

# Delft Meet Regen

Onderzoek naar de dagelijkse neerslagspreiding in Delft op basis van de  
neerslagdata van Delft Meet Regen

Rafaël van Beek

Bachelor of Science aan de Technische Universiteit Delft

Oktober 2021

Studentennummer: 4477421

Begeleiders:	Marie-Claire ten Veldhuis	TU Delft
	Remko Uijlenhoet	TU Delft
	Sandra de Vries	TU Delft

## Voorwoord

Dit is een rapport voor het Bachelor eind project, dit is voor de afronding van de Bachelor of Science Civiele Techniek aan de Technische Universiteit Delft. In dit rapport wordt de dagelijkse regenspreiding van Delft onderzocht op basis van de regendata van het Delft Meet Regen Project.

Hierbij wil ik Marie-Claire ten Veldhuis, Remko Uijlenhoet en Sandra de Vries bedanken voor hun begeleiding en waardevolle adviezen tijdens dit project.

Delft, oktober 2021  
Rafaël van Beek

## Samenvatting

In dit rapport zal er onderzocht worden wat de dagelijkse ruimtelijke spreiding van regen in Delft is op basis van de neerslagdata van het Delft Meet Regen project. In totaal hebben er 90 vrijwilligers meegedaan aan dit project en zijn er dus ook 90 meetpunten. De neerslagdata die gebruikt is voor dit onderzoek is de neerslagdata in de periode van 1 augustus 2021 tot en met 17 september 2021. De neerslagdata is vervolgens vergeleken met de data van KNMI station 449, door de dagelijkse gemiddelden normaal en cumulatief met elkaar te vergelijken. De KNMI data heeft een goede overeenkomst met de neerslagdata van het Delft Meet Regen project.

Hierna is het onderzoeksgebied ingedeeld in de volgende vier gebieden: noordwest, noordoost, zuidwest en zuidoost. Van deze gebieden zijn de dagelijkse gemiddelden, aantal metingen, standaard afwijkingen en de dagelijkse gemiddelden cumulatief met elkaar vergeleken. Dit is gedaan doormiddel van grafieken, staafdiagrammen en ruimtelijke plots. Hieruit is geconcludeerd dat er sprake is van gemiddelde dagelijkse neerslagverschillen tussen de gebieden, maar dat er geen sprake is van een patroon. Het aantal metingen per gebied verschilt per dag en dit heeft invloed op de nauwkeurigheid van de resultaten. Hoe meer metingen er beschikbaar zijn, hoe nauwkeuriger de resultaten zullen zijn.

Na de gebiedenanalyse is een ruimtelijke correlatie analyse uitgevoerd. Hierin is Pearsons correlatie coëfficiënt en de afstand tussen elke mogelijke combinatie van twee meetpunten berekend, dit is gedaan met een drempelwaarde van 5 overeenkomstige dagen en een drempelwaarde van 10 overeenkomstige dagen. De uitgerekende correlaties zijn vervolgens tegen de afstand uitgezet in een spreidingsdiagram en ingedeeld in afstandsintervallen. Per afstandsinterval is de gemiddelde afstand en gemiddelde correlatie uitgerekend. Door deze zwaartepunten is vervolgens een exponentiële functie gefit. De parameters van deze functie zijn met de kleinste kwadraten methode berekend. Uit de ruimtelijke correlatie analyse kan geconcludeerd worden dat er een verband is tussen de ruimtelijke correlatie en de afstand tussen de meetpunten. Hoe groter de afstand tussen de meetpunten, hoe lager de correlatie. Dit betekent dat de kans op neerslagverschillen tussen deze metingen groter wordt en dat dus de kans op neerslagspreiding in het onderzoeksgebied groter wordt. Verder neemt de nauwkeurigheid van de correlaties toe als de drempelwaarde hoger is.

Uit dit onderzoek kan geconcludeerd worden dat er sprake is van dagelijkse neerslagspreiding op basis van de Delft Meet Regen data, maar dat hierin geen patroon in te ontdekken is. Om hierin een eventueel patroon in te ontdekken zullen er meer metingen voor een langere periode uitgevoerd moeten worden. Dit zal de nauwkeurigheid van de resultaten bevorderen.

## Inhoudsopgave

Voorwoord .....	2
Samenvatting .....	3
1. Introductie .....	5
1.1 Probleem analyse.....	5
1.2 Delft Meet Regen project.....	5
1.3 Doelstelling en afbakening.....	6
1.4 Onderzoeksvraag en deelvragen .....	6
2. Data .....	7
3. Methodologie.....	10
4. Resultaten .....	13
4.1 Data verificatie .....	13
4.2 Gebiedenanalyse.....	14
4.3 Ruimtelijke correlatie analyse.....	17
5. Discussie .....	22
6. Conclusie .....	23
7. Aanbevelingen .....	24
Literatuurlijst.....	25
Bijlages .....	26
Bijlage A. Ruimtelijke plots neerslagmetingen per dag .....	26
Bijlage B. Matlab code converteren coördinaten .....	32
Bijlage C. Python scripts.....	35

# 1. Introductie

## 1.1 Probleem analyse

Door klimaatverandering is het mogelijk dat het neerslagpatroon gaat veranderen in Nederland, wat ook gevolgen zal hebben voor de stad Delft. De regenbuien kunnen hierdoor intensiever worden en frequenter voorkomen, wat op verschillende locaties in de stad overlast kan veroorzaken. Door de verharding in de stad kan het regenwater moeilijk infiltreren in de grond. Hierdoor wordt een grote hoeveelheid van het regenwater via het riool afgevoerd. De riolering kan na een intensieve regenbui overbelast raken, waardoor het regenwater samen met het afvalwater op straat komt, wat overlast zal geven aan de buurt. Het is ook mogelijk dat er door klimaatverandering periodes ontstaan waarin weinig neerslag valt. Dit heeft gevolgen voor de vegetatie, bodemdaling en leefklimaat in de stad. Hierdoor is het van belang om meer inzicht te krijgen in de ruimtelijke spreiding van regen in Delft. Hiermee kunnen mogelijke water- bottlenecks worden gelokaliseerd en kunnen er maatregelen bedacht worden om de stad meer klimaat adaptief te maken.

## 1.2 Delft Meet Regen project

Vanuit het Waterlab is het project Delft Meet Regen opgestart. Het project heeft als doel om meer inzicht te krijgen in de ruimtelijke neerslagverschillen in Delft. Dit jaar zullen zo rond de 90 deelnemers meedoen om individueel dagelijks de regen te meten in de periode van 1 augustus tot en met 30 oktober (Waterlab, 2021). Elke deelnemer ontvangt dezelfde regenmeter en zal rond hetzelfde tijdstip dagelijks meten, tussen 8 uur en 10 uur in de ochtend. De methode van meten is een directe methode: de regen valt in de trechter en kan met het oog afgelezen worden. Na het meetmoment dient de trechter geleegd te worden. Anders wordt de regendata van twee dagen bij elkaar opgeteld. De regenmeter is geïllustreerd in figuur 1.



*Figuur 1 - Regenmeter van het Delft Meet Regen project*

De neerslagmetingen van de regenmeters zijn door de deelnemers ingevoerd in de Survey123 App, samen met de desbetreffende meetlocatie, naam van het meetpunt, de datum van meten en een

foto van de meetopstelling. De neerslagmeting die doorgegeven wordt is de neerslaghoeveelheid in millimeters van de afgelopen 24 uur voor de desbetreffende meting. Deze data zal de basis vormen van de data-analyse in dit onderzoek.

Naast deze data is er ook data beschikbaar van het KNMI. Het KNMI meet op verschillende manieren, op een directe en een indirecte manier. De indirecte manier van meten wordt gedaan met de neerslagradar. Deze radar zendt pulsen uit die door mogelijke druppels weerkaatst worden. Dit geeft een momentopname van de neerslagintensiteit (KNMI, 2021).

De directe manier van meten wordt gedaan door middel van 48 automatische weerstations verdeeld over land (34 stations) en zee (14) in Nederland. Deze weerstations meten naast de neerslagintensiteit ook de luchttemperatuur, luchtvochtigheid, windsnelheid en nog meer parameters. Deze data wordt elke 10 minuten bijgewerkt (KNMI, 2021). Verder beschikt het KNMI over 322 vrijwillige neerslagwaarnemers die elke ochtend voor 9 uur de neerslag meten met handmatige regenmeters en dit digitaal doorgeven aan het KNMI. Dit is een netwerk wat verspreid zit over heel Nederland. Tot slot heeft het KNMI samen met Met Office, de Britse nationale weerdienst, het platform WOW-NL ontwikkeld. Er zijn ruim 10.000 stations in 220 landen gekoppeld aan dit netwerk (KNMI & WOW, 2021). Iedereen die beschikt over een digitaal automatisch weerstation kan zich aanmelden voor dit platform. In Delft zijn er zo ongeveer 15 weerstations die onderdeel zijn van dit netwerk, waarvan er een aantal meedoen aan het Delft Meet Regen project.

### 1.3 Doelstelling en afbakening

Het doel van dit onderzoek is om beter inzicht te krijgen in de ruimtelijke spreiding van regen in Delft. Er zal gekeken worden wat het verband is tussen de correlatie en de afstand tussen de meetpunten, dit geeft een indicatie van de ruimtelijke schaal van een bui. Om dit te onderzoeken zal de data gebruikt worden van de regenmeters van het project, Delft Meet Regen. Er zal alleen gekeken worden naar de regenspreiding in Delft voor de periode dat er gemeten is door de vrijwilligers. De gegevens die gebruikt zijn voor dit onderzoek is de neerslagdata voor de periode van 1 augustus 2021 tot en met 17 september 2021.

### 1.4 Onderzoeksvraag en deelvragen

Voor dit onderzoek is de volgende hoofdvraag opgesteld:

- Wat is de dagelijkse ruimtelijke spreiding van regen in Delft op basis van de neerslagdata van het Delft Meet Regen project?

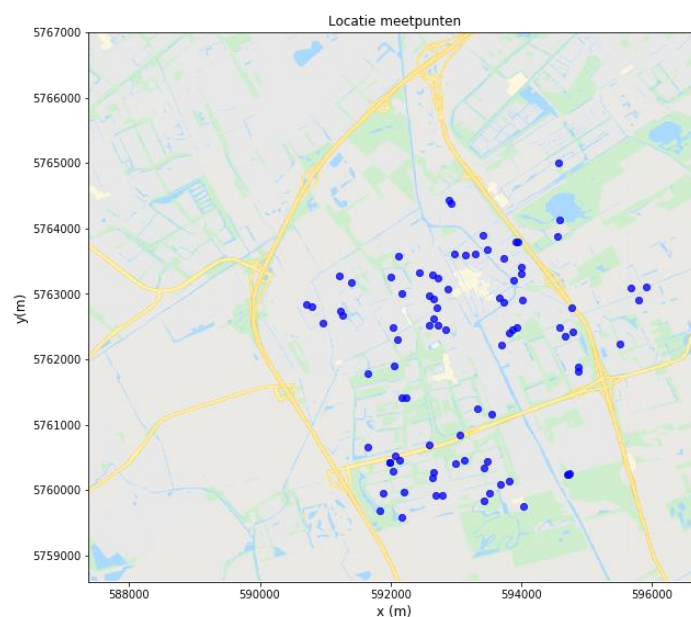
Om deze hoofdvraag te beantwoorden zijn de volgende deelvragen opgesteld:

1. Komen de neerslaggegevens van het Delft Meet Regen project overeen met die van het KNMI?
2. Is de ruimtelijke spreiding van regen te bepalen aan de hand van de vergelijking van gemiddelden voor gedefinieerde gebieden in Delft?
3. Wat zegt de ruimtelijke correlatie tussen de meetpunten over de spreiding van regen?
4. Is er een verband te vinden tussen de ruimtelijke correlatie en de afstand tussen de meetpunten?

## 2. Data

De data van het Delft Meet Regen project zijn, zoals beschreven in de introductie, aangeleverd via de SurfeY123 app. Deze data zijn vervolgens opgeslagen in een Excel-bestand. Dit Excel-bestand bevat de volgende kolommen: GlobalID, Meetpunt, Email, x, y, Datum, Neerslag (in mm), op- en aanmerkingen. De GlobalID kolom bevat per meting een unieke code. Deze kolom is voor het onderzoek niet nodig, dus is uit het Excel bestand verwijderd. De kolom Email is ook niet relevant voor het onderzoek, dus deze kolom is ook verwijderd. De op- en aanmerkingenkolom is in eerste instantie ook verwijderd, aangezien deze voor de data-analyses niet nodig is. De vrijwilligers konden in deze kolom bijzonderheden omtrent de meting melden

De x- en y-coördinaten van de locaties van de meetpunten zijn weergegeven in de geografische coördinaten lengtegraad en breedtegraad in graden. De afstanden tussen de meetpunten zullen in dit onderzoek nodig zijn in de data analyse. Om deze afstanden makkelijker uit te rekenen zijn de geografische coördinaten omgezet in cartesische coördinaten. Dit is een projectie van de lengte- en breedtegraad-coördinaten op het x- en y-vlak. Deze omzetting is gedaan door middel van Matlab, de Matlab code hiervan is weergegeven in Bijlage B (Beauducel, 2019). De locaties van alle meetpunten in Delft zijn weergegeven in figuur 2. De oorsprong van dit coördinaten systeem ligt dus op het punt waar de lengtegraad en breedtegraad nul zijn.



*Figuur 2 – Locaties van de gebruikte meetpunten van het Delft Meet Regen project*

Vervolgens zijn alle mogelijke fouten uit het Excel bestand verwijderd. Ten eerste zijn sommige meetpuntnamen niet consequent op dezelfde manier doorgegeven, door bijvoorbeeld een spelfout of een spatie. Dit leidt tot moeilijkheden bij de data-analyse. Ten tweede zijn er foutieve neerslagmetingen doorgegeven, door bijvoorbeeld een o (letter) in plaats van een 0 (cijfer) in te vullen of door tekst achter de meting te plaatsen. Het decimaalscheidingsteken was ook wisselend een punt of een komma. Deze zijn allemaal omgezet in een punt (Python heeft als decimale scheidingsteken een punt, in het verslag is het decimaalteken een komma) en foutieve neerslagmetingen zijn verbeterd. Ook werden er dubbele metingen doorgegeven. Tot slot zijn er ook afwijkende locaties doorgegeven, die zich niet in Delft bevonden. Het zou kunnen zijn dat

vrijwilligers de neerslagmeting op een andere locatie dan de regenmeter invulden. Deze locaties zijn aangepast naar de locatie van de juiste regenmeter om te voorkomen dat er data verloren gaat.

Na het aanpassen van de ruwe data is het Excel bestand opgeslagen als een CSV bestand. Deze data vormen de input van de data-analyses voor dit onderzoek. De data-analyses zullen met Python verwerkt worden. Het CSV bestand wordt ingeladen met het Pandas pakket van Python. De datums van de metingen staan op chronologische volgorde. Het totaal aantal metingen is gelijk aan 2708. Er zijn 90 meetpunten die 48 dagen zouden kunnen meten. Als er door elk meetpunt elke dag gemeten zou zijn, dan zouden er in totaal 4320 dag metingen zijn. De databeschikbaarheid is dus 63%,  $\left(\frac{2708}{4320}\right) \cdot 100\%$ . Het aantal regendagen is gelijk aan 34. Hierin is een regendag een dag waarbij het dagelijkse gemiddelde van alle neerslagmetingen groter is dan 0. Een voorbeeld van de data van de eerste zes metingen van twee meetpunten is weergegeven in tabel 1 en tabel 2. De locaties van deze meetpunten zijn weergegeven in figuur 3 en de afstand tussen deze meetpunten is 3555 m. Hierin is de neerslag in millimeters en de x en y-coördinaten in meters. De neerslagmeting geeft de neerslag weer van 24 uur voor de meting. De x en y coördinaten verschillen per dag meting per meetpunt. Dit komt doordat de vrijwilligers niet consequent dezelfde locatie ingevuld hebben.

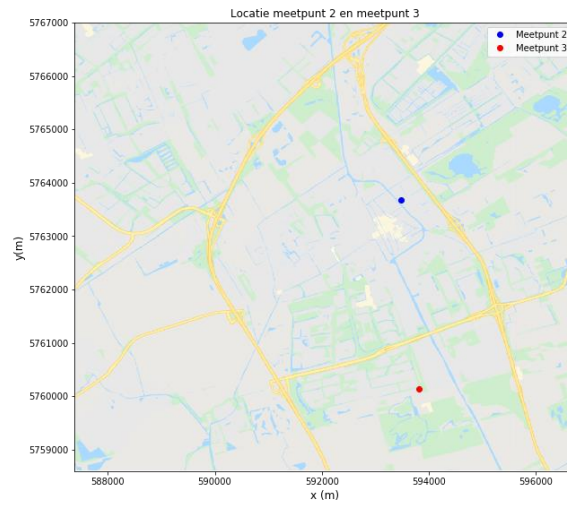
*Tabel 1 – Data van eerste 6 metingen van meetpunt 2*

Meetpuntnummer	Datum	Neerslag (mm)	y (m)	x (m)
2	09-08-2021	1,0	5763685	593480
2	10-08-2021	7,0	5763683	593412
2	11-08-2021	0,5	5763698	593413
2	12-08-2021	0,0	5763680	593414
2	14-08-2021	0,0	5763680	593414
2	15-08-2021	0,0	5763696	593433

*Tabel 2 – Data van eerste 6 metingen van meetpunt 3*

Meetpuntnummer	Datum	Neerslag (mm)	y (m)	x (m)
3	14-08-2021	0,0	5763685	593480
3	15-08-2021	0,0	5763683	593412
3	28-08-2021	0,3	5763698	593413
3	29-08-2021	0,0	5763680	593414
3	30-08-2021	0,0	5763680	593414
3	31-08-2021	0,4	5763696	593433





*Figuur 3 – Locatie van meetpunt 2 en meetpunt 3*

### 3. Methodologie

In dit hoofdstuk wordt de methodologie van het onderzoek besproken. Ten eerste zal de data van het Delft Meet Regen project geverifieerd worden met de KNMI data van station 449 in Delft (KNMI, 2021). Dit station valt onder het vrijwilligers netwerk van de KNMI, die zich bevindt in de Botanische Tuin in Delft. Dit is de 24-uurs som van de neerslag. Naast deze verificatie zal ook een algemeen beeld gegeven worden van de Delft Meet Regen data. Het dagelijkse gemiddelde (ook cumulatief), het aantal metingen en de standaard afwijking zal worden geplot. Het dagelijkse gemiddelde zal vergeleken worden met de 24-uurs sommen van het KNMI, ook voor de cumulatieve plot. Het gemiddelde ( $\bar{x}$ ) is als volgt berekend, met  $n$  het aantal metingen ( $x_1, x_2, \dots, x_n$ ):

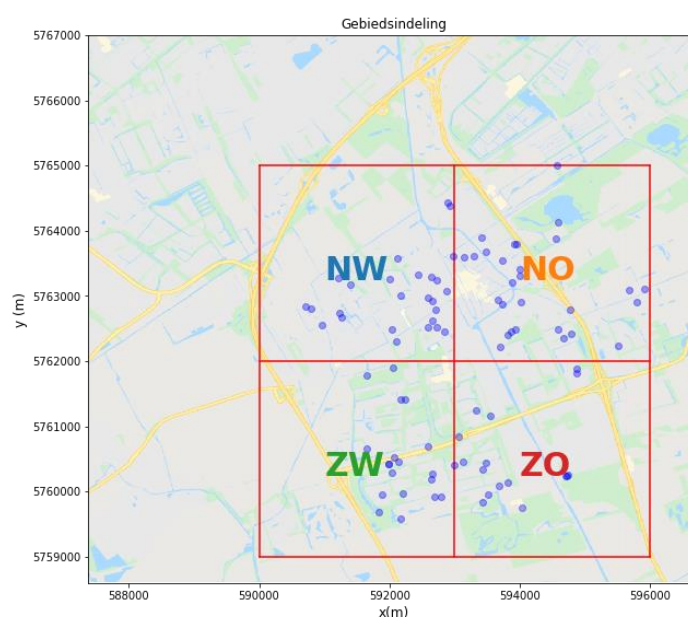
$$\bar{x} = \frac{1}{n} \sum_i x_i \quad (1)$$

De standaard afwijking ( $s$ ) van de dataset zal een eerste indicatie geven over de spreiding van de neerslag. De standaard afwijking is gedefinieerd als de wortel van de variantie en kan met onderstaande formule berekend worden (Dekking et al, 2015):

$$s^2 = \frac{1}{n-1} \sum_i (x_i - \bar{x})^2 \quad (2)$$

Het dagelijks aantal meetpunten ( $n$ ) heeft een invloed op de nauwkeurigheid van het gemiddelde en de standaard afwijking, dit is inclusief metingen die een neerslagmeting van nul hebben. Hoe meer meetpunten er zijn, des te nauwkeuriger de uitkomsten.

Na de dataverificatie zullen er twee analyses besproken worden in dit onderzoek. De gebiedsanalyse en de ruimtelijke correlatieanalyse. In de gebiedsanalyse zullen de meetpunten verdeeld worden in vier gebieden: noordwest (NW), noordoost (NO), zuidwest (ZW) en zuidoost (ZO). Er is gekozen voor vier gebieden, zodat de meetpunten zo gelijk mogelijk verdeeld zijn en de oppervlaktes gelijk aan elkaar zijn. De oppervlakte van een gebied is gelijk aan negen vierkante kilometer. De vier gebieden zijn weergegeven in figuur 4.



Figuur 4 – Meetpunten ingedeeld in de gebieden Noordwest, Noordoost, Zuidwest en Zuidoost

Van deze gebieden zal ook het gemiddelde, het aantal metingen en de standaard afwijking uitgerekend worden. Deze waarden voor de gebieden zullen weergegeven worden voor de hele datareeks. Hiernaast worden de dagelijkse neerslagsommen die gemiddeld minimaal 1 mm groot zijn weergegeven in staafdiagrammen. De dagen waarbij er veel verschil zit tussen de gebieden zullen op de kaart worden weergegeven. Op deze manier kunnen de gebieden met elkaar worden vergeleken op basis van de bovengenoemde statistische parameters en kan er onderzocht worden wat de verschillen zijn en of er sprake is van eventuele dagelijkse neerslagspreiding.

Na de bovengenoemde gebieden-analyse zal de ruimtelijke correlatieanalyse uitgevoerd worden. Tussen elke mogelijke combinatie van twee meetpunten zal de afstand en correlatiecoëfficiënt uitgerekend worden. De unieke mogelijke combinaties tussen de meetpunten wordt met onderstaande formule uitgerekend (Dekking et al, 2015), met  $n$  het totaal aantal meetpunten en  $k$  de grootte van de groepen. In dit geval willen we het aantal unieke combinaties van twee meetpunten uit 90 meetpunten,  $\binom{90}{2} = 4005$  unieke combinaties.

$$\binom{n}{k} = \frac{n!}{(n-k)! \cdot k!} \quad (3)$$

De correlatiecoëfficiënt is een maat voor de sterkte van de lineaire samenhang tussen twee variabelen. In dit geval zijn dat de neerslagmetingen van twee meetpunten die op dezelfde dag meten. De correlatiecoëfficiënt die gebruikt gaat worden in dit onderzoek is Pearsons correlatiecoëfficiënt ( $\rho$ ). Hierbij wordt de covariantie tussen twee variabelen gedeeld door het product van de standaardafwijkingen van deze variabelen:

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sigma(x) \cdot \sigma(y)} \quad (4)$$

Als de correlatie uitgerekend wordt tussen twee tijdreeksen die verkregen zijn uit metingen wordt de Pearson correlatiecoëfficiënt als volgt uitgerekend:

$$\rho(x, y) = \frac{\sum_i (x_i - \bar{x}) \cdot (y_i - \bar{y}) / (n-1)}{s(x) \cdot s(y)} \quad (5)$$

De waarde van deze correlatiecoëfficiënt ligt tussen de -1 en de 1. Een correlatiecoëfficiënt van 1 betekent dat er een positief lineair verband bestaat tussen de variabelen. Als de correlatiecoëfficiënt -1 is betekent dat de variabelen negatief lineair samenhangen. Als de coëfficiënt gelijk is aan nul betekent dat er geen lineaire samenhang bestaat (Dekking et al, 2015).

Een hoge correlatie tussen een reeks neerslagmetingen van twee meetpunten betekent dat er een grote kans is dat deze meetpunten vergelijkbare hoeveelheden neerslag meten. Als de correlatie dus hoog is tussen twee meetpunten, worden ze door dezelfde bui getroffen. Hoe lager de correlatie hoe groter de kans op neerslagverschillen tussen de meetpunten. Dit leidt dat een grotere kans op neerslagspreiding. Als neerslagmetingen die sterk gecorreleerd zijn tegen elkaar zouden worden uitgezet, zullen de punten in de buurt liggen van de regressielijn door deze punten. Hoe dichter de metingen bij de  $y = x$  lijn liggen, des te kleiner de verschillen tussen de metingen zijn.

Zoals eerder genoemd zal tussen elke combinatie (maximaal 4005 resultaten) van meetpunten de correlatie en afstand berekend worden. Aangezien niet alle meetpunten op dezelfde momenten meten, zal er eerst worden gekeken wanneer de meetpunten beide een neerslagmeting hebben. Alleen tussen deze metingen zal de correlatie berekend worden. Als beide metingen een neerslagmeting van nul hebben (dus beide geen regen meten) zullen deze metingen niet meetellen

met de correlatie berekening. Dit zou anders leiden tot een vertekend beeld van de uiteindelijke correlatie. Dit is in een functie verwerkt in Python. Deze functie kan de correlatie uitrekenen met een bepaalde drempelwaarde. De drempelwaarde is het aantal neerslagmetingen (dagen) waarvoor de correlatiecoëfficiënt uitgerekend wordt. Er is minimaal een drempelwaarde van twee nodig om de correlatie uit te rekenen. De correlaties zijn uitgerekend met een drempelwaarde van vijf en een drempelwaarde van tien. Hoe hoger de drempelwaarde is des te nauwkeuriger de berekende correlatie wordt. De Pearson-functie rekent voor alle combinaties tussen de meetpunten de correlatie en de afstand uit. Per meetpunt kan de locatie van de dag metingen verschillen. Dit is besproken in het hoofdstuk Data. Tussen de meetpunten wordt daarom de gemiddelde afstand uit gerekend. De correlaties en bijbehorende afstanden zijn opgeslagen in arrays.

Hierna zullen de afstanden en bijbehorende correlaties tegen elkaar uitgezet worden in een spreidingsdiagram. Negatieve correlaties zijn lastig te interpreteren in deze situatie dus deze correlaties zijn niet meegenomen in het spreidingsdiagram en de verdere analyse. Naast het spreidingsdiagram zal er een dichtheidshistogram geplot worden van alle correlaties voor alle afstanden. Hierin zal ook het gemiddelde van de correlaties en het 10<sup>e</sup> en 90<sup>e</sup> percentiel weergegeven worden. De percentielen zullen dienen als betrouwbaarheidsinterval.

Om het spreidingsdiagram verder te analyseren worden de afstanden in vier intervallen opgedeeld. Interval 1 is van afstand 0 m tot en met 500 m, interval 2 is van 500 m tot en met 1500 meter, interval 3 is van 1500 m tot en met 3000 m en interval 4 is van 3000 meter tot en met 7000 meter. Zo worden de correlaties onder verdeeld in deze intervallen. Van deze intervallen worden ook weer dichtheidshistogrammen gemaakt van de correlaties. Hier zal ook weer het gemiddelde, het 10<sup>e</sup> en het 90<sup>e</sup> percentiel van berekend worden. Hierbij wordt ook de kansdichtheidsfunctie samen met de dichtheidshistogram geplot. Deze gemiddelden van de correlaties zullen uitgezet worden tegen de gemiddelde afstand in het spreidingsdiagram. Dit zal ook gedaan worden voor de 10<sup>e</sup> en 90<sup>e</sup> percentielen. Hiervoor zal ook de gemiddelde afstand per interval berekend moeten worden. Zo wordt er per interval dus een zwaartepunt weergegeven in het spreidingsdiagram. Om de dichtheidshistogrammen van de intervallen goed te kunnen vergelijken met elkaar worden de kansdichtheidsfuncties behorend bij deze dichtheidshistogrammen weergegeven in één figuur.

Om een verband te vinden tussen de afstand en de correlatie wordt er een exponentiële curve door de bovengenoemde zwaartepunten van de intervallen getrokken. Deze curve wordt ook door de gemiddelde waardes van de 10<sup>e</sup> en 90<sup>e</sup> percentielen gefit. De parameters worden met de curvefit-functie van Python berekend door middel van de kleinste kwadraten methode. Het exponentiële verband wat gezocht wordt is weergegeven met onderstaande formule (Van Leth et al., 2021):

$$\rho(d) = \exp\left(-\left(\frac{d}{d_0}\right)^{s_0}\right) \quad (6)$$

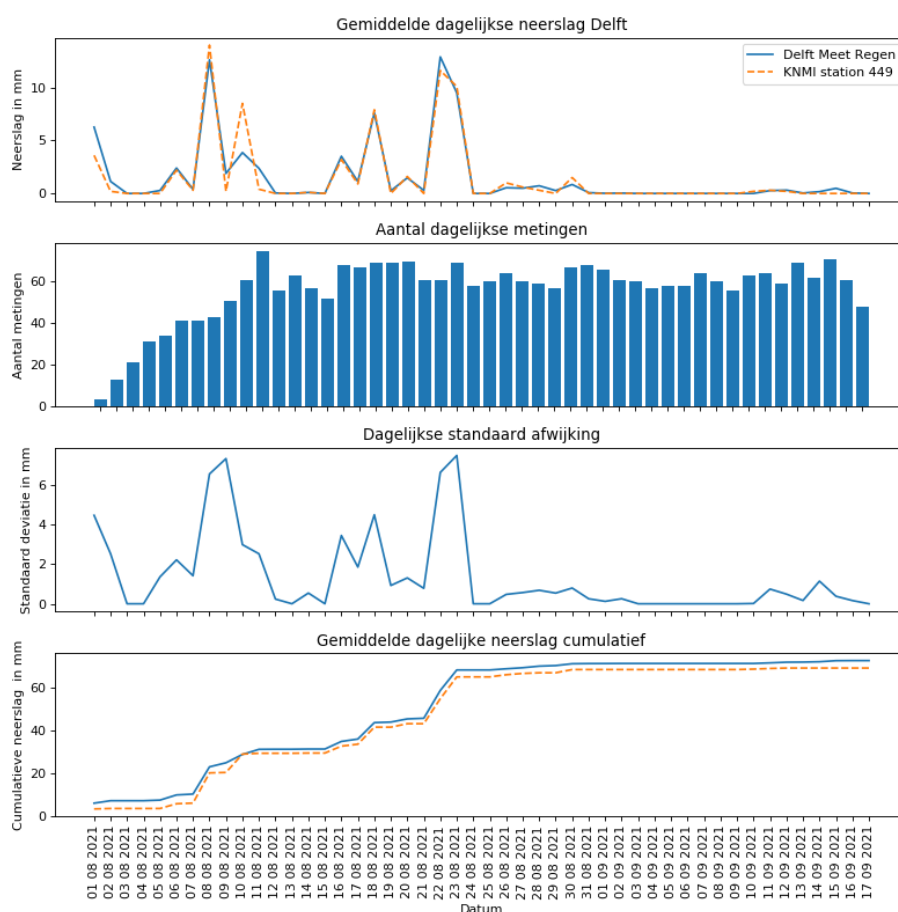
De parameters  $d_0$  en  $s_0$  zullen bepaald worden met boven genoemde methode. Hiermee kan de exponentiële functie geplot worden door de zwaartepunten van de correlaties heen. De waarde van  $d_0$  is de afstand waarbij de correlatie gelijk is aan  $1/e$  en  $s_0$  is de vormparameter (Van Leth et al., 2021).

## 4. Resultaten

In dit hoofdstuk worden de resultaten besproken van het onderzoek. Eerst zal de data verificatie besproken worden, daarna de gebiedenanalyse en als laatste de ruimtelijke correlatieanalyse.

### 4.1 Data verificatie

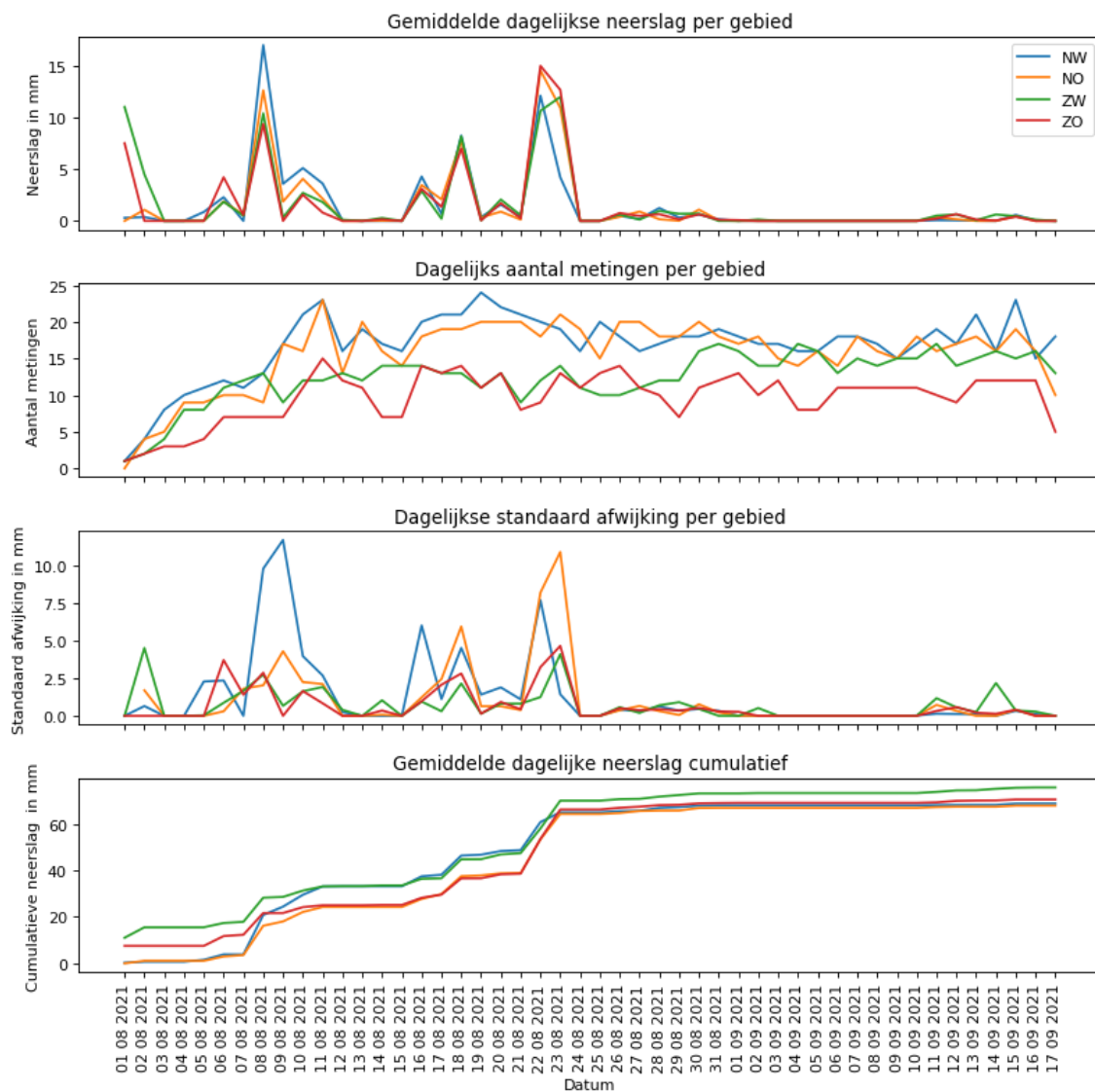
In figuur 5 is de gemiddelde dagelijkse neerslag (ook cumulatief) van de data van het Delft Meet Regen project geplot samen met de gemiddelde dagelijkse neerslag van KNMI station 449. Ook is het aantal metingen en de dagelijkse standaard afwijking geplot van de data van Delft Meet Regen. De dagelijkse gemiddelde neerslag van Delft Meet Regen komt in het algemeen overeen met de KNMI data. De regen volgens de KNMI data heeft op 08-08-2021 een iets hogere piek met een verschil van 1,38 mm en op 10-08-2021 is het verschil tussen de pieken 4,64 mm. Op 22-08-2021 heeft Delft Meet Regen een hogere piek van 1,29 mm. Op de andere dagen zijn de verschillen minimaal. Bij de cumulatieve plot liggen de grafieken dicht bij elkaar en is er in totaal gemiddeld 3,45 mm meer regen gemeten door de Delft Meet Regen meters. De eerste 8 dagen (tot 9 augustus) loopt het aantal op tot 50 metingen. Daarna zit het aantal metingen boven de 50, alleen de laatste dag zijn er 48 metingen. Tot slot gaat de standaard afwijking van Delft Meet Regen data omhoog als de dagelijkse gemiddelde neerslag ook omhoog gaat.



Figuur 5 – Gemiddelde dagelijkse neerslag (ook cumulatief) van Delft Meet Regen en KNMI station 449 en het aantal metingen en de standaard afwijking van Delft Meet Regen data

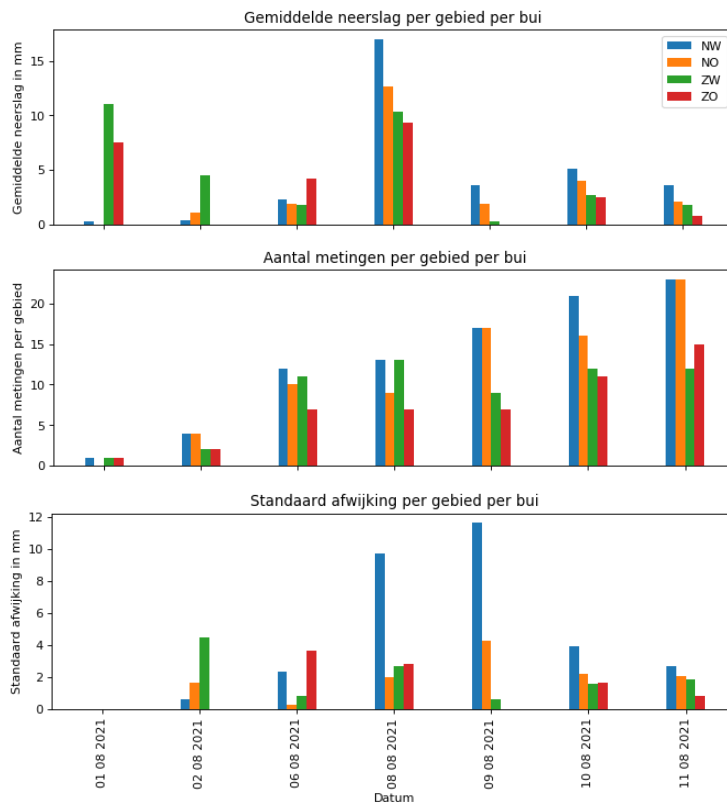
## 4.2 Gebiedenanalyse

Zoals besproken in hoofdstuk 3 zijn de meetpunten onder verdeeld in vier gebieden zoals weergegeven in figuur 4. In figuur 6 is de gemiddelde dagelijkse neerslag, het dagelijkse aantal metingen, de dagelijkse standaard afwijking en de gemiddelde dagelijkse neerslag cumulatief per gebied weergegeven.

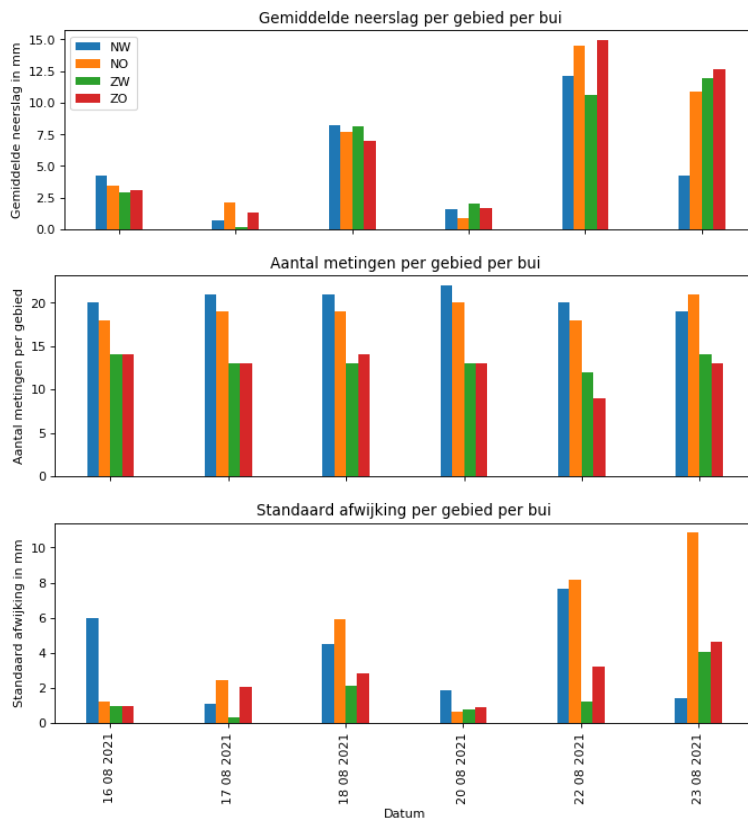


Figuur 6 – Gemiddelde dagelijkse neerslag, dagelijks aantal metingen, dagelijkse standaard afwijking en gemiddelde dagelijkse neerslag cumulatief van de gedefinieerde gebieden

In figuur 7 en figuur 8 zijn de dagelijkse neerslagsommen die gemiddeld groter zijn dan 1 mm weergegeven in staafdiagrammen. Hiervan is de gemiddelde neerslag, het aantal metingen en de standaard afwijking per gebied per bui weergegeven.



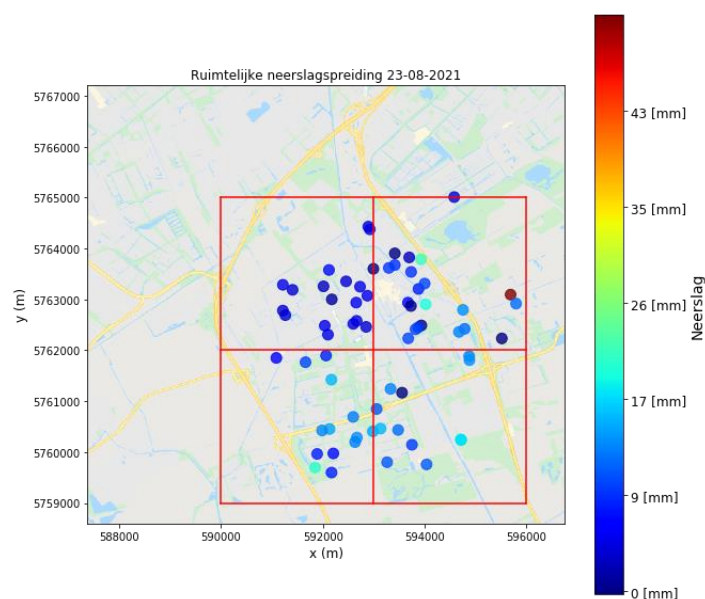
Figuur 7 – De gemiddelde neerslag, aantal metingen en standaard afwijking per gebied van de eerste zeven dagen met meer dan 1 mm regen



*Figuur 8 – De gemiddelde neerslag, aantal metingen en standaard afwijking per gebied van de laatste zes dagen met meer dan 1 mm regen*

Uit figuur 6 is af te leiden dat er gemiddeld gezien minder metingen zijn in de gebieden ZW en ZO. Verder is er te zien dat er op de dagen dat het gemiddeld gezien veel regent er duidelijk verschillen optreden in de hoeveel gemiddelde gemeten neerslag tussen de gebieden. Op deze dagen is de standaard afwijking van de gebieden NW en NO ook hoger dan de standaard afwijking van de gebieden ZW en ZO. Cumulatief gezien is gemiddeld de meeste regen gevallen in gebied ZW, in totaal is hier gemiddeld 75,7 mm gevallen. Het aantal metingen op 01-08-2021 en 02-08-2021 is laag en op deze twee dagen wordt er voor gebied ZW respectievelijk gemiddeld 11 mm en 4 mm gemeten. Op deze dagen wordt er in de andere gebieden gemiddeld weinig regen gemeten. Op 08-08-2021 wordt gemiddeld gezien de meeste regen gemeten in gebied NW (17,0 mm), op deze dag zijn de verschillen in de gemiddelde neerslag groot tussen de gebieden. Ook is op deze dag de standaard afwijking van gebied NW (9,8 mm) beduidend hoger dan de andere gebieden. Op 09-08-2021 is de standaard afwijking van gebied NW (11,7 mm) ook veel hoger dan de standaard afwijking in de andere gebieden. Op deze dag zijn er in gebied ZO wel metingen maar is de gemiddelde gemeten neerslag nul, dit betekent dat er alleen nulmetingen zijn en dat de standaard afwijking dan ook nul is. Op 22-08-2021 en 23-08-2021 is er gemiddeld ook veel neerslag gemeten. Op beide dagen wordt gemiddeld het meeste gemeten in gebied ZO, respectievelijk 15,0 mm en 12,7 mm. Verder valt op dat de standaard afwijking op 23-08-2021 in gebied NO (10,9 mm) hoger is dan de andere gebieden en dat er in gebied NW gemiddeld weinig regen (4,2 mm) gemeten wordt ten opzichte van de andere gebieden.

In figuur 9 zijn de metingen van alle meetpunten op 23-08-2021 ruimtelijk weergegeven. In figuur 9 is duidelijk te zien dat in gebied NW de minste regen valt. Hier zijn de metingen donkerblauw gekleurd (0-9 mm). De metingen zijn hier vrij uniform, wat betekent dat de standaard afwijking in dit gebied laag is (1,43 mm). In de andere gebieden zijn de gemiddelde neerslagmetingen hoger en zijn ook standaard afwijkingen hoger. In het gebied NO valt de meting van 50 mm op, hierdoor is de standaard afwijking van dit gebied 10,9 mm. De overige ruimtelijke plots van de andere dagen waarbij de neerslagsom gemiddeld groter is dan 1 mm zijn weergegeven in Bijlage A.

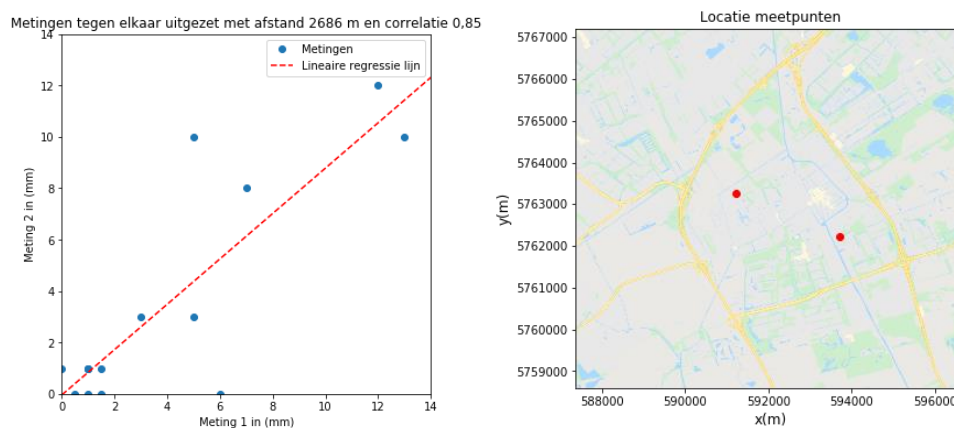


*Figuur 9 – Neerslag metingen alle meetpunten op 23-08-2021*

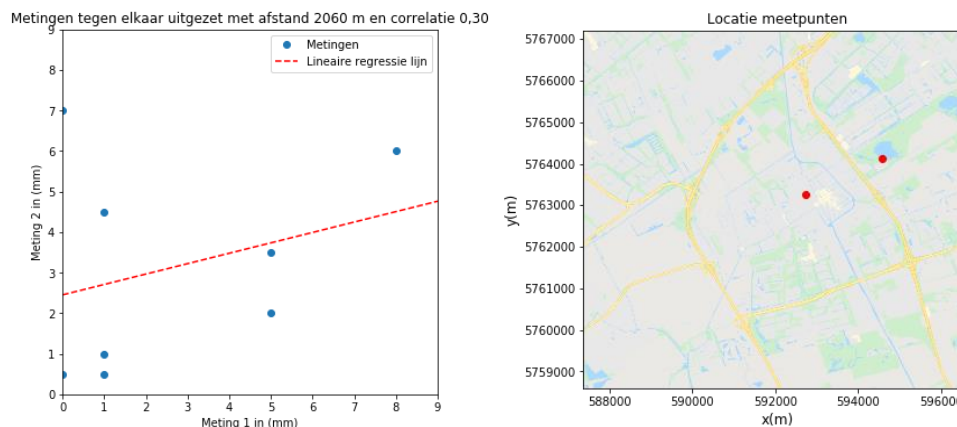


### 4.3 Ruimtelijke correlatie analyse

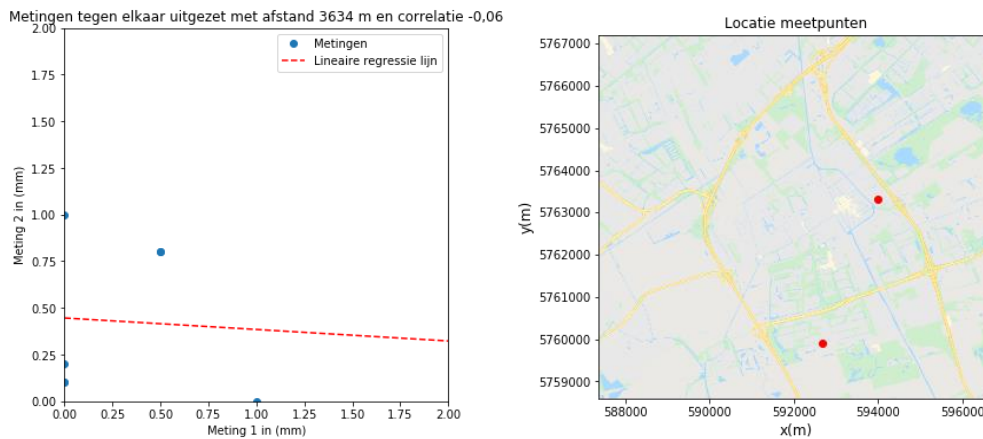
In figuur 10, 11 en 12 zijn voorbeelden weergegeven van metingen van twee meetpunten die tegen elkaar uitgezet zijn. Door deze metingen is ook de regressielijn geplot. Naast deze figuren zijn ook de locaties van de meetpunten weergegeven. In figuur 10 is de onderlinge afstand tussen de meetpunten gelijk aan 2686 meter en is de correlatie 0,85. Het aantal metingen waartussen de correlatie uitgerekend is, is 13. In figuur 11 is de onderlinge afstand 2060 meter en de correlatie 0,30 met 7 metingen. Tot slot is in figuur 12 een negatieve correlatie weergegeven van -0,06 met een onderlinge afstand van 3634 meter. Hier zijn er slechts 5 metingen.



*Figuur 10 – Metingen van twee meetpunten tegen elkaar uitgezet met onderlinge afstand van 2686 m en een berekende correlatie van 0,85*

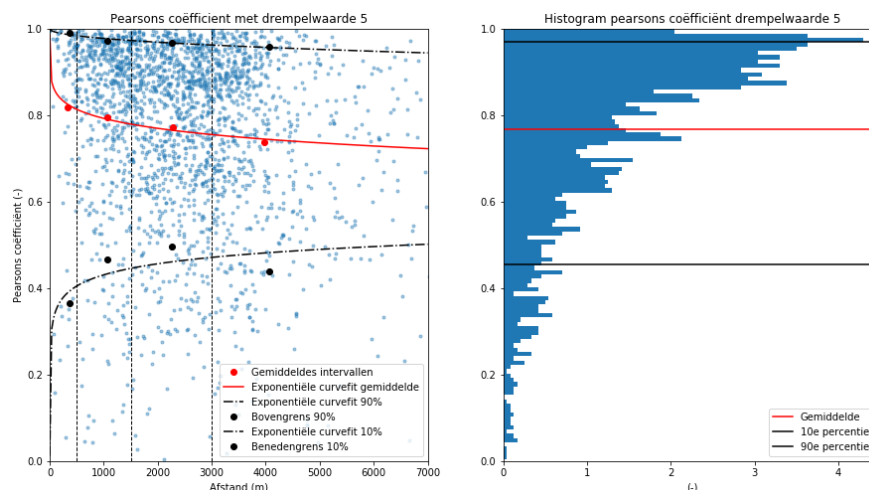


*Figuur 11 – Metingen van twee meetpunten tegen elkaar uitgezet met onderlinge afstand van 2060 m en een berekende correlatie van 0,30*

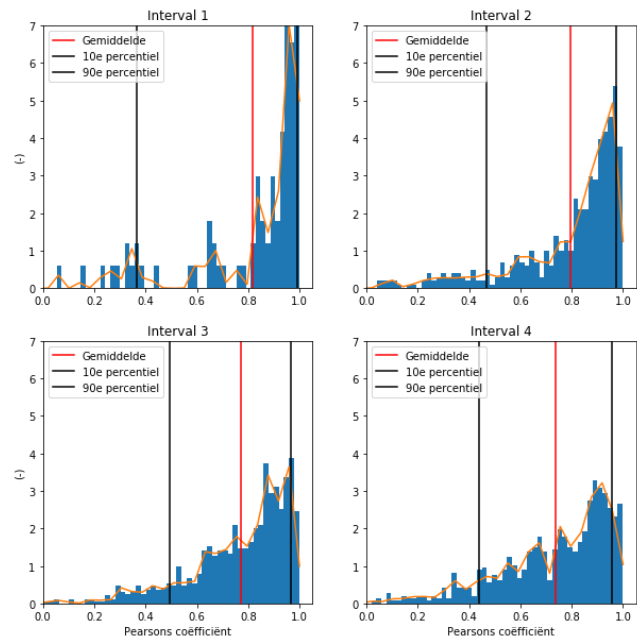


Figuur 12 – Metingen van twee meetpunten tegen elkaar uitgezet met onderlinge afstand van 3634 m en een berekende correlatie van -0,06

Vervolgens is de afstand tegen de correlatie uitgezet in een spreidingsdiagram, voor het uitrekenen van de correlaties is ten eerste een drempelwaarde van 5 gehanteerd. Daarnaast is ook een dichtheidshistogram voor alle correlaties weergegeven. Dit is geïllustreerd in figuur 13. De negatieve correlaties zoals weergegeven in figuur 12 zijn niet meegenomen in dit spreidingsdiagram. Er zijn 2534 correlaties uitgerekend, waaronder 127 negatieve correlaties. Het gemiddelde van de correlaties is 0,77 met als ondergrens (10<sup>e</sup> percentiel) 0,46 en als bovengrens (90<sup>e</sup> percentiel) 0,97. De intervallen zijn afgebakend met de verticale zwarte stippellijnen. De zwaartepunten van de intervallen gaan lager liggen naarmate de afstand toeneemt. De exponentiële functie die beschreven is met formule (6) is door de zwaartepunten heen gefit. De waarde van  $d_0$  is hierbij gelijk aan 4476 m en de waarde van  $s_0$  is gelijk aan 0,17. De gemiddelde correlaties per interval met 10<sup>e</sup> en 90<sup>e</sup> percentiel zijn weergegeven met dichtheidshistogrammen en bijbehorende kansdichtheidsfuncties in figuur 14.

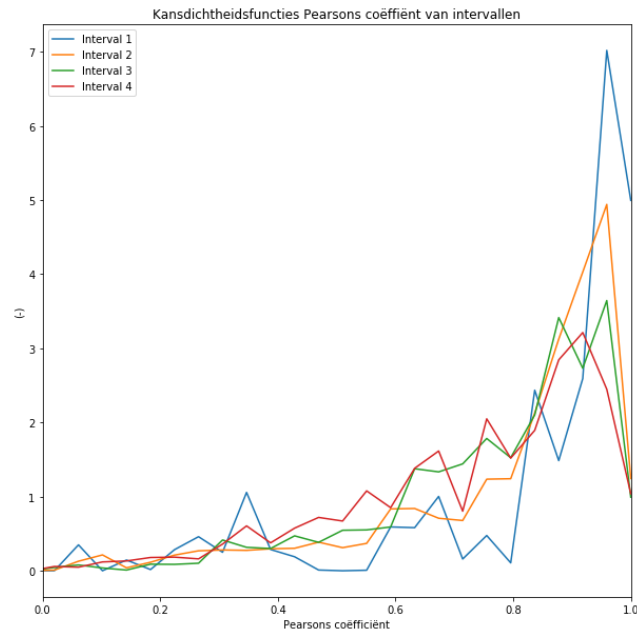


Figuur 13 – 1. Spreidingsdiagram van correlaties met gemiddelde, 10<sup>e</sup> percentielen en 90<sup>e</sup> percentielen van de correlaties per interval met drempelwaarde 5; 2. Dichtheidshistogram met gemiddelde, 10<sup>e</sup> percentiel en 90<sup>e</sup> percentiel van alle correlaties met drempelwaarde 5



Figuur 14 – Dichtheidshistogram en kansdichtheidsfunctie van de correlaties per afstand interval met gemiddelde, 10<sup>e</sup> percentiel en 90<sup>e</sup> percentiel van deze correlaties met drempelwaarde 5

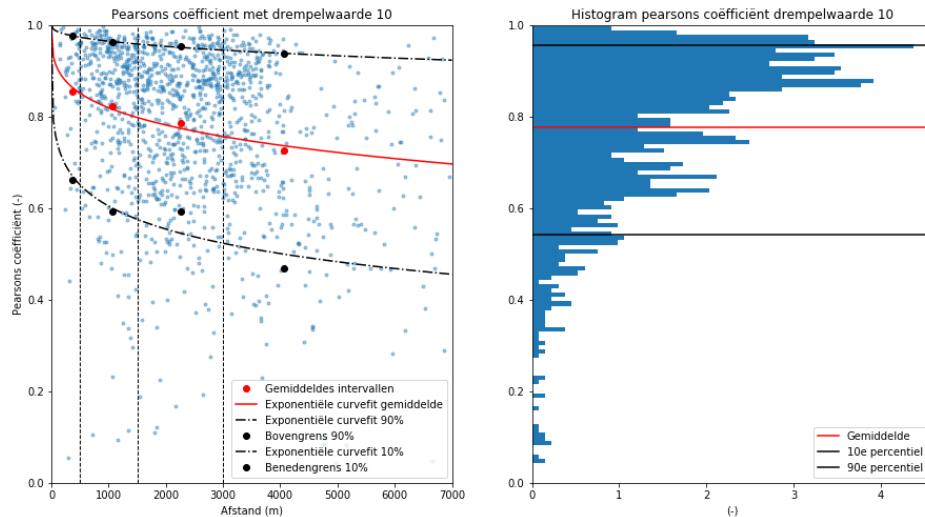
De gemiddelden zijn respectievelijk per interval: 0,82; 0,79; 0,77; 0,74. De gemiddelde correlatie per interval neemt dus af naarmate de afstand tussen de meetpunten toeneemt. In figuur 15 zijn de kansdichtheidsfuncties van de correlaties tegen elkaar uitgezet per interval. Hier valt duidelijk te zien dat de top van de grafiek lager gaat liggen naarmate de afstand groter wordt.



Figuur 15 – Kansdichtheidsfuncties van de correlaties per interval met drempelwaarde 5

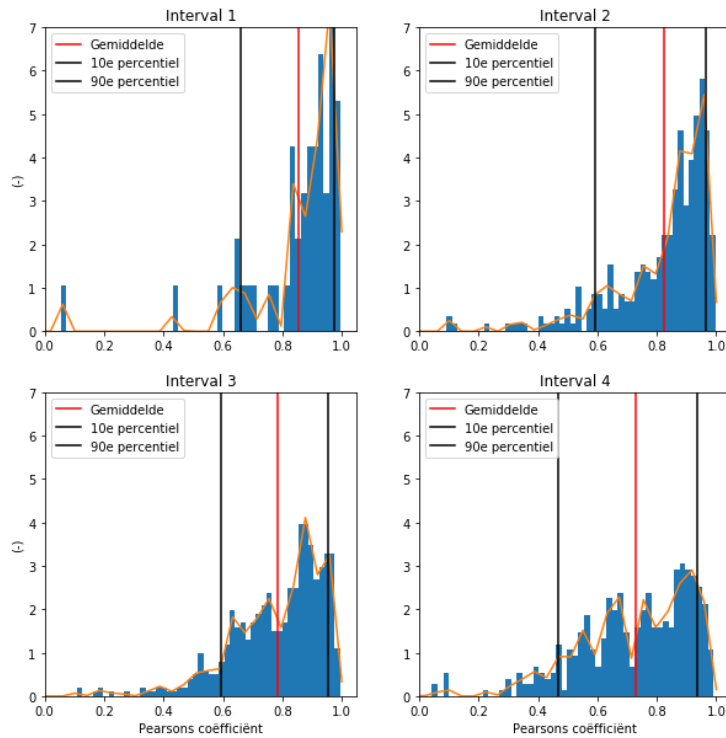
De resultaten van het spreidingsdiagram en dichtheidshistogram van de correlaties die berekend zijn met drempelwaarde 10 zijn weergegeven in figuur 16. Het totaal aantal uitgerekende correlaties is gelijk aan 1401 (van de 4005 unieke momentenparen) en het aantal negatieve correlaties is gelijk aan 10. Het gemiddelde van de correlaties is 0,78 met als gemiddelde ondergrens 0,54 en als gemiddelde bovengrens 0,96. Er zijn dus 1033 minder correlaties berekend vergeleken met

drempelwaarde 5 en er zijn 117 minder negatieve correlaties berekend. Het gemiddelde van de correlatie is 0,01 hoger, het gemiddelde van de ondergrens is 0,08 hoger en het gemiddelde van de bovengrens is 0,01 lager. De  $d_0$  waarde van de exponentiële fit door de zwaartepunten is gelijk aan 193 m en de waarde voor  $s_0$  is gelijk aan 0,31. De ondergrens is stukken hoger aangezien er minder negatieve correlaties uitgerekend worden, omdat de drempelwaarde hoger is. Verder is het verschil in  $d_0$  opmerkelijk, namelijk 4283 m.



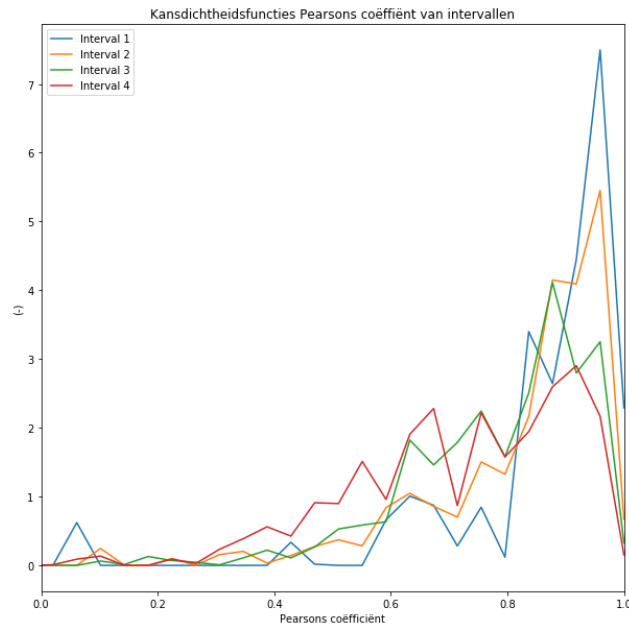
Figuur 16 – 1. Spreidingsdiagram van correlaties met gemiddeldes, 10<sup>e</sup> percentielen en 90<sup>e</sup> percentielen van de correlaties per interval met drempelwaarde 10; 2. Dichtheidshistogram met gemiddelde, 10<sup>e</sup> percentiel en 90<sup>e</sup> percentiel van alle correlaties met drempelwaarde 10

De dichtheidshistogrammen met bijbehorende kansdichtheidsfuncties van de correlaties met drempelwaarde 10 per afstandsinterval zijn weergegeven in figuur 17. De gemiddelden van de correlaties per interval zijn respectievelijk: 0,86; 0,82; 0,79; 0,73. Voor interval 1, 2 en 3 ligt het correlatie gemiddelde hoger dan bij drempelwaarde 5. Alleen bij interval 4 is het gemiddelde 0,01 lager.



*Figuur 17 – Dichtheidshistogram en kansdichtheidsfunctie van de correlaties per afstand interval met gemiddelde, 10<sup>e</sup> percentiel en 90<sup>e</sup> percentiel van deze correlaties met drempelwaarde 10*

De kansdichtheidsfuncties van de correlatie per interval met drempelwaarde 10 zijn weergegeven in figuur 18. Ook hier is te zien dat de toppen van de grafieken lager gaan liggen naarmate de afstand groter wordt. Dit betekent dat de correlatie afneemt naarmate de afstand groter wordt.



*Figuur 18 – Kansdichtheidsfuncties van de correlaties per interval met drempelwaarde 10*

## 5. Discussie

In hoofdstuk 2 (Data) zijn de fouten en de onvolledigheid van de data van het Delft Meet Regen project besproken. De volledigheid van de data is 63% (dit is het werkelijk aantal metingen gedeeld door het maximaal aantal metingen tot en met 17 september). Daarnaast verschilt het aantal metingen per meetpunt. Dit heeft van invloed gehad op de methodiek en resultaten. In 4.1 (Dataverificatie, figuur 5) is te zien dat het aantal metingen per dag verschilt en dat vooral in de beginperiode (01-08-2021 tot en met 09-08-2021) het aantal metingen laag is. Hierdoor waren het aantal metingen in de gebiedenanalyse ook niet aan elkaar gelijk. Ook heeft het aantal metingen invloed op de uitkomsten van de gemiddelden en standaard afwijkingen van de gebieden. Het lage aantal metingen zorgt ervoor dat de resultaten minder nauwkeurig zijn. Dit is bijvoorbeeld goed te zien op 01-08-2021, waar het totaal aantal metingen drie is, gebied NO heeft hier geen meting. De meting van gebied ZW heeft hier één meting die 11 mm is, hierdoor is het gemiddelde ook 11 mm en is de standaard afwijking gelijk aan nul. Deze waarden van de gemiddelden van de gebieden hebben weer invloed op de cumulatieve waarden.

Het aantal metingen heeft ook invloed gehad de ruimtelijke correlatieanalyse en de resultaten hiervan. Hoe hoger de drempelwaarde gelegd wordt van het uitrekenen van de correlaties, hoe nauwkeuriger deze correlaties worden. Dit is ook terug te zien in de resultaten. Bij een drempelwaarde van 5 worden er 2534 correlaties uitgerekend, waarvan 127 negatieve correlaties. Bij een drempelwaarde van 10 worden er 1401 correlaties uitgerekend, waarvan er 10 negatief zijn. Bij een hogere drempelwaarde worden er dus minder correlaties uitgerekend en ook minder negatieve correlaties (bijna nul negatieve correlaties). De gemiddelde ondergrens (10<sup>e</sup> percentiel) van de correlaties met drempelwaarde 10 ligt ook 0,08 hoger dan de gemiddelde ondergrens van de correlaties met drempelwaarde 5. Hierdoor kunnen de correlaties met drempelwaarde 10 als nauwkeuriger en betrouwbaarder beschouwd worden. Maar als de drempelwaarde te hoog wordt gelegd worden er te weinig correlaties uitgerekend om resultaten te zien. Deze analyse is namelijk ook uitgevoerd met een drempelwaarde van 20, maar hier werden te weinig correlaties uitgerekend om goede resultaten te geven en deze analyse is dus niet opgenomen in dit rapport.

Naast het verschil in het aantal dagelijkse metingen hebben mogelijke fouten in de neerslagmetingen en foutieve locaties van de meetpunten ook invloed op de resultaten. Het kan zijn dat vrijwilligers van het Delft Meet Regen project vergeten zijn om de regenmeter te legen waardoor dagelijkse metingen bij elkaar opgeteld worden. Of dat de waarden van de metingen niet goed ingevoerd zijn in de app. Dit zou in figuur 5 het verschil in de gemiddelde dagelijkse neerslag kunnen verklaren op 10-08-2021. Deze fouten zouden ook de oorzaak kunnen zijn van de grote verschillen in standaard afwijking tussen de gebieden die te zien zijn in figuur 6. Op 23-08-2021 is de standaardafwijking van gebied NO 10,9 mm. Deze hoge standaard afwijking wordt beïnvloed door de neerslag meting van 50 mm die geïllustreerd is in figuur 9. De andere neerslag metingen van deze dag zitten tussen de 0 en 25 mm. Dit kan betekenen dat de meting van 50 mm een meetfout is. Deze meetfouten hebben ook invloed op het uitrekenen van de correlaties. Door meetfouten zijn de correlaties lager dan dat ze in werkelijkheid zijn. Een meetpunt heeft in de data verschillende locaties (zie tabel 1 en tabel 2) terwijl de regenmeter in werkelijkheid maar één locatie heeft. Als de afstand tussen twee meetpunten berekend wordt, wordt de gemiddelde afstand tussen deze twee meetpunten berekend. Dit geeft dus een bepaalde foutmarge met de werkelijke afstand tussen de meetpunten. Dit kan voorkomen worden door consequent dezelfde locatie van de regenmeter per meting door te geven. Verder hebben externe factoren ook invloed op de neerslagmetingen. Verdamping en wind beïnvloeden de neerslagmetingen en daarom ook de resultaten.

## 6. Conclusie

De neerslaggegevens van het Delft Meet Regen project zijn vergeleken met de neerslaggegevens van KNMI station 449. De gemiddelde dagelijkse neerslagmetingen en de cumulatieve waarden hiervan komen zeer goed overeen met elkaar. Het grootste verschil wordt gemeten op 10-08-2021, op de andere dagen zijn de verschillen minimaal.

Uit de gebiedenanalyse kan er geconcludeerd worden dat er verschil zit tussen de dagelijkse gemiddelde neerslag per gedefinieerd gebied, dus dat er sprake is van ruimtelijke spreiding van regen. Dit gemiddelde wordt echter beïnvloed door het aantal metingen per gebied. Dit aantal metingen verschilt per dag. Hoe meer metingen er zijn, des te nauwkeuriger de gemiddelden zijn. Uit de gebiedenanalyse bleek ook dat in gebied ZW cumulatief gezien gemiddeld de meeste neerslag is gemeten. Er is echter geen neerslagpatroon te ontdekken in deze analyse. De dagelijkse regenspreiding verschilt per dag. De standaard afwijkingen waren voor de gebieden NW en NO het grootst. Dit is te verklaren door afwijkende metingen in deze gebieden. Dit is ook te zien in de ruimtelijke plots in bijlage A.

Tussen elke mogelijke combinatie van twee meetpunten (maximaal 4005) is de afstand en Pearsons correlatie coëfficiënt uitgerekend. Een hoge correlatie tussen twee meetpunten betekent dat deze meetpunten met een grote kans door dezelfde bui getroffen worden. Een lage correlatie betekent dus dat de meetpunten niet door dezelfde bui getroffen worden. Dit betekent dus dat de kans groter is dat de neerslagmetingen van elkaar verschillen. Dit maakt de kans op ruimtelijke spreiding van regen groter. De correlatie coëfficiënten zijn berekend met een drempelwaarde van 5 en een drempelwaarde van 10. De correlatie coëfficiënten die berekend zijn met drempelwaarde 10 blijken nauwkeuriger te zijn dan de correlatie coëfficiënten die berekend zijn met drempelwaarde 5. Dit blijkt uit de ondergrens (10<sup>e</sup> percentiel) en het aantal negatieve correlaties uitgerekend met drempelwaarde 10. De ondergrens van de gemiddelde correlaties ligt 0,08 hoger en er zijn 117 minder negatieve correlaties berekend (in totaal 10). Het verschil in het gemiddelde van de berekende correlaties met drempelwaarde 5 en 10 is minimaal. Deze gemiddelden zijn respectievelijk 0,77 en 0,78.

Uit de ruimtelijke correlatieanalyse kan geconcludeerd worden dat er een verband bestaat tussen de afstand en de ruimtelijke correlatie: hoe groter de afstand tussen de meetpunten is, des te lager de ruimtelijke correlatie wordt. Dit verband wordt bevestigd door de exponentiële functie (zie formule 6) die door de correlatie zwaartepunten van de afstandsintervallen is gefit. Een lage correlatie tussen meetpunten betekent dat de kans op neerslagverschillen groter wordt en dat dus de kans op neerslagspreiding in het desbetreffende gebied groter wordt.

Uit dit onderzoek kan geconcludeerd worden dat er sprake is van dagelijkse neerslagspreiding op basis van de Delft Meet Regen data, maar dat hierin geen patroon in te ontdekken is. Om hierin een eventueel patroon in te ontdekken zullen er meer metingen voor een langere periode uitgevoerd moeten worden. Dit zal de nauwkeurigheid van de resultaten bevorderen.

## 7. Aanbevelingen

De methodiek en de Python scripts die voor dit onderzoek gebruikt zijn kunnen voor elke regendataset gebruikt worden. Hiermee is het mogelijk om met elke regendataset de correlaties en bijbehorende afstanden uitgerekend worden. De resultaten zullen nauwkeuriger worden als er meer metingen voor een langere periode met meer verschillende (grotere) afstanden geanalyseerd worden. De op- en aanmerkingen over de metingen die in het Excel-bestand van de Delft Meet Regen data waren meegestuurd zijn in dit rapport niet onderzocht. Hiermee kunnen de meetfouten en afwijkingen in de neerslagmetingen verklaard worden.



## Literatuurlijst

Beauducel, F. (2019). *Latitude/longitude to and from UTM coordinates precise and vectorized conversion*. MathWorks. Geraadpleegd op 15-09-2021, van [LL2UTM and UTM2LL - File Exchange - MATLAB Central \(mathworks.com\)](#)

Dekking, F.M., Kraaikamp, C., Lopuhaä, H.P., & Meester, L.E. (2005). *A modern introduction to probability and statistics*. Springer.

KNMI. (2021). *Automatische weerstations*. Geraadpleegd op 11-09-2021, van <https://www.knmi.nl/kennis-en-datacentrum/uitleg/automatische-weerstations>

KNMI. (2021). *Hoe werkt de neerslagradar van het KNMI?* Geraadpleegd op 10-09-2021, van <https://www.knmi.nl/kennis-en-datacentrum/achtergrond/neerslagradar-knmi-uitgebreide-toelichting>

KNMI. (2021). *Vrijwillige neerslagwaarnemers*. Geraadpleegd op 10-09-2021, van <https://www.knmi.nl/kennis-en-datacentrum/uitleg/vrijwillige-neerslagmeters>

KNMI. (2021). *Dagwaarden neerslagstations*. Geraadpleegd op 12-09-2021, van <https://www.knmi.nl/nederland-nu/klimatologie/monv/reeksen>

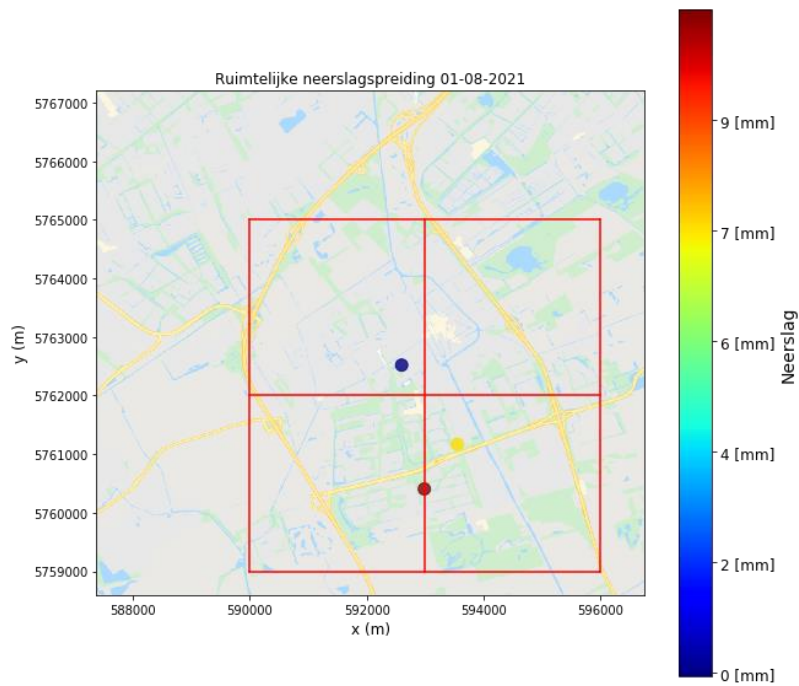
KNMI, & WOW. (2021). *Over WOW-NL*. Geraadpleegd op 08-09-2021, van [WOW-NL: Jouw weer op de kaart \(knmi.nl\)](#)

Van Leth, T.C., Leijnse, H., Overeem, A., & Uijlenhoet, R. (2021). *Rainfall Spatiotemporal Correlation and Intermittency Structure from Micro- $\gamma$  to Meso- $\beta$  Scale in the Netherlands* (Journal of Hydrometeorology, 22, 2227–2240).

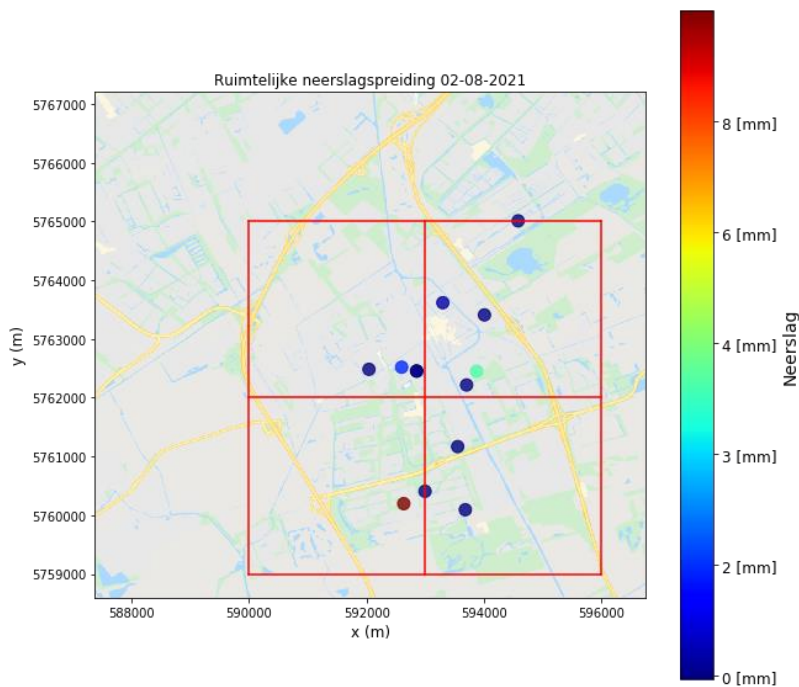
Waterlab. (2021). *Handleiding Delft meet regen*. Geraadpleegd op 10-09-2021, van <https://www.tudelft.nl/scd/waterlab/doe-mee-aan-onderzoek/project-7-delft-meet-regen>

# Bijlages

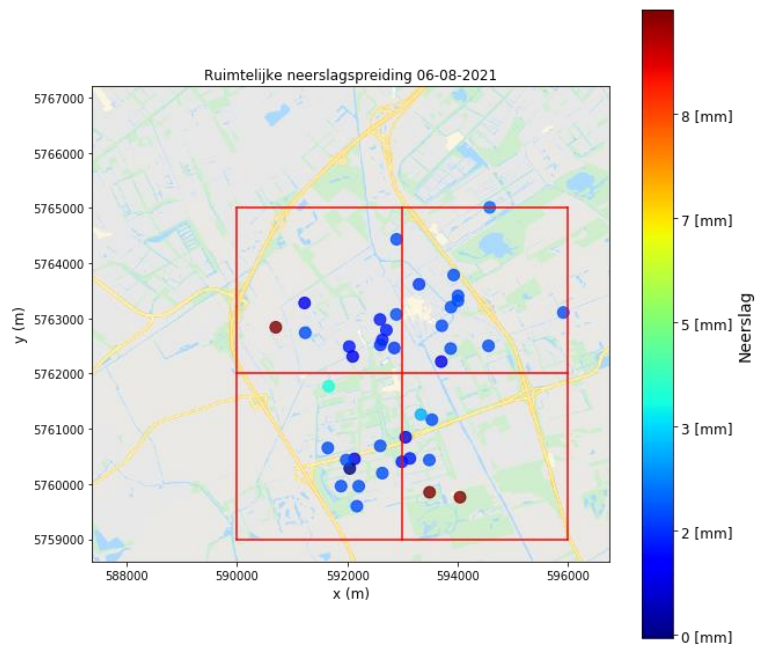
## Bijlage A. Ruimtelijke plots neerslagmetingen per dag



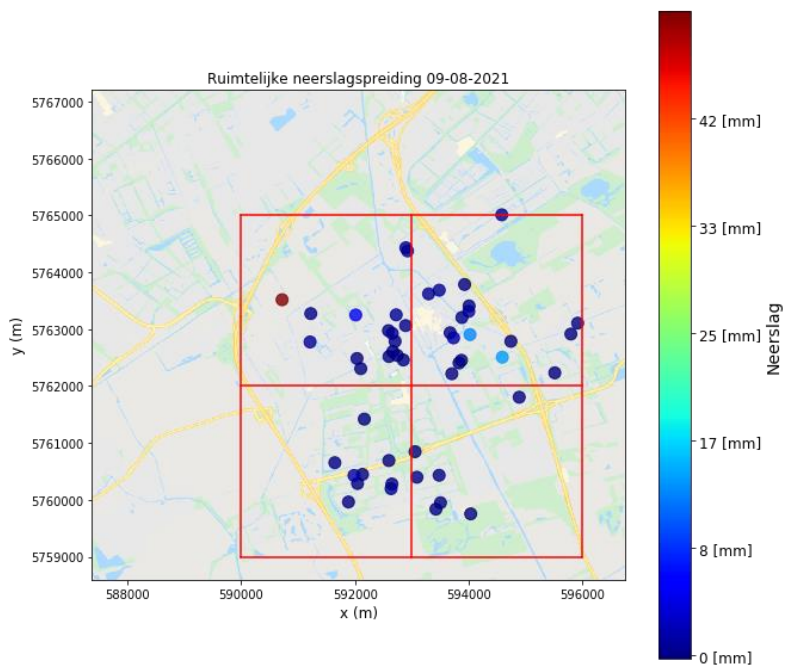
Figuur 19 – Neerslag metingen alle meetpunten op 01-08-2021



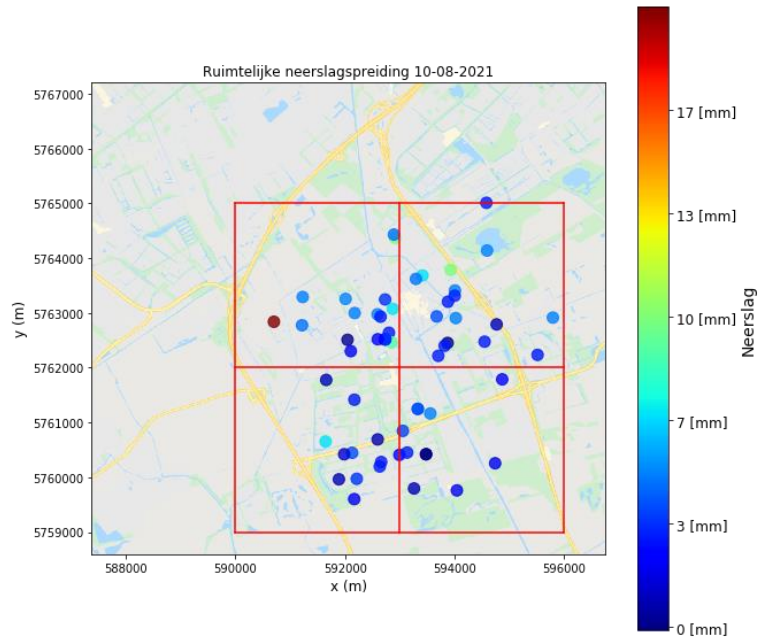
Figuur 20 – Neerslag metingen alle meetpunten op 02-08-2021



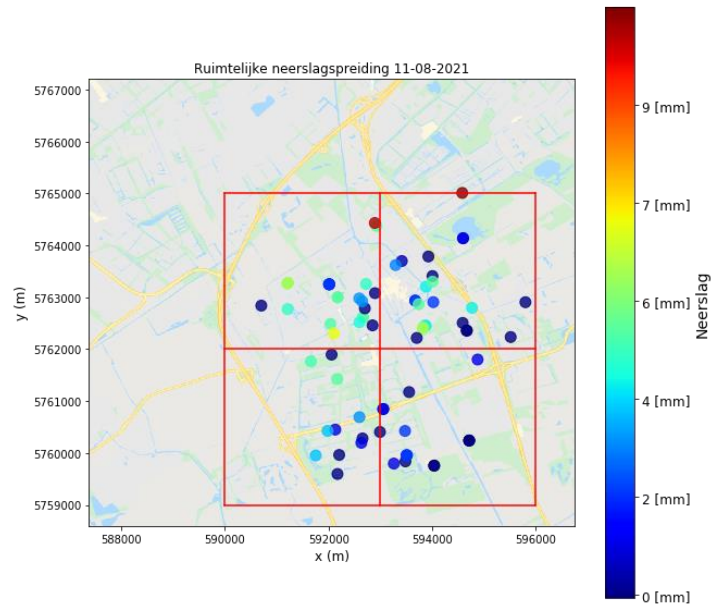
*Figuur 21 – Neerslag metingen alle meetpunten op 06-08-2021*



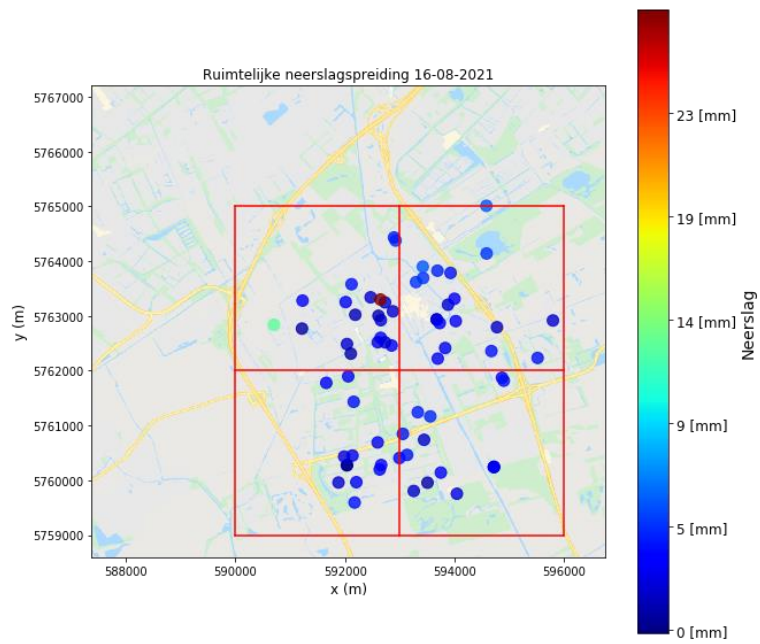
*Figuur 22 – Neerslag metingen alle meetpunten op 09-08-2021*



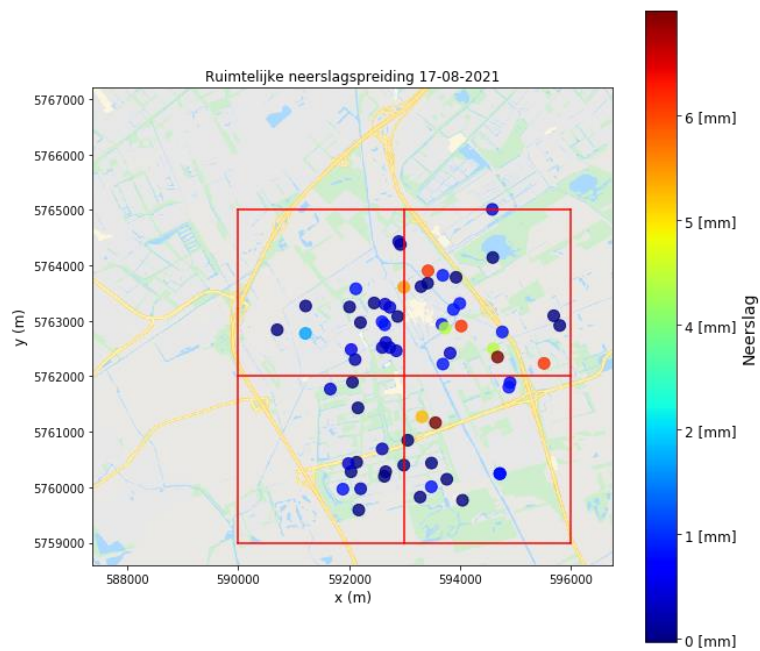
*Figuur 23 – Neerslag metingen alle meetpunten op 10-08-2021*



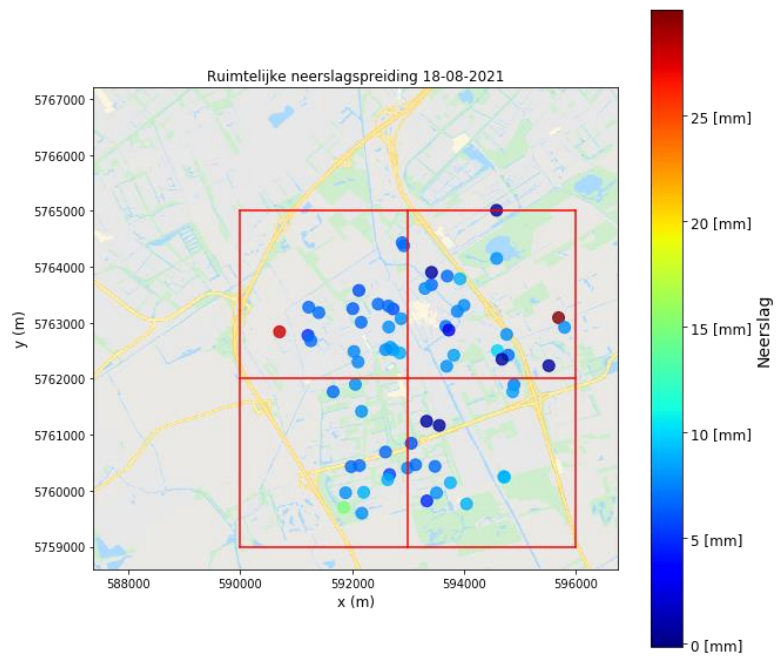
*Figuur 24 – Neerslag metingen alle meetpunten op 11-08-2021*



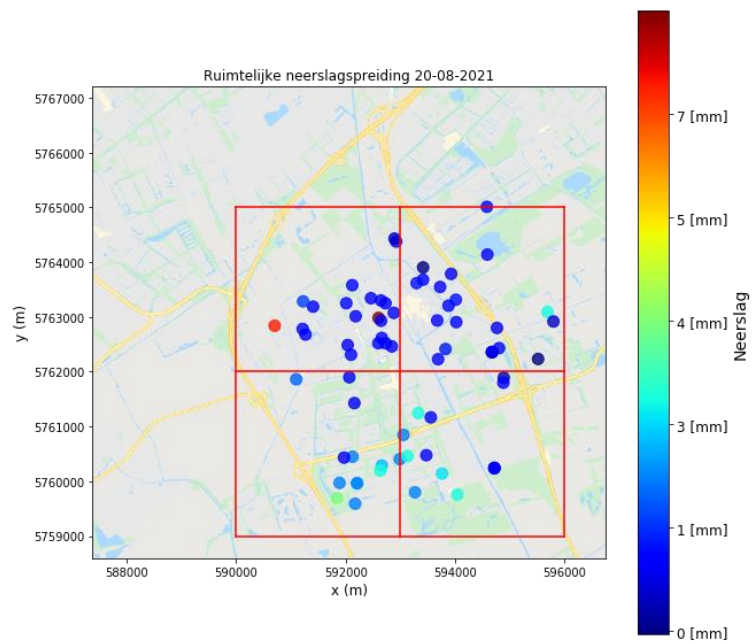
Figuur 25 – Neerslag metingen alle meetpunten op 16-08-2021



Figuur 26 – Neerslag metingen alle meetpunten op 17-08-2021

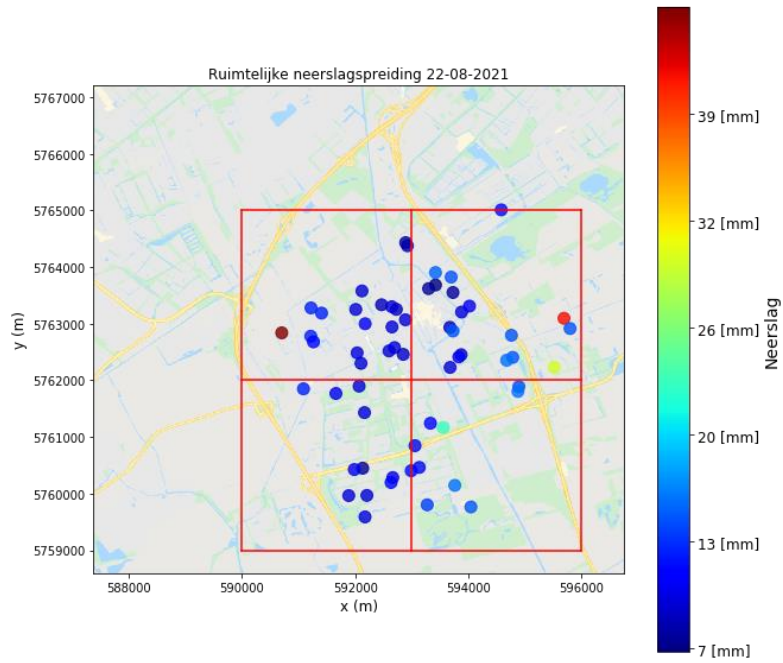


Figuur 27 – Neerslag metingen alle meetpunten op 18-08-2021

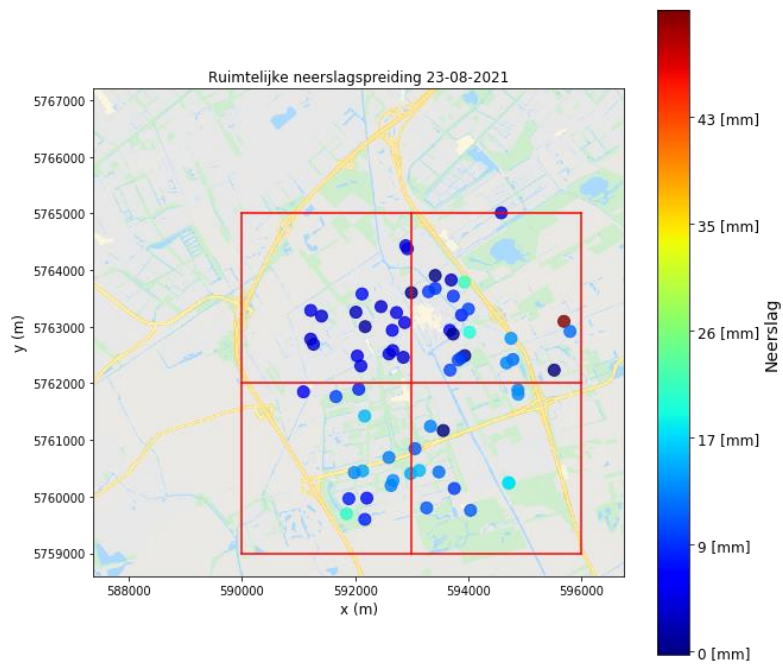


Figuur 28 – Neerslag metingen alle meetpunten op 20-08-2021





*Figuur 29 – Neerslag metingen alle meetpunten op 22-08-2021*



*Figuur 30 – Neerslag metingen alle meetpunten op 23-08-2021*

## Bijlage B. Matlab code converteren coördinaten

(Beauducel, 2019)

```
function [x,y,f]=ll2utm(varargin)
%LL2UTM Lat/Lon to UTM coordinates precise conversion.
% [X,Y]=LL2UTM2(LAT,LON) or LL2UTM([LAT,LON]) converts coordinates
% LAT,LON (in degrees) to UTM X and Y (in meters). Default datum is WGS84.
%
% LAT and LON can be scalars, vectors or matrix. Outputs X and Y will
% have the same size as inputs.
%
% LL2UTM(...,DATUM) uses specific DATUM for conversion. DATUM can be one
% of the following char strings:
% 'wgs84': World Geodetic System 1984 (default)
% 'nad27': North American Datum 1927
% 'clk66': Clarke 1866
% 'nad83': North American Datum 1983
% 'grs80': Geodetic Reference System 1980
% 'int24': International 1924 / Hayford 1909
% or DATUM can be a 2-element vector [A,F] where A is semimajor axis (in
% meters) and F is flattening of the user-defined ellipsoid.
%
% LL2UTM(...,ZONE) forces the UTM ZONE (scalar integer or same size as
% LAT and LON) instead of automatic set.
%
% [X,Y,ZONE]=LL2UTM(...) returns also the computed UTM ZONE (negative
% value for southern hemisphere points).
%
%
% XY=LL2UTM(...) or without any output argument returns a 2-column
% matrix [X,Y].
%
% Note:
% - LL2UTM does not perform cross-datum conversion.
% - precision is near a millimeter.
%
% Reference:
% I.G.N., Projection cartographique Mercator Transverse: Algorithmes,
% Notes Techniques NT/G 76, janvier 1995.
%
% Acknowledgments: Mathieu, Frederic Christen.
%
%
% Author: Francois Beauducel, <beauducel@ipgp.fr>
% Created: 2003-12-02
% Updated: 2019-05-29
% Copyright (c) 2001-2019, François Beauducel, covered by BSD License.
% All rights reserved.
%
% Redistribution and use in source and binary forms, with or without
% modification, are permitted provided that the following conditions are
% met:
%
% * Redistributions of source code must retain the above copyright
% notice, this list of conditions and the following disclaimer.
% * Redistributions in binary form must reproduce the above copyright
% notice, this list of conditions and the following disclaimer in
% the documentation and/or other materials provided with the distribution
%
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
```



```

% AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
% IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
% ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
% LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
% CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
% SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
% INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
% CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
% ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
% POSSIBILITY OF SUCH DAMAGE.
% Available datums
datums = [ ...
    { 'wgs84', 6378137.0, 298.257223563 };
    { 'nad83', 6378137.0, 298.257222101 };
    { 'grs80', 6378137.0, 298.257222101 };
    { 'nad27', 6378206.4, 294.978698214 };
    { 'int24', 6378388.0, 297.000000000 };
    { 'clk66', 6378206.4, 294.978698214 };
];
% constants
D0 = 180/pi; % conversion rad to deg
K0 = 0.9996; % UTM scale factor
X0 = 500000; % UTM false East (m)
% defaults
datum = 'wgs84';
zone = [];
if nargin < 1
    error('Not enough input arguments.')
end
if nargin > 1 && isnumeric(varargin{1}) && isnumeric(varargin{2}) ...
    && (all(size(varargin{1})==size(varargin{2})) ...
    || isscalar(varargin{1}) || isscalar(varargin{2}))
    lat = varargin{1};
    lon = varargin{2};
    v = 2;
elseif isnumeric(varargin{1}) && size(varargin{1},2) == 2
    lat = varargin{1}(:,1);
    lon = varargin{1}(:,2);
    v = 1;
else
    error('Single input argument must be a 2-column matrix [LAT, LON].')
end
if all([numel(lat),numel(lon)] > 1) && any(size(lat) ~= size(lon))
    error('LAT and LON must be the same size or scalars.')
end
if any(abs(lat)>90)
    error('LAT absolute values must be lower than 90.')
end
% checks for DATUM and/or ZONE syntax
% NOTE: the following strategy works in any case except if ZONE argument
% has a size of 1x2 (in that case it will be interpreted as a DATUM). To
% force the ZONE syntax with 2 elements, just use ZONE(:) to make a colum
% vector of 2x1.
for n = (v+1):nargin
    % LL2UTM(...,DATUM)
    if ischar(varargin{n}) || (isnumeric(varargin{n}) ...
        && all(size(varargin{n})==[1,2]))
        datum = varargin{n};
    % LL2UTM(...,ZONE)
    elseif isnumeric(varargin{n}) && (isscalar(varargin{n}) ...
        || (isscalar(lat) || all(size(varargin{n})==size(lat))) ...
        && (isscalar(lon) || all(size(varargin{n})==size(lon))))

```

```

        zone = round(varargin{n});
    else
        error('Unknown argument #%. See documentation.',n)
    end
end
if ischar(datum)
    % LL2UTM(...,DATUM) with DATUM as char
    if ~any(strcmpi(datum,datums(:,1)))
        error('Unkown DATUM name "%s"', datum);
    end
    k = find(strcmpi(datum,datums(:,1)));
    A1 = datums{k,2};
    F1 = datums{k,3};
else
    % LL2UTM(...,DATUM) with DATUM as [A,F] user-defined
    A1 = datum(1);
    F1 = datum(2);
end
p1 = lat/D0;           % Phi = Latitude (rad)
l1 = lon/D0;          % Lambda = Longitude (rad)
% UTM zone automatic setting
if isempty(zone)
    F0 = round((l1*D0 + 183)/6);
else
    F0 = abs(zone);
end
B1 = A1*(1 - 1/F1);
E1 = sqrt((A1*A1 - B1*B1)/(A1*A1));
P0 = 0/D0;
L0 = (6*F0 - 183)/D0; % UTM origin longitude (rad)
Y0 = 1e7*(p1 < 0);   % UTM false northern (m)
N = K0*A1;
C = coef(E1,0);
B = C(1)*P0 + C(2)*sin(2*P0) + C(3)*sin(4*P0) + C(4)*sin(6*P0) + C(5)*sin(8*P0);
YS = Y0 - N*B;
C = coef(E1,2);
L = log(tan(pi/4 + p1/2).*((1 - E1*sin(p1))./(1 + E1*sin(p1))).^(E1/2));
z = complex(atan(sinh(L)./cos(l1 - L0)),log(tan(pi/4 + asin(sin(l1 -
L0)./cosh(L))/2)));
Z = N.*C(1).*z + N.*(C(2)*sin(2*z) + C(3)*sin(4*z) + C(4)*sin(6*z) + C(5)*sin(8*z));
xs = imag(Z) + X0;
ys = real(Z) + YS;
% outputs zone if needed: scalar value if unique, or vector/matrix of the
% same size as x/y in case of crossed zones
if nargout > 2
    f = F0.*sign(lat);
    fu = unique(f);
    if isscalar(fu)
        f = fu;
    end
end
if nargout < 2
    x = [xs(:),ys(:)];
else
    x = xs;
    y = ys;
end
end

```

Figuur 31 – Matlab code voor het omzetten van latitude en longitude coördinaten naar x en y coördinaten (Beauducel, 2019)

## Bijlage C. Python scripts

```
import os
import tarfile
import urllib.request
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline
from scipy.stats import pearsonr
import math
import functools
from scipy.optimize import curve_fit
from numpy.random import randn
from numpy.random import seed
from tempfile import TemporaryFile
from matplotlib import pyplot
from numpy.random import normal
from numpy import hstack
from numpy import asarray
from numpy import exp
from sklearn.neighbors import KernelDensity
from IPython.display import display
from scipy.stats import linregress

regendata = pd.read_csv('myDatatrial.csv',sep=',',skipinitialspace=True)
regendata['Neerslag'] = pd.to_numeric(regendata['Neerslag'], errors='coerce')

KNMI = np.loadtxt('KNMI 24u.txt',skiprows = 10, usecols=1)
KNMI = KNMI*0.1

meetpunten = list(dict.fromkeys(regendata.Meetpunt))
datapunten = np.array(['01 08 2021','02 08 2021','03 08 2021','04 08 2021','05 08 2021','06 08 2021','07 08 2021','08 08 2021','0
```

Figuur 32 – Inladen data Delft Meet Regen en KNMI

```

def tweepunten(datum,punt1,punt2):

    meetpunt1=regendata.loc[regendata.Meetpunt == punt1]
    meetpunt2=regendata.loc[regendata.Meetpunt == punt2]
    combined_meetpunt= meetpunt1.append(meetpunt2)

    combined_meetpunt= meetpunt1.append(meetpunt2)
    vergelijking_1dag=combined_meetpunt.loc[combined_meetpunt.Datum == datum]

    meetpunt1 = meetpunt1.drop_duplicates(subset='Datum', keep="last")
    meetpunt2 = meetpunt2.drop_duplicates(subset='Datum', keep="last")
    combined_meetpunt= meetpunt1.append(meetpunt2)
    vergelijking_1dag=combined_meetpunt.loc[combined_meetpunt.Datum == datum]
    check_point = 0

    r, c = vergelijking_1dag.shape
    if r < 2:
        check_point = 1

    if check_point == 0:
        regenpunt1 = vergelijking_1dag.Neerslag[regendata.Meetpunt == punt1].values[0]
        regenpunt2 = vergelijking_1dag.Neerslag[regendata.Meetpunt == punt2].values[0]

        dx = vergelijking_1dag.North[regendata.Meetpunt == punt1].values[0] -vergelijking_1dag.North[regendata.Meetpunt == punt2]
        dy = vergelijking_1dag.East[regendata.Meetpunt == punt1].values[0]-vergelijking_1dag.East[regendata.Meetpunt == punt2].va
        d = np.sqrt(dx**2 + dy**2)

    else:
        d = np.nan
        regenpunt1 = np.nan
        regenpunt2 = np.nan

    return regenpunt1,regenpunt2,d

def newregenpunt (punt1,punt2):
    regenpunt1 = np.zeros(len(datapunten))
    regenpunt2 = np.zeros(len(datapunten))
    d = np.zeros(len(datapunten))
    for i in range(len(datapunten)):
        regenpunt1[i] = tweepunten(datapunten[i],punt1,punt2)[0]
        regenpunt2[i] = tweepunten(datapunten[i],punt1,punt2)[1]
        d[i] = tweepunten(datapunten[i],punt1,punt2)[2]

    if (regenpunt1[i] == 0) & (regenpunt2[i] == 0):
        regenpunt1[i] = np.nan
        regenpunt2[i] = np.nan
        d[i] = np.nan

    newregenpunt1 = np.array([x for x in regenpunt1 if math.isnan(x) == False])
    newregenpunt2 = np.array([x for x in regenpunt2 if math.isnan(x) == False])
    newd = [x for x in d if math.isnan(x) == False]

    return newregenpunt1, newregenpunt2

```

*Figuur 33 – De tweepuntenfunctie en de newregenpunt functie*

```

def pearson (punt1,punt2):
    regenpunt1 = np.zeros(len(datapunten))
    regenpunt2 = np.zeros(len(datapunten))
    d = np.zeros(len(datapunten))
    for i in range(len(datapunten)):
        regenpunt1[i] = tweepunten(datapunten[i],punt1,punt2)[0]
        regenpunt2[i] = tweepunten(datapunten[i],punt1,punt2)[1]
        d[i] = tweepunten(datapunten[i],punt1,punt2)[2]

    if (regenpunt1[i] == 0) & (regenpunt2[i] == 0):
        regenpunt1[i] = np.nan
        regenpunt2[i] = np.nan
        d[i] = np.nan

    newregenpunt1 = [x for x in regenpunt1 if math.isnan(x) == False]
    newregenpunt2 = [x for x in regenpunt2 if math.isnan(x) == False]
    newd = [x for x in d if math.isnan(x) == False]

    if len(newregenpunt1) > 5: # drempelwaarde
        pearson = pearsonr(newregenpunt1, newregenpunt2)[0]
    else:
        pearson = np.nan
        newd = np.nan

    return np.mean(newd), pearson

def area(gebied,datum):

    rain = np.array(gebied.Neerslag[gebied.Datum == datum])
    measures = len(rain)
    mean = np.mean(rain)
    s = np.std(rain)
    return measures,mean,s

def pdf(sample):
    model = KernelDensity(bandwidth=0.01, kernel='gaussian')
    sample = sample.reshape((len(sample), 1))
    model.fit(sample)
    values = asarray([value for value in np.linspace(-1,1,50)])
    values = values.reshape((len(values), 1))
    probabilities = model.score_samples(values)
    probabilities = exp(probabilities)
    return values,probabilities

def func(x, b, c):
    return np.exp(-(x/b)**c)

```

Figuur 34 – Pearson, area, pdf en func functies

```

delft_img=mpimg.imread('delft.png')

regendatadaily = regendata[regendata.Datum == '23 08 2021']

ax = regendatadaily.plot(kind="scatter", x="East", y="North", figsize=(10,10),
                        s=100,
                        c="Neerslag", cmap=plt.get_cmap("jet"),
                        colorbar=False, alpha=0.8)

plt.imshow(delft_img, extent=[ 587381.24,596741.68,5758599.04, 5767200.69], alpha=1,
          cmap=plt.get_cmap("jet"))
plt.ylabel("y (m)", fontsize=12)
plt.xlabel("x (m)", fontsize=12)
plt.xlim(587381.24,596741.68)
plt.ylim(5758599.04,5767200.69)
neerslag = regendatadaily.Neerslag
tick_values = np.linspace(np.min(neerslag), np.max(neerslag), 7)
cbar = plt.colorbar()
cbar.ax.set_yticklabels(["%d [mm]"%(round(v)) for v in tick_values], fontsize=12)
cbar.set_label('Neerslag', fontsize=14)

plt.plot([590000,590000],[5759000,5765000],'r')
plt.plot([596000,596000],[5759000,5765000],'r')
plt.plot([590000,596000],[5759000,5759000],'r')
plt.plot([590000,596000],[5765000,5765000],'r')
plt.plot([590000,596000],[5762000,5762000],'r')
plt.plot([593000,593000],[5759000,5765000],'r')

plt.title('Ruimtelijke neerslagspreiding 23-08-2021',fontsize=12)
plt.show();

```

Figuur 35 – Python script gebruikt voor figuur 9 en Bijlage A

```

punt1 = meetpunten[6]
punt2 = meetpunten[45]

regenpunt1 = np.zeros(len(datapunten))
regenpunt2 = np.zeros(len(datapunten))
d = np.zeros(len(datapunten))

for i in range(len(datapunten)):
    regenpunt1[i] = tweepunten(datapunten[i],punt1,punt2)[0]
    regenpunt2[i] = tweepunten(datapunten[i],punt1,punt2)[1]
    d[i] = tweepunten(datapunten[i],punt1,punt2)[2]
    if (regenpunt1[i] == 0) & (regenpunt2[i] == 0):
        regenpunt1[i] = np.nan
        regenpunt2[i] = np.nan
        d[i] = np.nan

newregenpunt1 = [x for x in regenpunt1 if math.isnan(x) == False]
newregenpunt2 = [x for x in regenpunt2 if math.isnan(x) == False]
newd = [x for x in d if math.isnan(x) == False]

fig= plt.figure(figsize=(6,6))
x = np.linspace(0,20)
a = linregress(newregenpunt1, newregenpunt2)
plt.plot(newregenpunt1, newregenpunt2,'o',label = 'Metingen')
plt.plot(x,a[1]+a[0]*x,'r--', label = 'Lineaire regressie lijn')
plt.ylabel('Meting 2 in (mm)')
plt.xlabel('Meting 1 in (mm)')
plt.xlim([0, max(max(newregenpunt1),max(newregenpunt2))+1])
plt.ylim([0, max(max(newregenpunt1),max(newregenpunt2))+1])
plt.title('Metingen tegen elkaar uitgezet met afstand 3634 m en correlatie -0,06')
plt.legend()

fig= plt.figure(figsize=(6,6))
plt.imshow(delft_img, extent=[ 587381.24,596741.68,5758599.04, 5767200.69], alpha=1,
        cmap=plt.get_cmap("jet"))
plt.ylabel("y(m)", fontsize=12)
plt.xlabel("x(m)", fontsize=12)
plt.title('Locatie meetpunten')
plt.plot(regendata.East[regendata.Meetpunt == punt1].values[0],regendata.North[regendata.Meetpunt == punt1].values[0],'bo')
plt.plot(regendata.East[regendata.Meetpunt == punt2].values[0],regendata.North[regendata.Meetpunt == punt2].values[0],'bo');

```

Figuur 36 – Python script gebruikt voor figuur 10, 11 en 12

```

fig= plt.figure(figsize=(10,10))

plt.plot([590000,590000],[5759000,5765000],'r')
plt.plot([596000,596000],[5759000,5765000],'r')
plt.plot([590000,596000],[5759000,5759000],'r')
plt.plot([590000,596000],[5765000,5765000],'r')
plt.plot([590000,596000],[5762000,5762000],'r')
plt.plot([593000,593000],[5759000,5765000],'r')

plt.imshow(delft_img, extent=[ 587381.24,596741.68,5758599.04, 5767200.69], alpha=1,
        cmap=plt.get_cmap("jet"))
plt.ylabel("y (m)", fontsize=12)
plt.xlabel("x(m)", fontsize=12)
plt.title('Gebiedsindeling',fontsize=12)

gebied1 = regendata[(regendata.East > 590000)&(regendata.East <593000)&(regendata.North > 5762000)&(regendata.North < 5765000)]
gebied2 = regendata[(regendata.East > 593000)&(regendata.East <596000)&(regendata.North > 5762000)&(regendata.North < 5765000)]
gebied3 = regendata[(regendata.East > 590000)&(regendata.East <593000)&(regendata.North > 5759000)&(regendata.North < 5762000)]
gebied4 = regendata[(regendata.East > 593000)&(regendata.East <596000)&(regendata.North > 5759000)&(regendata.North < 5762000)]
plt.text(591000,5763250,'NW', color = 'tab:blue', size =30, weight = 1000 )
plt.text(594000,5763250,'NO', color = 'tab:orange', size =30, weight = 1000 )
plt.text(591000,5760250,'ZN', color = 'tab:green', size =30, weight = 1000 )
plt.text(594000,5760250,'ZO', color = 'tab:red', size =30, weight = 1000 )
for i in range(len(meetpunten)):
    plt.plot(regendata.East[regendata.Meetpunt == meetpunten[i]].values[0],regendata.North[regendata.Meetpunt == meetpunten[i]].v
    plt.plot(regendata.East[regendata.Meetpunt == meetpunten[i]].values[0],regendata.North[regendata.Meetpunt == meetpunten[i]].v
plt.ylim( 5758599.04,5767000) ;

```

Figuur 37 - Python script gebruikt voor figuur 4

```

meandaily = np.zeros(len(datapunten))
measures = np.zeros(len(datapunten))
std = np.zeros(len(datapunten))

for i in range(len(datapunten)):
    measures[i] = area(regendata,datapunten[i])[0]
    meandaily[i] = area(regendata,datapunten[i])[1]
    std[i] = area(regendata,datapunten[i])[2]

x = np.arange(len(datapunten))

plt.figure(figsize=(10,10),dpi = 80)

plt.subplot(4, 1, 1)
plt.plot(x,meandaily, label = 'Delft Meet Regen')
plt.plot(x,KNMI,'--',label = 'KNMI station 449')
plt.title('Gemiddelde dagelijkse neerslag Delft')
plt.ylabel('Neerslag in mm')
plt.xticks(np.arange(len(x)),())
plt.legend()

plt.subplot(4, 1, 2)
plt.bar(x,measures)
plt.title('Aantal dagelijkse metingen')
plt.ylabel('Aantal metingen')
plt.xticks(np.arange(len(x)),())

plt.subplot(4, 1, 3)
plt.plot(x,std)
plt.title('Dagelijkse standaard afwijking')
plt.ylabel('Standaard deviatie in mm')
plt.xticks(np.arange(len(x)),())

plt.subplot(4, 1, 4)
plt.plot(x,np.cumsum(meandaily))
plt.plot(x,np.cumsum(KNMI),'--')
plt.title('Gemiddelde dagelijkse neerslag cumulatief')
plt.ylabel('Cumulatieve neerslag in mm')
plt.xlabel('Datum')
plt.xticks(np.arange(len(x)),datapunten,rotation = 90)

plt.tight_layout();

```

Figuur 38 - Python script gebruikt voor figuur 5

```

gebied1mean = np.zeros(len(datapunten))
gebied2mean = np.zeros(len(datapunten))
gebied3mean = np.zeros(len(datapunten))
gebied4mean = np.zeros(len(datapunten))
gebied1std = np.zeros(len(datapunten))
gebied2std = np.zeros(len(datapunten))
gebied3std = np.zeros(len(datapunten))
gebied4std = np.zeros(len(datapunten))
gebied1measures = np.zeros(len(datapunten))
gebied2measures = np.zeros(len(datapunten))
gebied3measures = np.zeros(len(datapunten))
gebied4measures = np.zeros(len(datapunten))

for Si in range(len(datapunten)):
    gebied1measures[i],gebied1mean[i],gebied1std[i] = area(gebied1,datapunten[i])[0],area(gebied1,datapunten[i])[1],area(gebied1,
    gebied2measures[i],gebied2mean[i],gebied2std[i] = area(gebied2,datapunten[i])[0],area(gebied2,datapunten[i])[1],area(gebied2,
    gebied3measures[i],gebied3mean[i],gebied3std[i] = area(gebied3,datapunten[i])[0],area(gebied3,datapunten[i])[1],area(gebied3,
    gebied4measures[i],gebied4mean[i],gebied4std[i] = area(gebied4,datapunten[i])[0],area(gebied4,datapunten[i])[1],area(gebied4,

x = np.arange(len(datapunten))
plt.figure(figsize=(8,8),dpi = 80)

plt.subplot(4, 1, 1)
plt.plot(x,gebied1mean, label = 'NW')
plt.plot(x,gebied2mean, label = 'NO')
plt.plot(x,gebied3mean, label = 'ZW')
plt.plot(x,gebied4mean, label = 'ZO')
plt.title('Gemiddelde dagelijkse neerslag per gebied')
plt.ylabel('Neerslag in mm')
plt.xticks(np.arange(len(x)),())
plt.legend(loc = 'best')

plt.subplot(4, 1, 2)
plt.plot(x,gebied1measures)
plt.plot(x,gebied2measures)
plt.plot(x,gebied3measures)
plt.plot(x,gebied4measures)
plt.title('Dagelijks aantal metingen per gebied')
plt.ylabel('Aantal metingen')
plt.xticks(np.arange(len(x)),())

plt.subplot(4, 1, 3)
plt.plot(x,gebied1std)
plt.plot(x,gebied2std)
plt.plot(x,gebied3std)
plt.plot(x,gebied4std)
plt.title('Dagelijkse standaard deviatie per gebied')
plt.ylabel('Standaard deviatie in mm')
plt.xticks(np.arange(len(x)),())

plt.subplot(4, 1, 4)
plt.plot(x,np.cumsum(gebied1mean))
plt.plot(x,np.cumsum(gebied2mean))
plt.plot(x,np.cumsum(gebied3mean))
plt.plot(x,np.cumsum(gebied4mean))
plt.title('Gemiddelde dagelijke neerslag cumulatief')
plt.ylabel('Cumulatieve neerslag in mm')
plt.xlabel('Datum')
plt.xticks(np.arange(len(x)),datapunten,rotation = 90)

plt.tight_layout();

```

Figuur 39 - Python script gebruikt voor figuur 6



```

buidatum = ('16 08 2021','17 08 2021','18 08 2021','20 08 2021','22 08 2021','23 08 2021')
gebieden = (gebied1,gebied2,gebied3,gebied4)
barsmeasures = np.zeros((len(gebieden),(len(buidatum))))
barsmean = np.zeros((len(gebieden),(len(buidatum))))
barsstd = np.zeros((len(gebieden),(len(buidatum))))

for i in range(len(gebieden)):
    for j in range(len(buidatum)):
        barsmeasures[i,j] = area(gebieden[i],buidatum[j])[0]
        barsmean[i,j] = area(gebieden[i],buidatum[j])[1]
        barsstd[i,j] = area(gebieden[i],buidatum[j])[2]

plt.figure(figsize=(10,10),dpi = 80)
plt.subplot(3, 1, 1)
barWidth = 0.1

bars1 = barsmean[0]
bars2 = barsmean[1]
bars3 = barsmean[2]
bars4 = barsmean[3]

r1 = np.arange(len(bars1))
r2 = [x + barWidth for x in r1]
r3 = [x + barWidth for x in r2]
r4 = [x + barWidth for x in r3]

plt.bar(r1, bars1, width=barWidth, label='NN')
plt.bar(r2, bars2, width=barWidth, label='NO')
plt.bar(r3, bars3, width=barWidth, label='ZW')
plt.bar(r4, bars4, width=barWidth, label='ZO')
plt.xticks([0.05 + r + barWidth for r in range(len(bars1))], (), rotation = 90)
plt.legend(loc = 'best')
plt.ylabel('Gemiddelde neerslag in mm')
plt.title('Gemiddelde neerslag per gebied per bui')
plt.show()

plt.figure(figsize=(10,10),dpi = 80)
plt.subplot(3, 1, 2)
barWidth = 0.1
bars1 = barsmeasures[0]
bars2 = barsmeasures[1]
bars3 = barsmeasures[2]
bars4 = barsmeasures[3]

r1 = np.arange(len(bars1))
r2 = [x + barWidth for x in r1]
r3 = [x + barWidth for x in r2]
r4 = [x + barWidth for x in r3]

plt.bar(r1, bars1, width=barWidth)
plt.bar(r2, bars2, width=barWidth)
plt.bar(r3, bars3, width=barWidth)
plt.bar(r4, bars4, width=barWidth)
plt.xticks([0.05 + r + barWidth for r in range(len(bars1))], (), rotation = 90)
plt.ylabel('Aantal metingen per gebied')
plt.title('Aantal metingen per gebied per bui')
plt.show()

plt.figure(figsize=(10,10),dpi = 80)
plt.subplot(3, 1, 3)
barWidth = 0.1
bars1 = barsstd[0]
bars2 = barsstd[1]
bars3 = barsstd[2]
bars4 = barsstd[3]

r1 = np.arange(len(bars1))
r2 = [x + barWidth for x in r1]
r3 = [x + barWidth for x in r2]
r4 = [x + barWidth for x in r3]

plt.bar(r1, bars1, width=barWidth)
plt.bar(r2, bars2, width=barWidth)
plt.bar(r3, bars3, width=barWidth)
plt.bar(r4, bars4, width=barWidth)

plt.xlabel('Datum')
plt.xticks([0.05 + r + barWidth for r in range(len(bars1))], buidatum, rotation = 90)
plt.ylabel('Standaard afwijking in mm')
plt.title('Standaard afwijking per gebied per bui')
plt.show();

```

Figuur 40 - Python script gebruikt voor figuur 7 en 8

```

meetpunten = meetpunten[:]
d = ()
r = ()
for i in range(len(meetpunten)):
    for j in range(len(meetpunten)):
        if j > i:
            d = np.append(d,pearson(meetpunten[i],meetpunten[j])[0])
            r = np.append(r,pearson(meetpunten[i],meetpunten[j])[1])

d = [x for x in d if math.isnan(x) == False]
r = [x for x in r if math.isnan(x) == False]
np.savetxt('d1last.txt',d,fmt="%s")
np.savetxt('r1last.txt',r,fmt="%s")

d = np.loadtxt('dnew.txt')
r = np.loadtxt('rnew.txt')
d = d[r>0]
r = r[r>0]
range1 = ()
range2 = ()
range3 = ()
range4 = ()
drange1 = ()
drange2 = ()
drange3 = ()
drange4 = ()

for i in range(len(r)):
    if (d[i] > 0)&(d[i] < 500):
        drange1 = np.append(drange1,d[i])
        range1 = np.append(range1,r[i])
    if (d[i] > 500)&(d[i] < 1500):
        drange2 = np.append(drange2,d[i])
        range2 = np.append(range2,r[i])
    if (d[i] > 1500)&(d[i] < 3000):
        drange3 = np.append(drange3,d[i])
        range3 = np.append(range3,r[i])
    if (d[i] > 3000)&(d[i] < 7000):
        drange4 = np.append(drange4,d[i])
        range4 = np.append(range4,r[i])

```

Figuur 41 – Opslaan van de afstanden en correlaties in arrays en de gedefinieerde afstandsintervallen

```

plt.figure(figsize = (10,10))
plt.subplot(2, 2, 1)
plt.hist(range1, bins=50, density=True)
plt.plot(pdf(range1)[0],pdf(range1)[1])
plt.xlim(0,1.05)
plt.ylim(0,7)
plt.plot((np.mean(range1),np.mean(range1)),(0,7), label = 'Gemiddelde',color = 'r')
lower = np.percentile(range1,10)
upper = np.percentile(range1,90)
plt.axvline(lower,color='k',label = '10e percentiel')
plt.axvline(upper,color='k', label = '90e percentiel')
plt.ylabel('(-)')
plt.legend()
plt.title('Interval 1')

plt.subplot(2, 2, 2)
plt.hist(range2, bins=50, density=True)
plt.plot(pdf(range2)[0],pdf(range2)[1])
plt.xlim(0,1.05)
plt.ylim(0,7)
plt.plot((np.mean(range2),np.mean(range2)),(0,7), label = 'Gemiddelde',color = 'r')
lower = np.percentile(range2,10)
upper = np.percentile(range2,90)
plt.axvline(lower,color='k',label = '10e percentiel')
plt.axvline(upper,color='k', label = '90e percentiel')
plt.legend()
plt.title('Interval 2')

plt.subplot(2, 2, 3)
plt.hist(range3, bins=50, density=True)
plt.plot(pdf(range3)[0],pdf(range3)[1])
plt.xlim(0,1.05)
plt.ylim(0,7)
plt.plot((np.mean(range3),np.mean(range3)),(0,7), label = 'Gemiddelde',color = 'r')
lower = np.percentile(range3,10)
upper = np.percentile(range3,90)
plt.axvline(lower,color='k',label = '10e percentiel')
plt.axvline(upper,color='k', label = '90e percentiel')
plt.xlabel('Pearsons coëfficiënt')
plt.ylabel('(-)')
plt.legend()
plt.title('Interval 3')

```

```

plt.subplot(2, 2, 4)
plt.hist(range4, bins=50, density=True)
plt.plot(pdf(range4)[0],pdf(range4)[1])
plt.xlim(0,1.05)
plt.ylim(0,7)
plt.plot((np.mean(range4),np.mean(range4)),(0,7), label = 'Gemiddelde',color = 'r')
lower = np.percentile(range4,10)
upper = np.percentile(range4,90)
plt.axvline(lower,color='k',label = '10e percentiel')
plt.axvline(upper,color='k', label = '90e percentiel')
plt.xlabel('Pearsons coëfficiënt')
plt.legend()
plt.title('Interval 4')

plt.figure(figsize = (10,10))
plt.plot(pdf(range1)[0],pdf(range1)[1],label = 'Interval 1')
plt.plot(pdf(range2)[0],pdf(range2)[1],label = 'Interval 2')
plt.plot(pdf(range3)[0],pdf(range3)[1],label = 'Interval 3')
plt.plot(pdf(range4)[0],pdf(range4)[1],label = 'Interval 4')
plt.xlim(0,1)
plt.xlabel('Pearsons coëfficiënt')
plt.ylabel('(-)')
plt.title('Kansdichtheidsfuncties Pearsons coëfficiënt van intervallen')
plt.legend();

```

Figuur 42 - Python script gebruikt voor figuur 14, 15, 17 en 18

```

plt.figure(figsize = (15,8))

plt.subplot(1,2,1)
g = (np.mean(drange1),np.mean(drange2),np.mean(drange3),np.mean(drange4))
h = (np.mean(range1),np.mean(range2),np.mean(range3),np.mean(range4))
plt.plot(d,r,'.',alpha = 0.4)
plt.ylim(0,1)
plt.xlim(0,7000)
plt.plot(g,h,'ro',label = 'Gemiddeldes intervallen')
plt.xlabel('Afstand (m)')
plt.ylabel('Pearsons coëfficiënt (-)')
plt.axvline(500,color='k',linewidth = 1,ls = '--')
plt.axvline(1500,color='k',linewidth = 1,ls = '--')
plt.axvline(3000,color='k',linewidth = 1,ls = '--')
plt.title('Pearsons coëfficiënt met drempelwaarde 10')

xx = np.linspace(0.0001,7000,200)

g = 0.36365692, 1.06869781, 2.25962747, 4.05692445
hup = np.percentile(range1,90),np.percentile(range2,90),np.percentile(range3,90),np.percentile(range4,90)
hdown = np.percentile(range1,10),np.percentile(range2,10),np.percentile(range3,10),np.percentile(range4,10)

popt, pcov = curve_fit(func, g, h)

b = popt[0]
c = popt[1]
plt.plot(xx, np.exp(-(xx/1000/b)**c), 'r-', label = 'Exponentiële curvefit gemiddelde')

poptup, pcovup = curve_fit(func, g, hup, maxfev=5000)

bup = poptup[0]
cup = poptup[1]
plt.plot(xx, np.exp(-(xx/1000)/bup)**cup), 'k-.', label = 'Exponentiële curvefit 90%')
plt.plot(np.true_divide(g, 0.001),hup,'ko', label = 'Bovengrens 90%')

poptdo, pcovdo = curve_fit(func, g, hdown, maxfev=5000)

bdo = poptdo[0]
cdo = poptdo[1]
plt.plot(xx, np.exp(-(xx/1000)/bdo)**cdo), 'k-.', label = 'Exponentiële curvefit 10%')
plt.plot(np.true_divide(g, 0.001),hdown,'ko', label = 'Benedengrens 10%')
plt.legend()

plt.subplot(1,2,2)
plt.hist(r,normed = True,bins = 100,orientation = 'horizontal')
plt.ylim(0,1)
plt.axhline(np.mean(r),color='r',label = 'Gemiddelde')
plt.axhline(np.percentile(r,10),color='k',label = '10e percentiel')
plt.axhline(np.percentile(r,90),color='k',label = '90e percentiel')
plt.legend()
plt.title('Histogram pearsons coëfficiënt drempelwaarde 10')
plt.xlabel('(-)');

```

Figuur 43 - Python script gebruikt voor figuur 13 en 16

```

fig= plt.figure(figsize=(8,8))
plt.imshow(delft_img, extent=[ 587381.24,596741.68,5758599.04, 5767200.69],
          cmap=plt.get_cmap("jet"))
plt.ylabel("y(m)", fontsize=12)
plt.xlabel("x (m)", fontsize=12)
plt.ylim( 5758599.04,5767000)
plt.title('Locatie meetpunten',fontsize =12 )
for i in range(len(meetpunten)):
    plt.plot(regendata.East[regendata.Meetpunt == meetpunten[i]].values[0],regendata.North[regendata.Meetpunt == meetpunten[i]].v
    plt.plot(regendata.East[regendata.Meetpunt == meetpunten[i]].values[0],regendata.North[regendata.Meetpunt == meetpunten[i]].v

```

*Figuur 44 – Python script van figuur 2*

```

fig= plt.figure(figsize=(6,6))
plt.imshow(delft_img, extent=[ 587381.24,596741.68,5758599.04, 5767200.69],
          cmap=plt.get_cmap("jet"))
plt.ylabel("y(m)", fontsize=12)
plt.xlabel("x (m)", fontsize=12)
plt.ylim( 5758599.04,5767000)
plt.title('Locatie meetpunt 2 en meetpunt 3',fontsize =12 )

plt.plot(regendata.East[regendata.Meetpunt == 'meetpunt2'].values[0],regendata.North[regendata.Meetpunt == 'meetpunt 2'].values[0]
plt.plot(regendata.East[regendata.Meetpunt == 'meetpunt3'].values[0],regendata.North[regendata.Meetpunt == 'meetpunt3'].values[0]
plt.legend()

```

*Figuur 45 – Python script van figuur 3*