun + JUN:nino1.2apr:nino1.2jul + JUN:nino1.2jun:nino1.2may +
   JUN:nino1.2jul:nino1.2may + SEP:nino3jul:nino3sep + SEP:nino1.2apr:nino1.2jun +
   SEP:nino1.2jun:nino1.2may + SEP:nino1.2jul:nino1.2may + JUN:nino4jun:APR +
   JUN:nino3jul:APR + JUN:nino3sep:APR + JUN:nino1.2sep:APR +
   JUN:NAO:APR + SEP:nino4jun:APR + SEP:nino3jul:APR + SEP:nino3sep:APR +
   SEP:nino1.2sep:APR + SEP:NAO:APR + nino4jun:nino3jul:nino3sep +
   nino4jun:nino1.2apr:nino1.2jun + nino4jun:nino1.2apr:nino1.2jul +
   nino4jun:nino1.2jun:nino1.2may + nino4jun:nino1.2jul:nino1.2may +
   nino3jul:nino1.2apr:nino1.2jun + nino3jul:nino1.2apr:nino1.2jul +
   nino3jul:nino1.2jun:nino1.2may + nino3jul:nino1.2jul:nino1.2may +
   nino3sep:nino1.2apr:nino1.2jun + nino3sep:nino1.2apr:nino1.2jul +
   nino3sep:nino1.2jun:nino1.2may + nino3sep:nino1.2jul:nino1.2may +
   nino3jul:nino3sep:nino1.2sep + nino1.2sep:nino1.2apr:nino1.2jun +
   nino1.2sep:nino1.2apr:nino1.2jul + nino1.2sep:nino1.2jun:nino1.2may +
   nino1.2sep:nino1.2jul:nino1.2may + nino3jul:nino3sep:NAO +
   nino1.2apr:nino1.2jun:nino1.2jul + nino1.2apr:nino1.2jun:nino1.2may +
   NAO:nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul:nino1.2may +
```

```
    NAO:nino1.2apr:nino1.2jul + nino1.2jun:nino1.2jul:nino1.2may +
    NAO:nino1.2jun:nino1.2may + NAO:nino1.2jul:nino1.2may + JUN:nino3jul:nino3sep:APR +
    JUN:APR:nino1.2apr:nino1.2jun + JUN:APR:nino1.2jun:nino1.2may +
    JUN:APR:nino1.2jul:nino1.2may + SEP:nino3jul:nino3sep:APR +
    SEP:APR:nino1.2apr:nino1.2jun + SEP:APR:nino1.2jun:nino1.2may +
    SEP:APR:nino1.2jul:nino1.2may + nino3jul:nino3sep:nino1.2apr:nino1.2jun +
    nino3jul:nino3sep:nino1.2apr:nino1.2jul, data = Data1,x=TRUE)$x
betaoverall2<-lm(formula = Jan ~ JUN + SEP + nino4jun + nino3jul + nino3sep +
    nino1.2sep + NAO + JUN:APR + JUN:nino4jun + JUN:nino3jul +
    JUN:nino3sep + JUN:nino1.2sep + JUN:NAO + SEP:APR + SEP:nino4jun +
    SEP:nino3jul + SEP:nino3sep + SEP:nino1.2sep + SEP:NAO +
    nino4jun:nino3jul + nino4jun:nino3sep + nino4jun:NAO + nino3jul:nino3sep +
    nino3jul:nino1.2sep + nino3jul:NAO + nino3sep:nino1.2sep +
    nino3sep:NAO + nino1.2sep:NAO + nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul +
    nino1.2jun:nino1.2may + nino1.2jul:nino1.2may + JUN:nino3jul:nino3sep +
    JUN:nino1.2apr:nino1.2jun + JUN:nino1.2apr:nino1.2jul + JUN:nino1.2jun:nino1.2may +
    JUN:nino1.2jul:nino1.2may + SEP:nino3jul:nino3sep + SEP:nino1.2apr:nino1.2jun +
    SEP:nino1.2jun:nino1.2may + SEP:nino1.2jul:nino1.2may + JUN:nino4jun:APR +
    JUN:nino3jul:APR + JUN:nino3sep:APR + JUN:nino1.2sep:APR +
    JUN:NAO:APR + SEP:nino4jun:APR + SEP:nino3jul:APR + SEP:nino3sep:APR +
    SEP:nino1.2sep:APR + SEP:NAO:APR + nino4jun:nino3jul:nino3sep +
    nino4jun:nino1.2apr:nino1.2jun + nino4jun:nino1.2apr:nino1.2jul +
    nino4jun:nino1.2jun:nino1.2may + nino4jun:nino1.2jul:nino1.2may +
    nino3jul:nino1.2apr:nino1.2jun + nino3jul:nino1.2apr:nino1.2jul +
    nino3jul:nino1.2jun:nino1.2may + nino3jul:nino1.2jul:nino1.2may +
    nino3sep:nino1.2apr:nino1.2jun + nino3sep:nino1.2apr:nino1.2jul +
    nino3sep:nino1.2jun:nino1.2may + nino3sep:nino1.2jul:nino1.2may +
    nino3jul:nino3sep:nino1.2sep + nino1.2sep:nino1.2apr:nino1.2jun +
    nino1.2sep:nino1.2apr:nino1.2jul + nino1.2sep:nino1.2jun:nino1.2may +
    nino1.2sep:nino1.2jul:nino1.2may + nino3jul:nino3sep:NAO +
    nino1.2apr:nino1.2jun:nino1.2jul + nino1.2apr:nino1.2jun:nino1.2may +
    NAO:nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul:nino1.2may +
    NAO:nino1.2apr:nino1.2jul + nino1.2jun:nino1.2jul:nino1.2may +
    NAO:nino1.2jun:nino1.2may + NAO:nino1.2jul:nino1.2may + JUN:nino3jul:nino3sep:APR +
    JUN:APR:nino1.2apr:nino1.2jun + JUN:APR:nino1.2jun:nino1.2may +
    JUN:APR:nino1.2jul:nino1.2may + SEP:nino3jul:nino3sep:APR +
    SEP:APR:nino1.2apr:nino1.2jun + SEP:APR:nino1.2jun:nino1.2may +
    SEP:APR:nino1.2jul:nino1.2may + nino3jul:nino3sep:nino1.2apr:nino1.2jun +
    nino3jul:nino3sep:nino1.2apr:nino1.2jul, data = Data1,x=TRUE)$coef
sum((Jan-H%*%betaoverall2)^2/diag(diag(1,105)-H%*%chol2inv(chol(t(H)%*%H))%*%t(H)))

## voor fitoverall3
i<-lm(Jan ~ JUN:nino4jun + nino4jun:SEP + JUN:nino4jun:APR + nino4jun:SEP:APR +
    nino4jun:nino1.2apr:nino1.2jun + nino4jun:nino1.2apr:nino1.2jul +
    nino4jun:nino1.2jun:nino1.2may + nino4jun:nino1.2jul:nino1.2may +
```

```
   nino1.2apr:nino1.2jul:nino1.2sep + nino1.2apr:nino1.2jun:NAO, data=Data1,
x=TRUE)$x
betaoverall3<-lm(Jan ~ JUN:nino4jun + nino4jun:SEP + JUN:nino4jun:APR + nino4jun:SEP:APR +
   nino4jun:nino1.2apr:nino1.2jun + nino4jun:nino1.2apr:nino1.2jul +
   nino4jun:nino1.2jun:nino1.2may + nino4jun:nino1.2jul:nino1.2may +
   nino1.2apr:nino1.2jul:nino1.2sep + nino1.2apr:nino1.2jun:NAO, data=Data1,
x=TRUE)$coef
sum((Jan-i%*%betaoverall3)^2/diag(diag(1,105)-i%*%chol2inv(chol(t(i)%*%i))%*%t(i)))

## Voorspellingen in 0,1 graad celsius en SD in graden celsius
## voor fit
beta<-lm(formula = Jan ~ MAY + SEP, data = Data1)$coef
AA<-matrix(0,nrow=1,ncol=3)
AA[,1]<-1
AA[,2]<-Voorspelling[,2]
AA[,3]<-Voorspelling[,6]
AA%*%beta
A<-matrix(0,nrow=105,ncol=3)
A[,1]<-1
A[,2]<-Data1$MAY
A[,3]<-Data1$SEP
SDfit<-(26.18^2 * (1+ AA%*%solve((t(A)%*%A))%*%t(AA)))^0.5
SDfit

## voor fit1
namesBB<-c("APR","MAY","JUN","AUG","SEP","APR:MAY","APR:JUN","APR:AUG","APR:SEP",
"MAY:AUG","JUN:AUG","intercept")
BB<-matrix(0,nrow=1,ncol=12)
colnames(BB)<-namesBB
BB[,1]<-Voorspelling[,1]
BB[,2]<-Voorspelling[,2]
BB[,3]<-Voorspelling[,3]
BB[,4]<-Voorspelling[,5]
BB[,5]<-Voorspelling[,6]
BB[,6]<-Voorspelling[,1]*Voorspelling[,2]
BB[,7]<-Voorspelling[,1]*Voorspelling[,3]
BB[,8]<-Voorspelling[,1]*Voorspelling[,5]
BB[,9]<-Voorspelling[,1]*Voorspelling[,6]
BB[,10]<-Voorspelling[,2]*Voorspelling[,5]
BB[,11]<-Voorspelling[,3]*Voorspelling[,5]
BB[,12]<-Voorspelling[,38]
beta1<-c(8.19198,10.01153,-7.76585,0.99771,2.22820,-0.04365,0.02815,-0.02349,
-0.02252,-0.03881,0.03622,-632.45060)
BB%*%beta1
SDfit1<-(25.33^2 * (1+ BB%*%solve(t(B)%*%B)%*%t(BB)))^0.5
```

```
SDfit1

## voor fit2
CC<-matrix(0,nrow=1,ncol=3)
namesCC<-c("intercept","nino4jun","nino4jul")
CC[,1]<-Voorspelling[,38]
CC[,2]<-Voorspelling[,9]
CC[,3]<-Voorspelling[,10]
colnames(CC)<-namesCC
beta2<-c(-58.30,17.37,-14.97)
CC%*%beta2
SDfit2<-(26.74^2 * (1+ CC%*%solve(t(C)%*%C)%*%t(CC)))^0.5
SDfit2

## voor fit3
DD<-matrix(0,nrow=1,ncol=5)
namesDD<-c("intercept","nino4jun","nino4jul","nino4sep","nino4jul:nino4sep")
DD[,1]<-Voorspelling[,38]
DD[,2]<-Voorspelling[,9]
DD[,3]<-Voorspelling[,10]
DD[,4]<-Voorspelling[,12]
DD[,5]<-Voorspelling[,10]*Voorspelling[,12]
colnames(DD)<-namesDD
beta3<-c(4182.274,18.784,-175.967,-177.490,6.681)
DD%*%beta3
SDfit3<-(27.49^2 * (1+ DD%*%solve(t(D)%*%D)%*%t(DD)))^0.5
SDfit3

## voor fit4
EE<-matrix(0,nrow=1,ncol=1)
EE[,1]<-Voorspelling[,38]
beta4<-16.276
EE%*%beta4
SDfit4<-(26.94^2 * (1+ EE%*%solve(t(E)%*%E)%*%t(EE)))^0.5
SDfit4

## voor fit5
GG<-matrix(0,nrow=1,ncol=3)
GG[,1]<-Voorspelling[,38]
GG[,2]<-Voorspelling[,21]
GG[,3]<-Voorspelling[,22]
beta6<-c(-30.740,18.723,-17.265)
GG%*%beta6
SDfit5<-(26.87^2 * (1+ GG%*%solve(t(G)%*%G)%*%t(GG)))^0.5
SDfit5
```

```
## voor fit6
aa<-matrix(0,nrow=1,ncol=13)
aa[,1]<-Voorspelling[,38]
aa[,2]<-Voorspelling[,19]
aa[,3]<-Voorspelling[,20]
aa[,4]<-Voorspelling[,21]
aa[,5]<-Voorspelling[,22]
aa[,6]<-Voorspelling[,23]
aa[,7]<-Voorspelling[,24]
aa[,8]<-Voorspelling[,19]*Voorspelling[,21]
aa[,9]<-Voorspelling[,19]*Voorspelling[,23]
aa[,10]<-Voorspelling[,19]*Voorspelling[,24]
aa[,11]<-Voorspelling[,20]*Voorspelling[,21]
aa[,12]<-Voorspelling[,20]*Voorspelling[,23]
aa[,13]<-Voorspelling[,22]*Voorspelling[,24]
beta7<-c(7567.648,286.655,-95.273,-284.397,-505.122,-349.315,393.558,-44.511
,67.002,-32.161,56.040,-54.698,18.483)
aa%*%beta7
SDfit6<-(26.54^2 * (1+ aa%*%solve(t(a)%*%a)%*%t(aa)))^0.5
SDfit6

## voor fit7
bb<-matrix(0,nrow=1,ncol=3)
bb[,1]<-Voorspelling[,38]
bb[,2]<-Voorspelling[,21]
bb[,3]<-Voorspelling[,22]
beta8<-c(38.558,15.523,-16.773)
bb%*%beta8
SDfit7<-(26.27^2 * (1+ bb%*%solve(t(b)%*%b)%*%t(bb)))^0.5
SDfit7

## voor fit8
cc<-matrix(0,nrow=1,ncol=11)
cc[,1]<-Voorspelling[,38]
cc[,2]<-Voorspelling[,25]
cc[,3]<-Voorspelling[,26]
cc[,4]<-Voorspelling[,27]
cc[,5]<-Voorspelling[,28]
cc[,6]<-Voorspelling[,29]
cc[,7]<-Voorspelling[,30]
cc[,8]<-Voorspelling[,25]*Voorspelling[,27]
cc[,9]<-Voorspelling[,25]*Voorspelling[,28]
cc[,10]<-Voorspelling[,26]*Voorspelling[,27]
cc[,11]<-Voorspelling[,26]*Voorspelling[,28]
```

```
beta9<-c(-1025.918,195.790,-154.556,-488.502,556.448,17.664,-20.569,72.145
,-84.194,-56.125,65.733)
cc%*%beta9
SDfit8<-(25.73^2 * (1+ cc%*%solve(t(c)%*%c)%*%t(cc)))^0.5
SDfit8

## voor fit9
dd<-matrix(0,nrow=1,ncol=1)
dd[,1]<-Voorspelling[,38]
beta10<-16.276
beta10
SDfit9<-(26.95^2 * (1+ dd%*%solve(t(d)%*%d)%*%t(dd)))^0.5
SDfit9


## voor fit10
ff<-matrix(0,nrow=1,ncol=10)
ff[,1]<-Voorspelling[,38]
ff[,2]<-Voorspelling[,32]
ff[,3]<-Voorspelling[,33]
ff[,4]<-Voorspelling[,34]
ff[,5]<-Voorspelling[,35]
ff[,6]<-Voorspelling[,36]
ff[,7]<-Voorspelling[,32]*Voorspelling[,34]
ff[,8]<-Voorspelling[,33]*Voorspelling[,34]
ff[,9]<-Voorspelling[,33]*Voorspelling[,35]
ff[,10]<-Voorspelling[,34]*Voorspelling[,36]
beta11<-c(4.522e+02,1.028e-01,-1.795e-01,-5.971e-01,3.542e-01,-8.958e-02,-3.202e-05
,3.144e-04,-2.275e-04,4.998e-05)
ff%*%beta11
SDfit10<-(27.31^2 * (1+ ff%*%chol2inv(chol(t(f)%*%f))%*%t(ff)))^0.5
SDfit10

## voor fit11
gg<-matrix(0,nrow=1,ncol=2)
gg[,1]<-Voorspelling[,38]
gg[,2]<-Voorspelling[,37]
beta12<-c(16.190,5.683)
gg%*%beta12
SDfit11<-(26.06^2 * (1+ gg%*%solve(t(g)%*%g)%*%t(gg)))^0.5
SDfit11

## voor fitoverall
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
```

```
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## voor fitoverall2
H<-lm(formula = Jan ~ JUN + SEP + nino4jun + nino3jul + nino3sep +
   nino1.2sep + NAO + JUN:APR + JUN:nino4jun + JUN:nino3jul +
   JUN:nino3sep + JUN:nino1.2sep + JUN:NAO + SEP:APR + SEP:nino4jun +
   SEP:nino3jul + SEP:nino3sep + SEP:nino1.2sep + SEP:NAO +
   nino4jun:nino3jul + nino4jun:nino3sep + nino4jun:NAO + nino3jul:nino3sep +
   nino3jul:nino1.2sep + nino3jul:NAO + nino3sep:nino1.2sep +
   nino3sep:NAO + nino1.2sep:NAO + nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul +
   nino1.2jun:nino1.2may + nino1.2jul:nino1.2may + JUN:nino3jul:nino3sep +
   JUN:nino1.2apr:nino1.2jun + JUN:nino1.2apr:nino1.2jul + JUN:nino1.2jun:nino1.2may +
   JUN:nino1.2jul:nino1.2may + SEP:nino3jul:nino3sep + SEP:nino1.2apr:nino1.2jun +
```

```
  SEP:nino1.2jun:nino1.2may + SEP:nino1.2jul:nino1.2may + JUN:nino4jun:APR +
  JUN:nino3jul:APR + JUN:nino3sep:APR + JUN:nino1.2sep:APR +
  JUN:NAO:APR + SEP:nino4jun:APR + SEP:nino3jul:APR + SEP:nino3sep:APR +
  SEP:nino1.2sep:APR + SEP:NAO:APR + nino4jun:nino3jul:nino3sep +
  nino4jun:nino1.2apr:nino1.2jun + nino4jun:nino1.2apr:nino1.2jul +
  nino4jun:nino1.2jun:nino1.2may + nino4jun:nino1.2jul:nino1.2may +
  nino3jul:nino1.2apr:nino1.2jun + nino3jul:nino1.2apr:nino1.2jul +
  nino3jul:nino1.2jun:nino1.2may + nino3jul:nino1.2jul:nino1.2may +
  nino3sep:nino1.2apr:nino1.2jun + nino3sep:nino1.2apr:nino1.2jul +
  nino3sep:nino1.2jun:nino1.2may + nino3sep:nino1.2jul:nino1.2may +
  nino3jul:nino3sep:nino1.2sep + nino1.2sep:nino1.2apr:nino1.2jun +
  nino1.2sep:nino1.2apr:nino1.2jul + nino1.2sep:nino1.2jun:nino1.2may +
  nino1.2sep:nino1.2jul:nino1.2may + nino3jul:nino3sep:NAO +
  nino1.2apr:nino1.2jun:nino1.2jul + nino1.2apr:nino1.2jun:nino1.2may +
  NAO:nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul:nino1.2may +
  NAO:nino1.2apr:nino1.2jul + nino1.2jun:nino1.2jul:nino1.2may +
  NAO:nino1.2jun:nino1.2may + NAO:nino1.2jul:nino1.2may + JUN:nino3jul:nino3sep:APR +
  JUN:APR:nino1.2apr:nino1.2jun + JUN:APR:nino1.2jun:nino1.2may +
  JUN:APR:nino1.2jul:nino1.2may + SEP:nino3jul:nino3sep:APR +
  SEP:APR:nino1.2apr:nino1.2jun + SEP:APR:nino1.2jun:nino1.2may +
  SEP:APR:nino1.2jul:nino1.2may + nino3jul:nino3sep:nino1.2apr:nino1.2jun +
  nino3jul:nino3sep:nino1.2apr:nino1.2jul, data = Data1,x=TRUE)$x
betaoverall2<-lm(formula = Jan ~ JUN + SEP + nino4jun + nino3jul + nino3sep +
  nino1.2sep + NAO + JUN:APR + JUN:nino4jun + JUN:nino3jul +
  JUN:nino3sep + JUN:nino1.2sep + JUN:NAO + SEP:APR + SEP:nino4jun +
  SEP:nino3jul + SEP:nino3sep + SEP:nino1.2sep + SEP:NAO +
  nino4jun:nino3jul + nino4jun:nino3sep + nino4jun:NAO + nino3jul:nino3sep +
  nino3jul:nino1.2sep + nino3jul:NAO + nino3sep:nino1.2sep +
  nino3sep:NAO + nino1.2sep:NAO + nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul +
  nino1.2jun:nino1.2may + nino1.2jul:nino1.2may + JUN:nino3jul:nino3sep +
  JUN:nino1.2apr:nino1.2jun + JUN:nino1.2apr:nino1.2jul + JUN:nino1.2jun:nino1.2may +
  JUN:nino1.2jul:nino1.2may + SEP:nino3jul:nino3sep + SEP:nino1.2apr:nino1.2jun +
  SEP:nino1.2jun:nino1.2may + SEP:nino1.2jul:nino1.2may + JUN:nino4jun:APR +
  JUN:nino3jul:APR + JUN:nino3sep:APR + JUN:nino1.2sep:APR +
  JUN:NAO:APR + SEP:nino4jun:APR + SEP:nino3jul:APR + SEP:nino3sep:APR +
  SEP:nino1.2sep:APR + SEP:NAO:APR + nino4jun:nino3jul:nino3sep +
  nino4jun:nino1.2apr:nino1.2jun + nino4jun:nino1.2apr:nino1.2jul +
  nino4jun:nino1.2jun:nino1.2may + nino4jun:nino1.2jul:nino1.2may +
  nino3jul:nino1.2apr:nino1.2jun + nino3jul:nino1.2apr:nino1.2jul +
  nino3jul:nino1.2jun:nino1.2may + nino3jul:nino1.2jul:nino1.2may +
  nino3sep:nino1.2apr:nino1.2jun + nino3sep:nino1.2apr:nino1.2jul +
  nino3sep:nino1.2jun:nino1.2may + nino3sep:nino1.2jul:nino1.2may +
  nino3jul:nino3sep:nino1.2sep + nino1.2sep:nino1.2apr:nino1.2jun +
  nino1.2sep:nino1.2apr:nino1.2jul + nino1.2sep:nino1.2jun:nino1.2may +
  nino1.2sep:nino1.2jul:nino1.2may + nino3jul:nino3sep:NAO +
```

```
    nino1.2apr:nino1.2jun:nino1.2jul + nino1.2apr:nino1.2jun:nino1.2may +
    NAO:nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul:nino1.2may +
    NAO:nino1.2apr:nino1.2jul + nino1.2jun:nino1.2jul:nino1.2may +
    NAO:nino1.2jun:nino1.2may + NAO:nino1.2jul:nino1.2may + JUN:nino3jul:nino3sep:APR +
    JUN:APR:nino1.2apr:nino1.2jun + JUN:APR:nino1.2jun:nino1.2may +
    JUN:APR:nino1.2jul:nino1.2may + SEP:nino3jul:nino3sep:APR +
    SEP:APR:nino1.2apr:nino1.2jun + SEP:APR:nino1.2jun:nino1.2may +
    SEP:APR:nino1.2jul:nino1.2may + nino3jul:nino3sep:nino1.2apr:nino1.2jun +
    nino3jul:nino3sep:nino1.2apr:nino1.2jul, data = Data1,x=TRUE)$coef
HH<-matrix(0,nrow=1,ncol=89)
HH[,1]<-Voorspelling[,38]
HH[,2]<-Voorspelling[,3]
HH[,3]<-Voorspelling[,6]
HH[,4]<-Voorspelling[,9]
HH[,5]<-Voorspelling[,22]
HH[,6]<-Voorspelling[,24]
HH[,7]<-Voorspelling[,30]
HH[,8]<-Voorspelling[,37]
HH[,9]<-Voorspelling[,3]*Voorspelling[,1]
HH[,10]<-Voorspelling[,3]*Voorspelling[,9]
HH[,11]<-Voorspelling[,3]*Voorspelling[,22]
HH[,12]<-Voorspelling[,3]*Voorspelling[,24]
HH[,13]<-Voorspelling[,3]*Voorspelling[,30]
HH[,14]<-Voorspelling[,3]*Voorspelling[,37]
HH[,15]<-Voorspelling[,6]*Voorspelling[,1]
HH[,16]<-Voorspelling[,6]*Voorspelling[,9]
HH[,17]<-Voorspelling[,6]*Voorspelling[,22]
HH[,18]<-Voorspelling[,6]*Voorspelling[,24]
HH[,19]<-Voorspelling[,6]*Voorspelling[,30]
HH[,20]<-Voorspelling[,6]*Voorspelling[,37]
HH[,21]<-Voorspelling[,9]*Voorspelling[,22]
HH[,22]<-Voorspelling[,9]*Voorspelling[,24]
HH[,23]<-Voorspelling[,9]*Voorspelling[,37]
HH[,24]<-Voorspelling[,22]*Voorspelling[,24]
HH[,25]<-Voorspelling[,22]*Voorspelling[,30]
HH[,26]<-Voorspelling[,22]*Voorspelling[,37]
HH[,27]<-Voorspelling[,24]*Voorspelling[,30]
HH[,28]<-Voorspelling[,24]*Voorspelling[,37]
HH[,29]<-Voorspelling[,30]*Voorspelling[,37]
HH[,30]<-Voorspelling[,25]*Voorspelling[,27]
HH[,31]<-Voorspelling[,25]*Voorspelling[,28]
HH[,32]<-Voorspelling[,25]*Voorspelling[,26]
HH[,33]<-Voorspelling[,28]*Voorspelling[,26]
HH[,34]<-Voorspelling[,3]*Voorspelling[,22]*Voorspelling[,24]
HH[,35]<-Voorspelling[,3]*Voorspelling[,25]*Voorspelling[,27]
```

```
HH[,36]<-Voorspelling[,3]*Voorspelling[,25]*Voorspelling[,28]
HH[,37]<-Voorspelling[,3]*Voorspelling[,27]*Voorspelling[,26]
HH[,38]<-Voorspelling[,3]*Voorspelling[,28]*Voorspelling[,26]
HH[,39]<-Voorspelling[,6]*Voorspelling[,22]*Voorspelling[,24]
HH[,40]<-Voorspelling[,6]*Voorspelling[,25]*Voorspelling[,27]
HH[,41]<-Voorspelling[,6]*Voorspelling[,27]*Voorspelling[,26]
HH[,42]<-Voorspelling[,6]*Voorspelling[,28]*Voorspelling[,26]
HH[,43]<-Voorspelling[,3]*Voorspelling[,9]*Voorspelling[,1]
HH[,44]<-Voorspelling[,3]*Voorspelling[,22]*Voorspelling[,1]
HH[,45]<-Voorspelling[,3]*Voorspelling[,24]*Voorspelling[,1]
HH[,46]<-Voorspelling[,3]*Voorspelling[,30]*Voorspelling[,1]
HH[,47]<-Voorspelling[,3]*Voorspelling[,37]*Voorspelling[,1]
HH[,48]<-Voorspelling[,6]*Voorspelling[,9]*Voorspelling[,1]
HH[,49]<-Voorspelling[,6]*Voorspelling[,22]*Voorspelling[,1]
HH[,50]<-Voorspelling[,6]*Voorspelling[,24]*Voorspelling[,1]
HH[,51]<-Voorspelling[,6]*Voorspelling[,30]*Voorspelling[,1]
HH[,52]<-Voorspelling[,6]*Voorspelling[,37]*Voorspelling[,1]
HH[,53]<-Voorspelling[,9]*Voorspelling[,22]*Voorspelling[,24]
HH[,54]<-Voorspelling[,9]*Voorspelling[,25]*Voorspelling[,27]
HH[,55]<-Voorspelling[,9]*Voorspelling[,25]*Voorspelling[,28]
HH[,56]<-Voorspelling[,9]*Voorspelling[,27]*Voorspelling[,26]
HH[,57]<-Voorspelling[,9]*Voorspelling[,28]*Voorspelling[,26]
HH[,58]<-Voorspelling[,22]*Voorspelling[,25]*Voorspelling[,27]
HH[,59]<-Voorspelling[,22]*Voorspelling[,25]*Voorspelling[,28]
HH[,60]<-Voorspelling[,22]*Voorspelling[,27]*Voorspelling[,26]
HH[,61]<-Voorspelling[,22]*Voorspelling[,28]*Voorspelling[,26]
HH[,62]<-Voorspelling[,24]*Voorspelling[,25]*Voorspelling[,27]
HH[,63]<-Voorspelling[,24]*Voorspelling[,25]*Voorspelling[,28]
HH[,64]<-Voorspelling[,24]*Voorspelling[,27]*Voorspelling[,26]
HH[,65]<-Voorspelling[,24]*Voorspelling[,28]*Voorspelling[,26]
HH[,66]<-Voorspelling[,22]*Voorspelling[,24]*Voorspelling[,30]
HH[,67]<-Voorspelling[,30]*Voorspelling[,25]*Voorspelling[,27]
HH[,68]<-Voorspelling[,30]*Voorspelling[,25]*Voorspelling[,28]
HH[,69]<-Voorspelling[,30]*Voorspelling[,27]*Voorspelling[,26]
HH[,70]<-Voorspelling[,30]*Voorspelling[,28]*Voorspelling[,26]
HH[,71]<-Voorspelling[,22]*Voorspelling[,24]*Voorspelling[,37]
HH[,72]<-Voorspelling[,25]*Voorspelling[,27]*Voorspelling[,28]
HH[,73]<-Voorspelling[,25]*Voorspelling[,27]*Voorspelling[,26]
HH[,74]<-Voorspelling[,37]*Voorspelling[,25]*Voorspelling[,27]
HH[,75]<-Voorspelling[,25]*Voorspelling[,28]*Voorspelling[,26]
HH[,76]<-Voorspelling[,37]*Voorspelling[,25]*Voorspelling[,28]
HH[,77]<-Voorspelling[,27]*Voorspelling[,28]*Voorspelling[,26]
HH[,78]<-Voorspelling[,37]*Voorspelling[,27]*Voorspelling[,26]
HH[,79]<-Voorspelling[,37]*Voorspelling[,28]*Voorspelling[,26]
HH[,80]<-Voorspelling[,3]*Voorspelling[,22]*Voorspelling[,24]*Voorspelling[,1]
```

```
HH[,81]<-Voorspelling[,3]*Voorspelling[,1]*Voorspelling[,25]*Voorspelling[,27]
HH[,82]<-Voorspelling[,3]*Voorspelling[,1]*Voorspelling[,27]*Voorspelling[,26]
HH[,83]<-Voorspelling[,3]*Voorspelling[,1]*Voorspelling[,28]*Voorspelling[,26]
HH[,84]<-Voorspelling[,6]*Voorspelling[,22]*Voorspelling[,24]*Voorspelling[,1]
HH[,85]<-Voorspelling[,6]*Voorspelling[,1]*Voorspelling[,25]*Voorspelling[,27]
HH[,86]<-Voorspelling[,6]*Voorspelling[,1]*Voorspelling[,27]*Voorspelling[,26]
HH[,87]<-Voorspelling[,6]*Voorspelling[,1]*Voorspelling[,28]*Voorspelling[,26]
HH[,88]<-Voorspelling[,22]*Voorspelling[,24]*Voorspelling[,25]*Voorspelling[,27]
HH[,89]<-Voorspelling[,22]*Voorspelling[,24]*Voorspelling[,25]*Voorspelling[,28]
HH%*%betaoverall2
SDfitoverall2<-(15.49^2 * (1+ HH%*%chol2inv(chol(t(H)%*%H))%*%t(HH)))^0.5
SDfitoverall2

## voor fitoverall3
i<-lm(Jan ~ JUN:nino4jun + nino4jun:SEP + JUN:nino4jun:APR + nino4jun:SEP:APR +
   nino4jun:nino1.2apr:nino1.2jun + nino4jun:nino1.2apr:nino1.2jul +
   nino4jun:nino1.2jun:nino1.2may + nino4jun:nino1.2jul:nino1.2may +
   nino1.2apr:nino1.2jul:nino1.2sep + nino1.2apr:nino1.2jun:NAO, data=Data1,
x=TRUE)$x
betaoverall3<-lm(Jan ~ JUN:nino4jun + nino4jun:SEP + JUN:nino4jun:APR + nino4jun:SEP:APR +
   nino4jun:nino1.2apr:nino1.2jun + nino4jun:nino1.2apr:nino1.2jul +
   nino4jun:nino1.2jun:nino1.2may + nino4jun:nino1.2jul:nino1.2may +
   nino1.2apr:nino1.2jul:nino1.2sep + nino1.2apr:nino1.2jun:NAO, data=Data1,
x=TRUE)$coef
ii<-matrix(0,nrow=1,ncol=11)
ii[,1]<-Voorspelling[,38]
ii[,2]<-Voorspelling[,3]*Voorspelling[,9]
ii[,3]<-Voorspelling[,9]*Voorspelling[,6]
ii[,4]<-Voorspelling[,3]*Voorspelling[,9]*Voorspelling[,1]
ii[,5]<-Voorspelling[,9]*Voorspelling[,6]*Voorspelling[,1]
ii[,6]<-Voorspelling[,9]*Voorspelling[,25]*Voorspelling[,27]
ii[,7]<-Voorspelling[,9]*Voorspelling[,25]*Voorspelling[,28]
ii[,8]<-Voorspelling[,9]*Voorspelling[,27]*Voorspelling[,26]
ii[,9]<-Voorspelling[,9]*Voorspelling[,28]*Voorspelling[,26]
ii[,10]<-Voorspelling[,25]*Voorspelling[,28]*Voorspelling[,30]
ii[,11]<-Voorspelling[,25]*Voorspelling[,27]*Voorspelling[,37]
ii%*%betaoverall3
SDfitoverall3<-(23.55^2 * (1+ ii%*%solve((t(i)%*%i))%*%t(ii)))^0.5
SDfitoverall3

## Gemiddelde van gemiddelde temperatuur in Januari:
sum(Jan)/length(Jan)

## Mixed model
climAR1.e<-lme(Jan ~ APR + MAY + JUN + AUG + SEP + nino4jun + nino3apr +
```

```
   nino3may + nino3jul + nino3sep + nino1.2apr + nino1.2may +
   nino1.2jun + nino1.2jul + nino1.2aug + nino1.2sep + NAO +
   APR:MAY + APR:JUN + APR:SEP + MAY:AUG + nino3jul:nino3sep +
   nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul + nino1.2may:nino1.2jun +
   nino1.2may:nino1.2jul
,random=~1|group,data=Data1, corr=corAR1(.2,form=~Time))
summary(climAR1.e)
climAR1.ml.e<-lme(Jan ~ APR + MAY + JUN + AUG + SEP + nino4jun + nino3apr +
   nino3may + nino3jul + nino3sep + nino1.2apr + nino1.2may +
   nino1.2jun + nino1.2jul + nino1.2aug + nino1.2sep + NAO +
   APR:MAY + APR:JUN + APR:SEP + MAY:AUG + nino3jul:nino3sep +
   nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul + nino1.2may:nino1.2jun +
   nino1.2may:nino1.2jul
,random=~1|group,data=Data1, corr=corAR1(.2,form=~Time),method="ML")
summary(climAR1.ml.e)$AIC

climMA.e<-lme(Jan ~ APR + MAY + JUN + AUG + SEP + nino4jun + nino3apr +
   nino3may + nino3jul + nino3sep + nino1.2apr + nino1.2may +
   nino1.2jun + nino1.2jul + nino1.2aug + nino1.2sep + NAO +
   APR:MAY + APR:JUN + APR:SEP + MAY:AUG + nino3jul:nino3sep +
   nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul + nino1.2may:nino1.2jun +
   nino1.2may:nino1.2jul
,random=~1|group,data=Data1, corr=corARMA(0.1,p=0,q=1,form=~Time))
summary(climMA.e)
climMA.ml.e<-lme(Jan ~ APR + MAY + JUN + AUG + SEP + nino4jun + nino3apr +
   nino3may + nino3jul + nino3sep + nino1.2apr + nino1.2may +
   nino1.2jun + nino1.2jul + nino1.2aug + nino1.2sep + NAO +
   APR:MAY + APR:JUN + APR:SEP + MAY:AUG + nino3jul:nino3sep +
   nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul + nino1.2may:nino1.2jun +
   nino1.2may:nino1.2jul
,random=~1|group,data=Data1, corr=corARMA(0.1,p=0,q=1,form=~Time), method="ML")
summary(climMA.ml.e)$AIC

climARMA.e<-lme(Jan ~ APR + MAY + JUN + AUG + SEP + nino4jun + nino3apr +
   nino3may + nino3jul + nino3sep + nino1.2apr + nino1.2may +
   nino1.2jun + nino1.2jul + nino1.2aug + nino1.2sep + NAO +
   APR:MAY + APR:JUN + APR:SEP + MAY:AUG + nino3jul:nino3sep +
   nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul + nino1.2may:nino1.2jun +
   nino1.2may:nino1.2jul
,random=~1|group,data=Data1, corr=corARMA(c(0.1,0.1),p=1,q=1,form=~Time))
summary(climARMA.e)
climARMA.ml.e<-lme(Jan ~ APR + MAY + JUN + AUG + SEP + nino4jun + nino3apr +
   nino3may + nino3jul + nino3sep + nino1.2apr + nino1.2may +
   nino1.2jun + nino1.2jul + nino1.2aug + nino1.2sep + NAO +
   APR:MAY + APR:JUN + APR:SEP + MAY:AUG + nino3jul:nino3sep +
```

```
   nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul + nino1.2may:nino1.2jun +
   nino1.2may:nino1.2jul
,random=~1|group,data=Data1, corr=corARMA(c(0.1,0.1),p=1,q=1,form=~Time), method="ML")
summary(climARMA.ml.e)$AIC

## Ridge regressie
## AIC van Ridge regressie berekenen:
X<-lm(formula = Jan ~ APR + MAY + JUN + AUG + SEP + nino4jun + nino3apr +
   nino3may + nino3jul + nino3sep + nino1.2apr + nino1.2may +
   nino1.2jun + nino1.2jul + nino1.2aug + nino1.2sep + NAO +
   APR:MAY + APR:JUN + APR:SEP + MAY:AUG + nino3jul:nino3sep +
   nino1.2apr:nino1.2jun + nino1.2apr:nino1.2jul + nino1.2may:nino1.2jun +
   nino1.2may:nino1.2jul, data = Data1,x=TRUE)$x
y<-Jan
Z<-X[,2:27]
n=105
I<-diag(1,26)
J<-diag(1,n)
K<-diag(1,27)
AIC<-
matrix(c(seq(from=0,to=0.001,by=0.00001),rep(0,length(c(seq(from=0,to=0.001,by=0.00001))))),ncol
=2)

for(lambda in seq(from=0,to=0.001,by=0.00001))
{
S<-Z%*%solve(t(Z)%*%Z+lambda*I)%*%t(Z)
df<-sum(diag(S))
AIC[100000*lambda+1,2]<-n*log((t(y-X%*%solve(t(X)%*%X+lambda*K)%*%t(X)%*%y)%*%(y-
X%*%solve(t(X)%*%X+lambda*K)%*%t(X)%*%y))/n)+n*(log(2*pi)+1)+2*(df+2)
}
AIC
plot(AIC,xlab="lambda",ylab="AIC")


## April 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]+10
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
```

```
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-(Voorspelling[,1]+10)*Voorspelling[,2]
hh[,20]<-(Voorspelling[,1]+10)*Voorspelling[,3]
hh[,21]<-(Voorspelling[,1]+10)*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## Mei 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]+10
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
```

```
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*(Voorspelling[,2]+10)
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-(Voorspelling[,2]+10)*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## Juni 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]+10
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*(Voorspelling[,3]+10)
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
```

```
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## Augustus 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]+10
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*(Voorspelling[,5]+10)
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
```

SDfitoverall

September 1 graad warmer
```
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]+10
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*(Voorspelling[,6]+10)
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall
```

nino4jun 1 graad warmer
```
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
```

```
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]+1
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall


## nino3apr 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]+1
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
```

```
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## nino3may 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]+1
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
```

```
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## nino3jul 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]+1
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-(Voorspelling[,22]+1)*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
```

```
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## nino3sep 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]+1
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*(Voorspelling[,24]+1)
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall
```

```
## nino1.2apr 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]+1
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-(Voorspelling[,25]+1)*Voorspelling[,27]
hh[,25]<-(Voorspelling[,25]+1)*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall


## nino1.2may 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
```

```
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]+1
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-(Voorspelling[,26]+1)*Voorspelling[,27]
hh[,27]<-(Voorspelling[,26]+1)*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## nino1.2jun 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
```

```
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]+1
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*(Voorspelling[,27]+1)
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*(Voorspelling[,27]+1)
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## nino1.2jul 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]+1
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
```

```
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*(Voorspelling[,28]+1)
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*(Voorspelling[,28]+1)
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## nino1.2aug 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]+1
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
```

```
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## nino1.2sep 1 graad warmer
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]+1
hh[,18]<-Voorspelling[,37]
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall

## NAO 1 eenheid groter
```

```
hh<-matrix(0,nrow=1,ncol=27)
hh[,1]<-Voorspelling[,38]
hh[,2]<-Voorspelling[,1]
hh[,3]<-Voorspelling[,2]
hh[,4]<-Voorspelling[,3]
hh[,5]<-Voorspelling[,5]
hh[,6]<-Voorspelling[,6]
hh[,7]<-Voorspelling[,9]
hh[,8]<-Voorspelling[,19]
hh[,9]<-Voorspelling[,20]
hh[,10]<-Voorspelling[,22]
hh[,11]<-Voorspelling[,24]
hh[,12]<-Voorspelling[,25]
hh[,13]<-Voorspelling[,26]
hh[,14]<-Voorspelling[,27]
hh[,15]<-Voorspelling[,28]
hh[,16]<-Voorspelling[,29]
hh[,17]<-Voorspelling[,30]
hh[,18]<-Voorspelling[,37]+1
hh[,19]<-Voorspelling[,1]*Voorspelling[,2]
hh[,20]<-Voorspelling[,1]*Voorspelling[,3]
hh[,21]<-Voorspelling[,1]*Voorspelling[,6]
hh[,22]<-Voorspelling[,2]*Voorspelling[,5]
hh[,23]<-Voorspelling[,22]*Voorspelling[,24]
hh[,24]<-Voorspelling[,25]*Voorspelling[,27]
hh[,25]<-Voorspelling[,25]*Voorspelling[,28]
hh[,26]<-Voorspelling[,26]*Voorspelling[,27]
hh[,27]<-Voorspelling[,26]*Voorspelling[,28]
betaoverall<-c(7563.59265,4.55563,5.48363,-2.21285,1.90491,3.35483,38.37964,18.45336
,-20.26791,-388.40573,-356.70918,171.39887,-129.06865,-274.20071,346.94822,13.75564
,-26.62768,4.71410,-0.02955,0.03096,-0.03874,-0.01736,13.59921,59.20868,-69.43331
,-50.96622,58.60629)
hh%*%betaoverall
SDfitoverall<-(22.49^2 * (1+ hh%*%solve(t(h)%*%h)%*%t(hh)))^0.5
SDfitoverall
```