DELFT MEET REGEN

Vergelijking van neerslagdata van Delft Meet Regen met radardata van het KNMI

Sonny Harnam

Bachelor of Science aan de Technische Universiteit Delft

Juni 2022

Studentennummer: 4591828

Begeleiders: Marie-Claire ten Veldhuis TU Delft

Remko Uijlenhoet TU Delft

Voorwoord

Dit document fungeert als het rapport van een Bachelor Eindproject, voor de afronding van de Bachelor of Science in Civiele Techniek aan de Technische Universiteit Delft. In dit document wordt een analyse besproken van neerslagdata verzamelt als deel van het Delft Meet Regen project. De nadruk van de analyse in dit document, bedraagt de vergelijking van de verzamelde regendata met beschikbare regendata van het KNMI.

De realisatie van deze analyse is te danken aan Marie-Claire ten Veldhuis, Remko Uijlenhoet, Sandra de Vries en Marit Bogert voor hun begeleiding en instructies. Ook gaat dank uit naar Aart Overeem voor zijn adviezen en instructies met betrekking tot de KNMI data.

Samenvatting

In dit document wordt een analyse beschreven die is uitgevoerd met behulp van vrijwilligersdata. Deze data is beschikbaar gesteld door het Delf Meet Regen project. De vrijwilligersdata betreft neerslagdata die is verzameld door een selectie van vrijwilligers in Delft, in samenwerking met de Technische Universiteit Delft. Deze analyse bouwt voort op andere onderzoeken die gebruik hebben gemaakt van dezelfde vrijwilligersdata.

In de analyse wordt deze vrijwilligersdata van het Delft Meet Regen project vergeleken met klimatologische radardata van het KNMI. Deze analyse betreft hoofdzakelijk een vergelijking van de ruimtelijke neerslagpatronen van deze verschillende datasets. De analyse is uitgevoerd in Python.

Voor de aanvang van de analyse zijn beide datasets verwerkt en verfijnd, zodat deze gebruikt kunnen worden in de analyse. Vervolgens zijn, op basis van de Delft Meet Regen vrijwilligersdata, een aantal regendagen geselecteerd. Voor deze regendagen zijn de neerslagdata van zowel de vrijwilligersdata als de radardata verzameld.

De datasets hebben significante verschillen in met betrekking tot de ruimtelijke spreiding van de meetpunten. De radardata heeft een homogene spreiding, in een duidelijk rasterpatroon en met een constante aantal meetpunten. Dit is in tegenstelling tot de vrijwilligersdata. Deze kent namelijk een fluctuerende aantal metingen per dag en heeft een irreguliere ruimtelijke spreiding van meetpunten.
De ruimtelijke correlaties van beide datasets laten positieve correlatiecoëfficiënten zien voor verschillende afstand intervallen. De radardata van het KNMI heeft een hogere correlatie dan de vrijwilligersdata van het Delft Meet Regen project.
Beide datasets laten zien dat de correlatiecoëfficiënten afnemen naarmate de afstand tussen meetpunten toeneemt. Echter is deze afname sterker voor de vrijwilligersdata.
Verder zijn is er ook gezocht naar de aanwezigheid van vrijwilligersmeetpunten die systematisch een hoge of lage waarde meten. Dit is gedaan door de gemiddelde standaardwaarden van de meetpunten te evalueren. Hieruit is geen duidelijk patroon uit opgemaakt.

Tot slot is de effect van sterk afwijkende neerslagwaarden in de vrijwilligersdata geëvalueerd. Deze afwijkende waarden bleken een zekere invloed te hebben op de andere resultaten van de analyse. Hieruit is gebleken dat de aanwezigheid van deze afwijkende waarden de resultaten op een significante manier kunnen beïnvloeden.

De resultaten van deze analyse geven een inzicht in de verschillen tussen de vrijwilligersdata van het Delft Meet Regen project en de KNMI radardata. Zo zijn er duidelijke verschillen in de ruimtelijke spreiding van de meetpunten. De radardata heeft een duidelijk patroon en de vrijwilligersdata heeft dit niet, door de fluctuerende participatie van de vrijwilligers.
Ook zijn er opmerkelijke verschillen in de correlaties van de datasets. Beide datasets hebben een positieve ruimtelijke correlatie. Echter heeft de radardata een duidelijk sterkere correlatie van de vrijwilligersdata.

De uitkomsten van de analyse kunnen gebruikt worden in verdere voortgang en verbetering van het Delft Meet Regen project. De methodes en middelen die gepresenteerd zijn in dit document kunnen helpen bij de uitvoering van toekomstige analyses en onderzoeken.

Inhoud

Voorwoord 1
Samenvatting 2
1. Introductie 4
1.1 Doelstelling 4
2. Data 5
2.1 Delft Meet Regen data beschrijving 5
2.2 KNMI radardata beschrijving 5
3. Methode 6
3.1 Dataverwerking: Delft Meet Regen vrijwilligersdata 6
3.2 Dataverwerking: KNMI radardata 7
3.3 Analyse van regendagen: Selectie van regendagen 7
3.4 Analyse van regendagen: Vergelijking van meetpuntdata 8
3.5 Vergelijking van Correlaties 9
3.6 Delft Meet Regen standaardwaarden 9
4. Resultaten 11
4.1 Voorafgaande resultaten: Regenmeter vergelijking 11
4.2 Voorafgaande resultaten: Delft Meet Regen dagelijkse neerslagspreiding 11
4.3 Vergelijking van meetpunten 12
4.3.1 Delft Meet Regen Meetpunten 12
4.3.2 KNMI radarmetingen 13
4.3.3 Verschillen tussen meetpunten 14
4.4 Correlatievergelijking 16
4.5 Standaardwaarden Meetpunten 17
4.6 Invloed van afwijkende neerslagwaarden 17
5. Discussie 19
6. Conclusie 20
7. Aanbeveling 20
Verwijzingen 21
Bijlages 22
Bijlage 1: Aanvullende plots, tabellen en diagrammen 22
Bijlage 2: Python script 25

1. Introductie

Als deel van het project Delft Meet Regen wordt regendata verzameld, met behulp van een selectie vrijwilligers in Delft. De data betreft de dagelijkse neerslag voor verschillende meetlocaties in Delft. Deze data is gemeten over een periode van een aantal maanden in 2021 (Technische Universiteit Delft, 2022).

Deze verzamelde regendata kan potentieel informatie bieden die niet kan worden verkregen uit beschikbare data van het KNMI. De vrijwilligersdata van het Delft Meet Regen project zou een beeld van een neerslagpatroon kunnen vormen met een hogere resolutie dan mogelijk is door de data van het KNMI. Wellicht zou de data een beeld kunnen scheppen van hoe een stedelijk gebied de plaatselijke regenval beïnvloed.
Een deel van de vrijwilligersdata is, voorafgaand aan de analyse in dit document, gebruikt voor de uitvoering van andere onderzoeken. Een van deze onderzoeken heeft de kwaliteit en validiteit van de vrijwilligersmetingen, vergeleken met metingen van conventionelere en professionalere regenmeters (Chen, 2021). Ook zijn de metingen van de vrijwilligers gebruikt voor een onderzoek van de ruimtelijke neerslagspreiding in Delft (Beek, 2021).

De analyse die centraal staat in dit document zal voortbouwen op deze voorafgaande onderzoeken. Het hoofddoel van de analyse zal zijn om de verzamelde regendata te vergelijken met beschikbare KNMI data voor specifieke regenbuien in Delft.

De opbouw van dit document begint met het hoofdstuk Data. In dit hoofdstuk wordt een beschrijving gegeven van de data die gebruikt is in de analyse. Het daaropvolgende hoofdstuk, Methode, zal de methodes bespreken die gebruikt zijn bij de analyse van de data. Vervolgens worden, in het hoofdstuk Resultaten, de resultaten van de analyse gepresenteerd.
Daarna wordt in het hoofdstuk Discussie een kritische blik geworpen op de resultaten en het verloop van de analyse. Het volgende hoofdstuk, Conclusie, betreft het antwoord of de hoofdvraag en de conclusie van de analyse. Tot slot zal er in het hoofdstuk, genaamd Aanbeveling, een aanbeveling gegeven worden over hoe de uitkomsten van de analyse het Delft Meet Regen project kunnen bevorderen.

1.1 Doelstelling

Het hoofdzakelijke doel van dit document is het presenteren en beschrijven van een analyse. Deze analyse betreft de vergelijking van de vrijwilligersdata van het Delft Meet Regen project, met data van het KNMI. Deze analyse is geleid door de volgende hoofdvraag:

“Zijn er verschillen tussen de neerslagpatronen gemeten door de vrijwilligers van het Delft Meet Regen project en de neerslagpatronen uit de data van het KNMI?”

Er zijn meerdere vormen en van data beschikbaar via het KNMI. De andere onderzoeken voorafgaand aan dit document, (Beek, 2021) en (Chen, 2021) , hebben gebruik gemaakt van een aantal van deze datavormen. Om de scope van deze analyse te beperken, is slechts een enkele datavorm van het KNMI gebruikt; radardata.

Om voort te bouwen op de voorafgaande onderzoeken, is in deze analyse gebruik gemaakt van radardata van het KNMI. Deze data is niet gebruikt in de voorafgaande onderzoeken. Echter was het gebruik van deze data geacht als de logische volgende stap voor deze onderzoeken.

2. Data

In dit hoofdstuk wordt een overzicht gegeven van de verschillende data die gebruikt wordt in dit document.

2.1 Delft Meet Regen data beschrijving

De verzamelde vrijwilligersdata van het Delft Meet Regen project bevat verschillende gegevens. Voor elke meting die gedaan is door een vrijwilliger zijn deze gegevens verzameld. Deze data is in eerste instantie opgeslagen in een Excel bestand.

Niet al deze gegevens noodzakelijk voor de uitvoering van de analyse. Vertrouwelijke informatie is in een vroeg stadia gescheiden van de data die uiteindelijk gebruikt is. Hieronder in Tabel 1 is een overzicht van de vrijwilligersdata. Deze gegevens zijn beschikbaar voor elke meting, na het verwijderen van de vertrouwelijke informatie.

Data	**Beschrijving**
ParentGlobalID	Unieke code voor de meting
GlobalID	Unieke code voor de meting
Datum	Datum van de meting
Tijd	Tijd van de meting
Meetpunt	Benaming van het meetpunt van de meting
x	Lengtegraad coördinaat van de meting
y	Breedtegraad coördinaat van de meting
Neerslag in mm	Gemeten neerslag in millimeter
Op- en aanmerkingen toevoegen	Extra opmerkingen bij de meting van de vrijwilliger

Tabel 1. Beschrijving van Delft Meet Regen data.

2.2 KNMI radardata beschrijving

Het KNMI heeft vele datasets die informatie bieden over neerslag. Voor deze analyse zijn klimatologische radardatasets van het KNMI gebruikt (KNMI, 2022). Elke dataset bevat gegevens voor de neerslag over Nederland. De verschillende neerslagen in deze datasets, zijn gemeten over een tijdsinterval van vijf minuten en in een raster met een resolutie van 1 kilometer.

De gemeten datasets zijn verwerkt in h5-bestanden. Dit bestandtype bied de mogelijkheid om grote hoeveelheid data op te slaan. Voor de uitvoering van de analyse zijn slechts een aantal gegevens nodig. In Tabel 2 is een overzicht gegeven van deze benodigde data.

Data	**Beschrijving**
Kaart Projectie	Het coördinaten systeem dat gebruikt moet worden bij de metingen
Neerslag	De reeks neerslagwaarden; één waarde voor elke pixel in het radarbeeld.

Tabel 2. Beschrijving van KNMI radardata.

3. Methode

In dit hoofdstuk zal de opbouw en methodiek van de analyse gepresenteerd worden. De methode die hier is besproken is in zijn geheel uitgevoerd in Python. Het Python script is terug te vinden in Bijlage 2: Python script.

3.1 Dataverwerking: Delft Meet Regen vrijwilligersdata

Er zijn een aantal inconsistenties en ongewenste kwaliteiten aanwezig in de vrijwilligersdata van het Delft Meet Regen project. Voor deze redenen is de vrijwilligersdata eerst bewerkt voordat deze gebruikt is in de analyse.

Allereerst is er gekeken naar welke gegevens mogelijk niet nodig zijn voor de analyse. Dit zijn bijvoorbeeld de gegevens die niet kwantificeerbaar zijn.
Voor de vrijwilligersdata zijn dit bijvoorbeeld de 'Op- en aanmerkingen toevoegen' gegevens. Deze gegevens kunnen op het eerste gezicht niet gekwantificeerd worden in een betekenisvolle manier. Echter worden deze gegevens worden beschouwd als de metadata van de vrijwilligersdata. Deze metadata heeft de mogelijkheid om inzicht te geven in de validiteit van specifieke metingen. Hierom is gekozen om deze gegevens te behouden.
De 'ParentGlobalID' en 'GlobalID' gegevens kunnen eveneens niet gekwantificeerd worden. Desondanks zullen deze gegevens ook worden behouden. Deze gegevens kunnen namelijk worden gebruikt voor het lokaliseren en identificeren van specifieke metingen.

Het volgende aandachtspunt is de inconsistentie van de 'Neerslag in mm' gegevens. Deze waarden zijn niet door alle vrijwilligers op dezelfde manier ingevoerd. Deze invoerfouten maakt vele metingen potentieel onbruikbaar. Om het verlies van deze metingen te voorkomen, is rekening gehouden met de meest voorkomende invoerfouten. De gegevens die lijden aan deze fouten zijn gecorrigeerd.

Ten derde zijn er inconsistenties in de 'Datum' en 'Tijd' gegevens. Niet alle vrijwilligers voeren hun metingen uit op hetzelfde tijdstip iedere dag. Om hier rekening mee te houden zijn de tijdstippen van elke meting afgerond op de dag. Zo zijn de metingen die gedaan zijn in de ochtend beschouwd als een meting van de neerslag van de voorgaande dag. Een meting in de middag wordt beschouwd als een meting van de neerslag van de dag waarin de meting heeft plaatsgevonden.
Het is belangrijk om op te merken dat de gemeten neerslag in dit geval niet altijd een neerslag zal zijn die heeft plaatsgevonden over hetzelfde tijdsinterval. Verdere inconsistenties in de tijdstippen van de dagelijkse metingen worden verwaarloosd.

Tot slot is het gebied waarin de metingen van de vrijwilligers hebben plaatsgevonden afgebakend. De afbakening wordt gerealiseerd door de hoekpuntcoördinaten van dit gebied vast te stellen. Dit zorgt ervoor dat enkel de metingen binnen dit gebied meegenomen worden in de analyse. In Tabel 3 wordt een beschrijving van de verwerkte vrijwilligersdata weergegeven.

Data	**Beschrijving**
ParentGlobalID	Unieke code voor de meting
GlobalID	Unieke code voor de meting
ABS_time	Dag waarover de neerslag is gemeten
Meetpunt	Benaming van het meetpunt van de meting
x	Lengtegraad van de meting
y	Breedtegraad van de meting
Neerslag in mm	Gemeten neerslag in millimeter
Op- en aanmerkingen toevoegen	Extra opmerkingen bij de meting van de vrijwilliger

Tabel 3. Beschrijving van de Delft Meet Regen verwerkte data.

3.2 Dataverwerking: KNMI radardata

Het verwerken van de KNMI radardata betreft met name het ophalen van de benodigde data en het uitsluiten van onnodige informatie. Uit deze data zijn uiteindelijk neerslagwaardes verkregen van meetpunten over Delft.

Zoals eerder aan bod is gekomen in hoofdstuk 2, zijn er meerdere datasets in de radardata die neerslagwaardes bevatten. Deze waardes zijn gemeten over tijdsintervallen van vijf minuten en in een raster met een resolutie van één kilometer.

Om een totale neerslag voor een enkele dag te realiseren is een sommatie gedaan van alle neerslagwaardes voor een betreffende dag. Van deze totale neerslagwaardes zijn vervolgens enkel de neerslagwaardes opgeslagen die binnen het gebied van interesse liggen. Dit gebied is hetzelfde als het afgebakend gebied dat gebruikt is voor de vrijwilligersdata, besproken in paragraaf 3.1.

Bij het selecteren van de benodigde neerslagwaardes is rekening gehouden met het feit dat de KNMI radardata gebruikt maakt van specifiek coördinatensysteem (KNMI, 2022). Dit coördinatensysteem verschilt van het systeem dat gebruikt wordt in de Delft Meet Regen vrijwilligersdata.
De coördinaten van de neerslagwaardes van de KNMI radardata zijn getransformeerd naar coördinaten in het coördinatensysteem dat gebruikt wordt in de Delft Meet Regen vrijwilligersdata. Dit resulteert in een enkel coördinatensysteem dat gebruikt wordt voor alle metingen (zowel radardata als vrijwilligersdata).

De verwerkte radardata is vervolgens opgedeeld in verschillende datasets. Elk van deze individuele dataset bevat de neerslagwaardes en coördinaten voor een specifieke dag. In Tabel 4 wordt de beschrijving van de verwerkte KNMI radardata weergegeven.

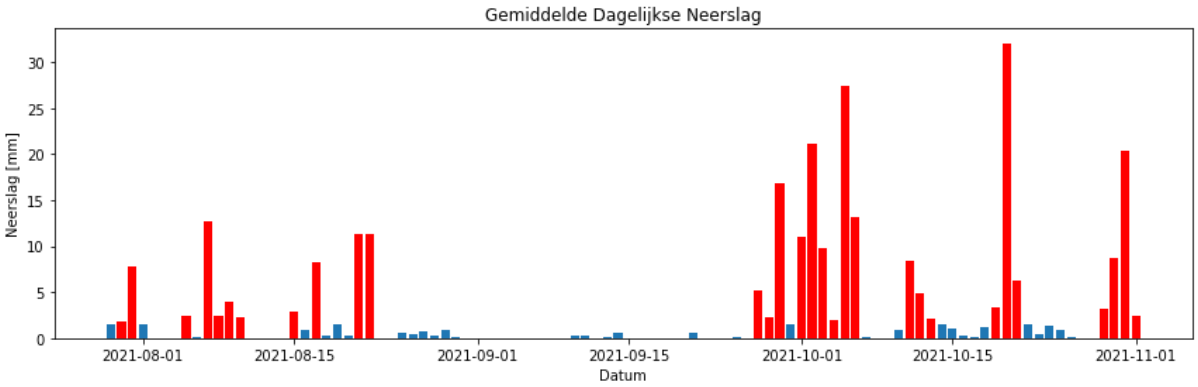
Data	**Beschrijving**
Neerslag	Neerslagwaarde in millimeter
Longitude	Lengtegraad van de meting
Latitude	Breedtegraad van de meting

Tabel 4. Beschrijving van de KNMI verwerkte radardata.

3.3 Analyse van regendagen: Selectie van regendagen

Specifieke dagen zijn gekozen waarop de neerslagwaardes van de verschillende data met elkaar vergeleken kunnen worden. Deze (regen)dagen zijn geselecteerd op basis van de gemiddelde dagelijkse neerslagwaardes van de Delft Meet Regen vrijwilligersdata.

De dertig dagen met de hoogste gemiddelde dagelijkse neerslag zijn gekozen als de geselecteerde regendagen. Voor elk van deze regendagen worden de neerslaggegevens uit zowel de vrijwilligersdata als de radardata opgehaald. In Figuur 1 is een overzicht van de gemiddelde dagelijkse neerslag gegeven, op basis van de vrijwilligersdata. Hierin zijn de geselecteerde regendagen aangegeven in het rood.

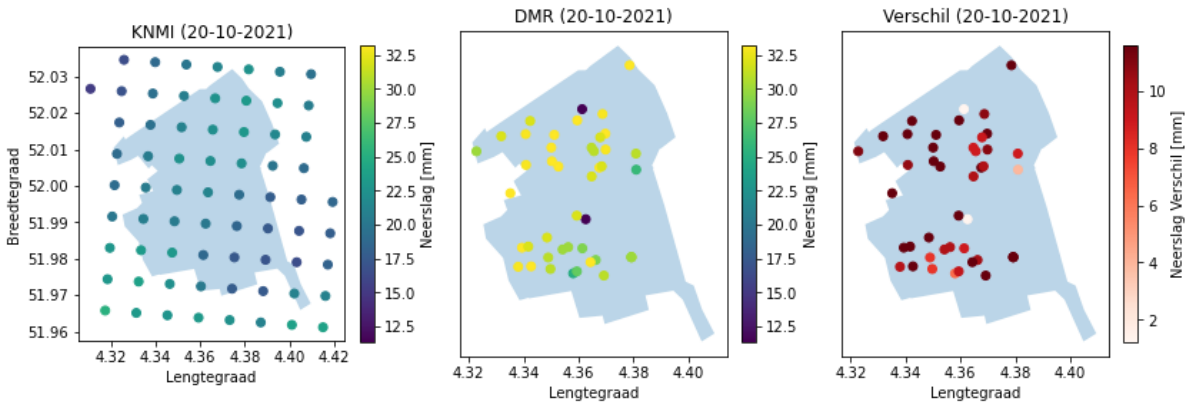


Figuur 1. Geselecteerde regendagen op basis van gemiddelde dagelijkse neerslag.

3.4 Analyse van regendagen: Vergelijking van meetpuntdata

Voor de geselecteerde regendagen worden de neerslaggegevens van de vrijwilligersdata en de radardata met elkaar vergeleken. Voor beide datasets zijn voor elke regendag de gemiddelde-, maximum- en minimum neerslagwaarden bepaald. Ook is het aantal meetpunten bepaald voor elke geselecteerde regendag. Voor de radardata zal dit aantal niet variëren. Dit aantal is namelijk gebaseerd op het aantal pixels van het radarbeeld dat zich bevindt in het afgebakende gebied van interesse.

In het uitvoeren van deze stap is gebleken dat er een aantal vrijwilligersmetingen zijn die relatief sterk afwijken van andere nabijgelegen metingen. Met behulp van de aanwezige metadata is duidelijk dat deze sterk afwijkende waarden het resultaat zijn van meetfouten. Voor deze redenen is er gekozen om de sterk afwijkende waarden te excluderen van verdere analyse.
Een gevonden waarde wordt sterk afwijkend geacht, als deze hoger is dan de dagelijkse gemiddelde neerslag plus drie maal de standaarddeviatie van de neerslagwaarden van de dag. De invloed van deze afwijkende waarden zijn in groter detail besproken in paragraaf 4.6.
Voor elke regendag is het verschil berekend tussen elke vrijwilligersmeting en de dichtstbijzijnde radarmeting. Voor alle dagen zijn eveneens de gemiddelde-, maximum- en minimum verschillen bepaald (na het excluderen van de afwijkende waarden). Hieronder in Figuur 2 is een ruimtelijk overzicht geven van deze data over de gemeente Delft voor een enkele regendag.



Figuur 2. KNMI radar metingen, Delft Meet Regen (DMR) metingen en verschillen tussen beiden in de DMR meetpunten op 20/10/2021.

3.5 Vergelijking van Correlaties

Een andere manier om de Delft Meet Regen data te vergelijken met de KNMI data, is om te kijken naar de ruimtelijke correlatie van de datasets. Hiervoor zijn de functies van de gebruikte Python packages geraadpleegd.
De benodigde gegevens voor bepalen van de ruimtelijke correlatie volgen uit de vergelijking van de meetpuntdata. Voor elke unieke combinatie van twee meetpunten op een gegeven regendag, zijn een aantal gegevens verzameld. Deze gegevens zijn de afstand tussen de meetpunten en de waarden van beiden meetpunten verzameld. Deze gegevens zijn verzameld voor alle geselecteerde regendagen, voor zowel de Delft Meet Regen dataset en de KNMI dataset.

Voor beide datasets zijn de gegevens vervolgens gerangschikt aan de hand van de afstand tussen de meetpunten. Daarna zijn de gegevens in groepen ingedeeld. Hierbij is elke groep een collectie van meetpuntcombinaties waarvan de afstand tussen de meetpunten binnen een bepaald interval valt.
Met behulp van de inbegrepen functies van de gebruikte Python packages zijn vervolgens de Pearson correlatiecoëfficiënten bepaald voor elk interval. Hierbij is gekozen om enkel een coëfficiënt te berekenen voor de intervallen met ten minste tien meetpuntcombinaties. Deze methodiek bouwt voort op de analyse uit een voorafgaand onderzoek. Het doel van dat onderzoek was een analyse van de dagelijkse neerslagspreiding van de Delft Meet Regen vrijwilligersmetingen (Beek, 2021).
Dit resulteert in een correlatiecoëfficiënt voor de verschillende intervallen. Deze zijn waarden zijn vervolgens gepaard met de gemiddelde afstanden van de meetpuntcombinaties voor de betreffende intervallen.

3.6 Delft Meet Regen standaardwaarden

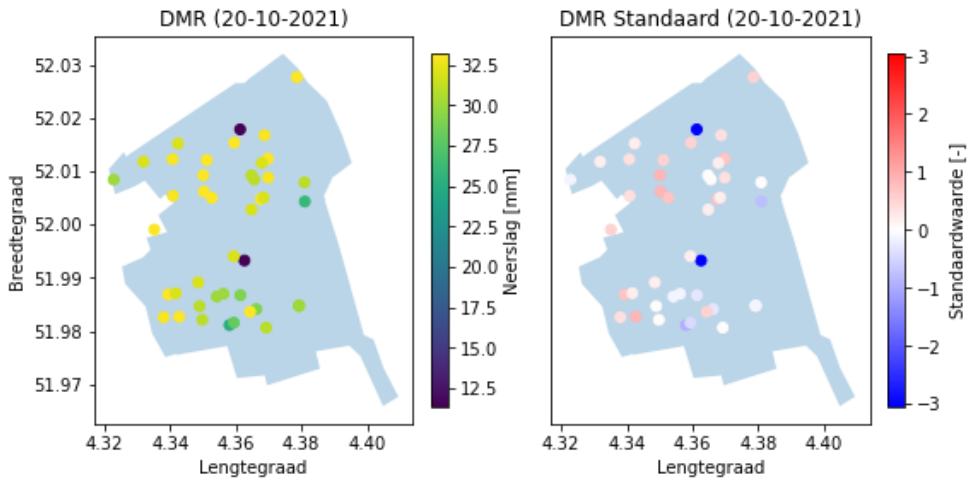
Om een beter inzicht te krijgen in de plaatselijke variaties van neerslag, is gekeken naar de gemiddelde gestandaardiseerde neerslagwaarden van de Delft Meet Regen meetpunten. Dit geeft potentieel een beeld van een systematisch hoge of lage neerslagmeting in een bepaald meetpunt.

Voor deze stap van de analyse zijn de gemeten neerslagwaarden eerst omgezet standaard waarden.
Dit geeft voor elke regendag een standaardwaarde voor elk meetpunt op de betreffende dag.
Het berekenen van de standaardwaarden is gedaan met de volgende formule:

$$v = \frac{d-m}{s}$$

v	=	gestandaardiseerde waarde [-]
d	=	neerslag meting [mm]
m	=	gemiddelde gemeten neerslag [mm]
s	=	standaarddeviatie van de neerslag [mm]

Een standaardwaarde van een meetpunt geeft aan hoeveel de neerslagwaarde van dat meetpunt afwijkt van de gemeten gemiddelde dagelijkse neerslagwaarde. Specifieker gezegd; hoeveel standaarddeviaties een neerslagwaarde afwijkt van het gemiddelde.
Een positieve of negatieve standaardwaarde voor een meetpunt, betekent dat de neerslagwaarde voor dat meetpunt respectievelijk hoger of lager is dan de gemiddelde dagelijkse neerslagwaarde. Een grotere absolute standaardwaarde betekent een grotere afwijking van het gemiddelde.
In Figuur 3 is een ruimtelijk overzicht gegeven van de standaardwaarden van een specifieke regendag, naast de gemeten neerslagwaarden.



Figuur 3. Delft Meet Regen gemeten neerslagwaarden en berekende standaardwaarden voor 20/10/2021.

Voor elk meetpunt wordt een gemiddelde genomen van alle standaardwaarde. Deze gemiddelde standaardwaarden geven een indicatie van hoeveel elke gemeten neerslagwaarde in een bepaald meetpunt gemiddeld afwijkt van de dagelijkse gemiddelde neerslag.
Echter zullen niet alle meetpunten een gemeten neerslagwaarde hebben voor elke regendag. Om hier rekening mee te houden, zijn alleen de meetpunten geëvalueerd die minstens voor de helft van alle geselecteerde regendagen een gemeten neerslagwaarde hebben.

4. Resultaten

In dit hoofdstuk zullen de resultaten van de analyse gepresenteerd en besproken worden, evenals de resultaten van de onderzoeken die voorafgingen aan deze analyse.

4.1 Voorafgaande resultaten: Regenmeter vergelijking

Voorafgaand aan de analyse besproken in dit document, is de vrijwilligersdata van het Delft Meet Regen project reeds gebruikt voor andere onderzoeken. Eén van deze onderzoeken betreft een vergelijking van regenmeters. In dit onderzoek zijn de regenmeters, gebruikt door de vrijwilligers van het Delft Meet Regen project, vergeleken met regenmeters gebruikt door het KNMI (Chen, 2021).

Uit dit onderzoek is gebleken dat de metingen van de regenmeters van het Delft Meet Regen project dezelfde patronen laten zien als die van het KNMI. De resultaten van dit onderzoek kunnen dienen als een indicatie van de nauwkeurigheid van de vrijwilligersmetingen. De resultaten van de analyse beschreven in dit document, bouwen voort op deze voorafgaande resultaten door een nieuwe vergelijking te maken; een vergelijking van de Delft Meet Regen vrijwilligersdata en de radardata van het KNMI.

4.2 Voorafgaande resultaten: Delft Meet Regen dagelijkse neerslagspreiding

Een tweede onderzoek dat heeft plaatsgevonden vóór de uitvoering van de analyse beschreven in dit document, geeft een beeld van de ruimtelijke spreiding van de vrijwilligersdata van het Delft Meet Regen project. In dit onderzoek is namelijk gekeken naar de dagelijkse neerslagspreiding op basis van de vrijwilligersdata van het Delft Meet Regen project (Beek, 2021).

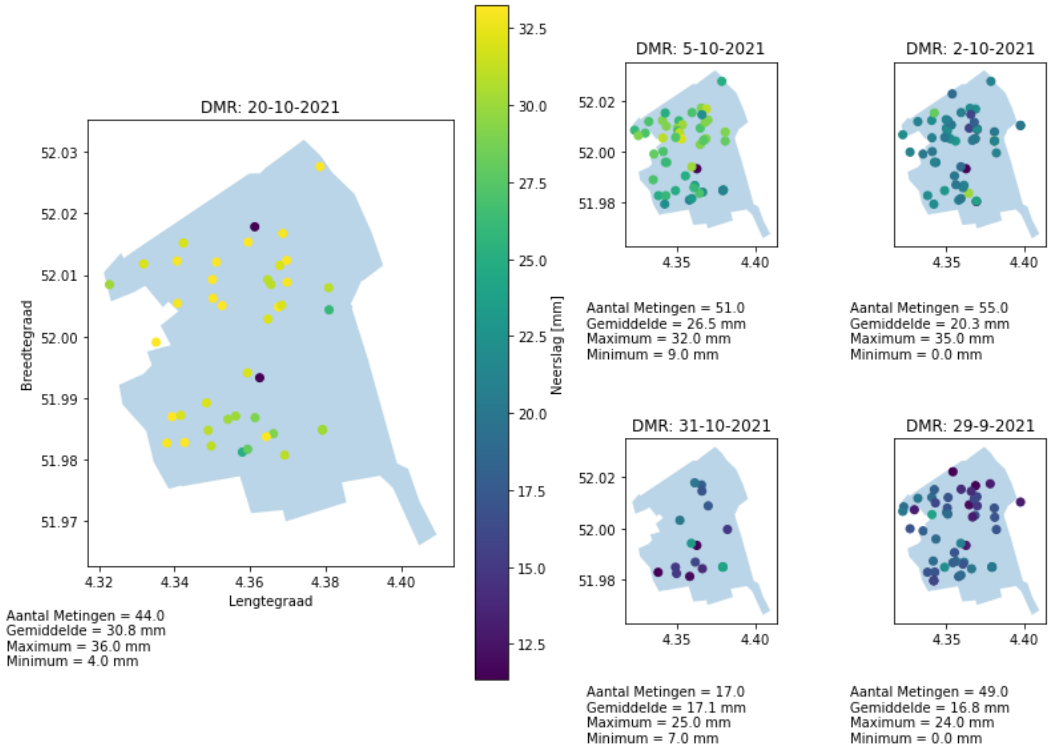
De resultaten van dit onderzoek laten zien dat er inderdaad sprake is van een ruimtelijke correlatie van de vrijwilligersmetingen van het Delft Meet Regen project. Echter heeft dit onderzoek geen patroon in de dagelijkse neerslagspreiding kunnen vaststellen.

4.3 Vergelijking van meetpunten

De resultaten die hier besproken worden focussen zich op de ruimtelijke spreiding van de meetpunten en de verschillen in neerslagwaarden tussen data van het Delft Meet Regen project en de data van het KNMI. Deze resultaten volgen direct uit de methode besproken in paragraaf 3.4.

4.3.1 Delft Meet Regen Meetpunten

Voor de geselecteerde regendagen zijn de vrijwilligersmetingen verzameld die gedaan zijn als deel van het Delft Meet Regen project. In Figuur 4 is een overzicht gegeven van deze metingen voor de vijf geselecteerde regendagen met de hoogste gemiddelde dagelijkse neerslag.

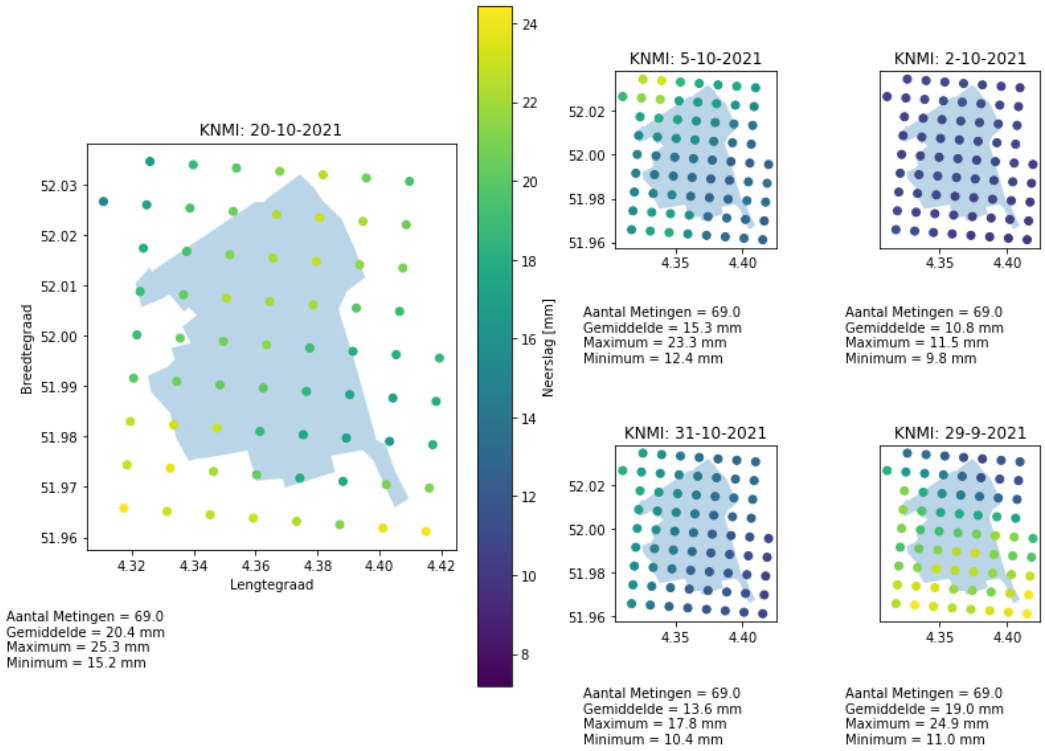


Figuur 4. Delft Meet Regen metingen voor de vijf regendagen met de hoogste dagelijkse gemiddelde neerslag.

In Figuur 4 is te zien dat het aantal metingen voor sterk kan verschillen voor elke individuele regendag. Zo is bijvoorbeeld goed te zien dat de regendag met het vierde hoogste dagelijkse gemiddelde een significante hoeveelheid minder metingen heeft. Naast het fluctuerende aantal metingen, is het ook opmerkelijk dat er instanties zijn dat er relatief grote verschillen zijn in de neerslagwaarden van nabijgelegen meetpunten.

4.3.2 KNMI radarmetingen

Zoals besproken is in paragraaf 3.4, zijn ook de KNMI metingen verzameld voor alle geselecteerde regendagen. In Figuur 5 is een overzicht gegeven van de KNMI radarmetingen voor de vijf geselecteerde regendagen met de hoogste gemiddelde dagelijkse neerslag volgens de Delft Meet Regen vrijwilligersdata.



Figuur 5. KNMI radar metingen voor de vijf regendagen met de hoogste gemiddelde neerslag volgens de Delft Meet Regen vrijwilligersdata.

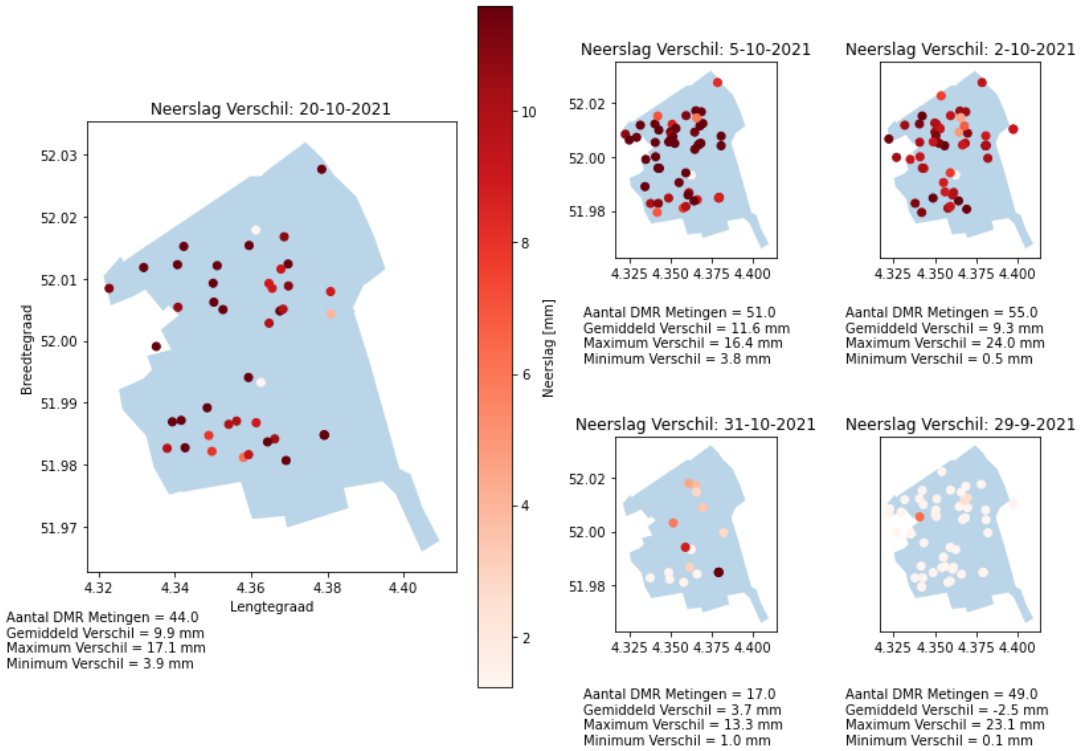
Belangrijk om op te merken, is dat de radarmetingen voor de geselecteerde regendagen niet gerangschikt zijn op de gemiddelde dagelijkse neerslag van de radardata. Met andere woorden; de dagen van de radarmetingen in Figuur 5 zijn niet gerangschikt volgens de neerslagwaarden van de radarmetingen. Deze zijn gerangschikt volgens de neerslagwaarden van de vrijwilligersmetingen voor dezelfde regendagen.

Een tweede opmerking die gemaakt kan worden betreft de ruimtelijke spreiding en oriëntatie van de radarmeetpunten. Een duidelijk rasterpatroon is te herkennen met een vast aantal meetpunten. Deze zijn te verklaren door het feit dat deze afkomstig zijn van radarbeelden. Elk radarmeetpunt zichtbaar in de figuur correspondeert met een enkele pixel van het oorspronkelijke radarbeeld.
De oorspronkelijke radarbeelden hebben een resolutie van 1 kilometer. Dit zorgt ervoor dat de neerslagwaarden van de meetpunten in de figuur elk een gemiddelde neerslagwaarde zijn over een gebied van 1 km^2.

De ‘scheve’ oriëntatie van de radarmetingen is te verklaren door het verschil in de kaartprojectie tussen de oorspronkelijke radarbeelden en de kaartprojectie van de figuur. De transformatie van de coördinaten resulteert in de ‘scheve’ oriëntatie van het raster.

4.3.3 Verschillen tussen meetpunten

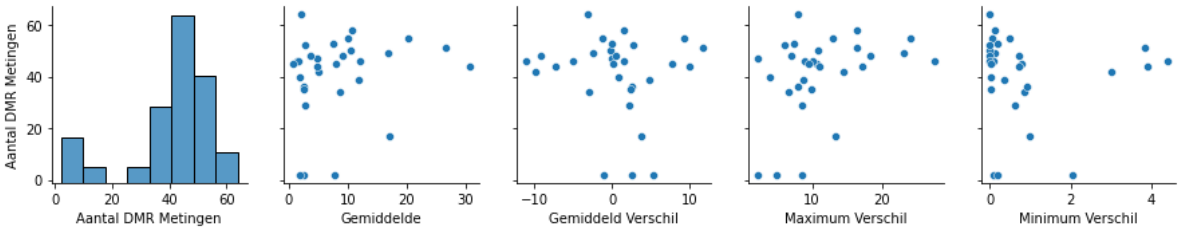
Voor alle regendagen zijn de verschillen bepaald tussen elke vrijwilligersmeting en de dichtstbijzijnde radarmeting van dezelfde dag. Deze verschillen zijn hieronder in Figuur 6 weergegeven voor de vijf geselecteerde regendagen met de hoogste gemiddelde dagelijkse neerslag.



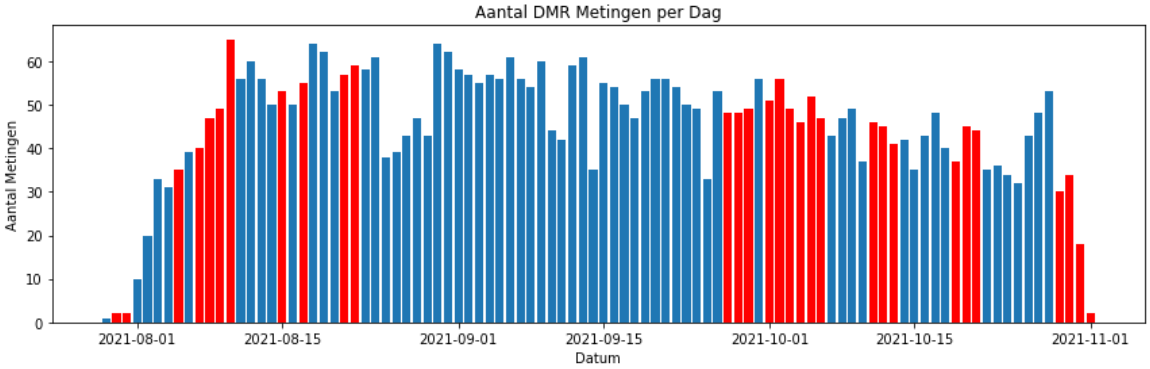
Figuur 6. Verschil tussen Delft Meet Regen metingen en KNMI radarmetingen voor de vijf regendagen met de hoogste gemiddelde neerslag volgens de Delft Meet Regen metingen.

Figuur 6 schept een beeld van hoe de vrijwilligersmetingen van het Delft Meet Regen project afwijken van de KNMI. Zo is te zien dat voor de dagen met de vier hoogste gemiddelden, de meeste vrijwilligersmetingen hoger zijn dan de radarmetingen. Ook hier is weer te zien dat er sterke verschillen zijn tussen de neerslagmetingen van sommige nabijgelegen meetpunten.

Met deze stap van de analyse is het fluctuerende aantal vrijwilligersmetingen beter in kaart te brengen. Dit geeft een goed beeld van de participatie van de vrijwilligers en ook hoe dit gereflecteerd is in de geselecteerde regendagen.
Dit is te zien in Figuur 7, waar het aantal vrijwilligersmetingen is geplot tegen de gemiddelde dagelijkse neerslag, het gemiddelde verschil tussen de Delft Meet Regen metingen en de KNMI radarmetingen, het maximum verschil tussen deze metingen, en het minimum verschil tussen deze metingen.

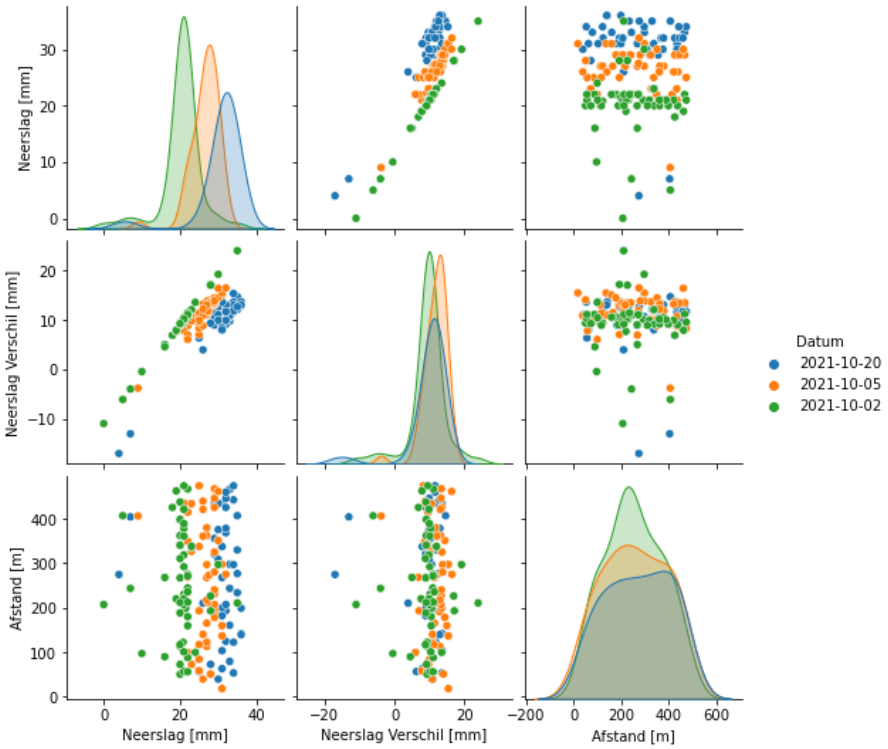


Figuur 7. Aantal vrijwilligersmetingen met neerslag verschillen van geselecteerde regendagen.



Figuur 8. Aantal vrijwilligersmetingen met geselecteerde regendagen.

Figuur 8 laat zien hoeveel vrijwilligersmetingen hebben plaatsgevonden op de geselecteerde regendagen. In deze figuur zijn de geselecteerde regendagen rood gekleurd.

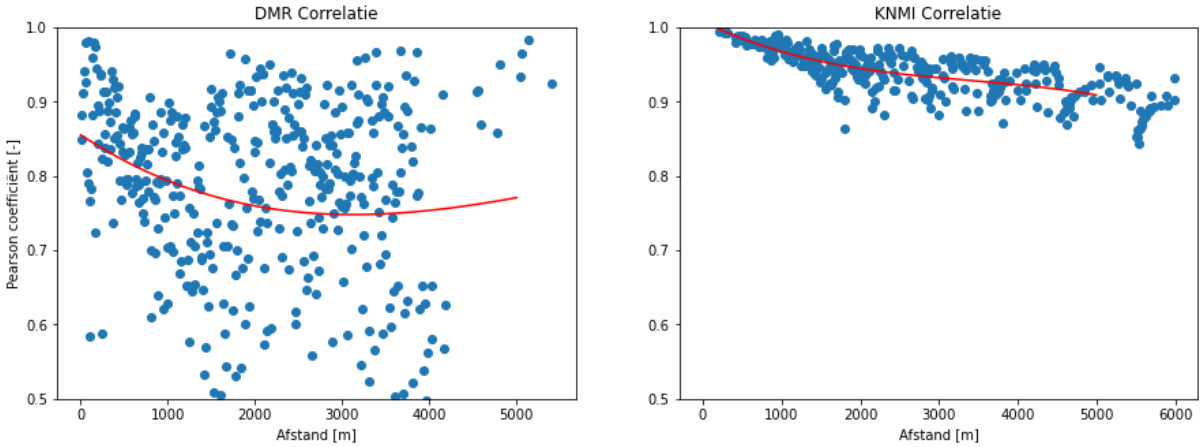


Figuur 9. Crossplot van de drie regendagen met de hoogste gemiddelde dagelijkse neerslag.

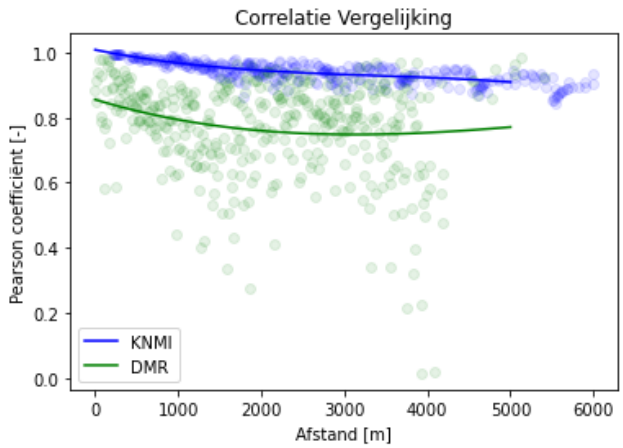
In Figuur 9 zijn een aantal gegevens geplot voor de drie regendagen met het hoogste gemiddelde dagelijkse neerslag, gebaseerd op de Delf Meet Regen vrijwilligersdata. De neerslag in deze figuur, zijn de neerslagwaarden van de individuele vrijwilligersmetingen. Het neerslag verschil, is het verschil tussen de vrijwilligersmetingen en ieders dichtstbijzijnde KNMI neerslagwaarde. De afstand aangegeven in de figuur is de afstand tussen elk vrijwilligersmeetpunt en de dichtstbijzijnde KNMI meetpunt.

4.4 Correlatievergelijking

De resultaten die weergegeven zijn in Figuur 10 volgen direct uit de methode besproken in paragraaf 3.5. Zoals daar besproken is, zijn de correlatiecoëfficiënten elk bepaald voor een specifiek interval. Figuur 10 is gebaseerd op intervallen van elk 10 meter.



Figuur 10. Correlaties van Delft Meet Regen data (DMR) en KNMI data.



Figuur 11. Correlatievergelijking van Delft Meet Regen data (DMR) en KNMI data.

In Figuur 11 zijn de overeenkomsten en verschillen tussen de correlaties van de datasets goed te zien. Zo blijkt dat beide datasets een positieve lineaire correlatie hebben. Dit betekent dat, in de meeste gevallen, een grotere afstand tussen meetpunten gepaard gaat met een groter verschil tussen de neerslagwaarden van de meetpunten.

In de figuur is te zien dat deze correlatie tussen de afstand en het verschil in neerslagwaarden, relatief hoog is voor beide datasets. Dit is vooral waar op korte afstanden.

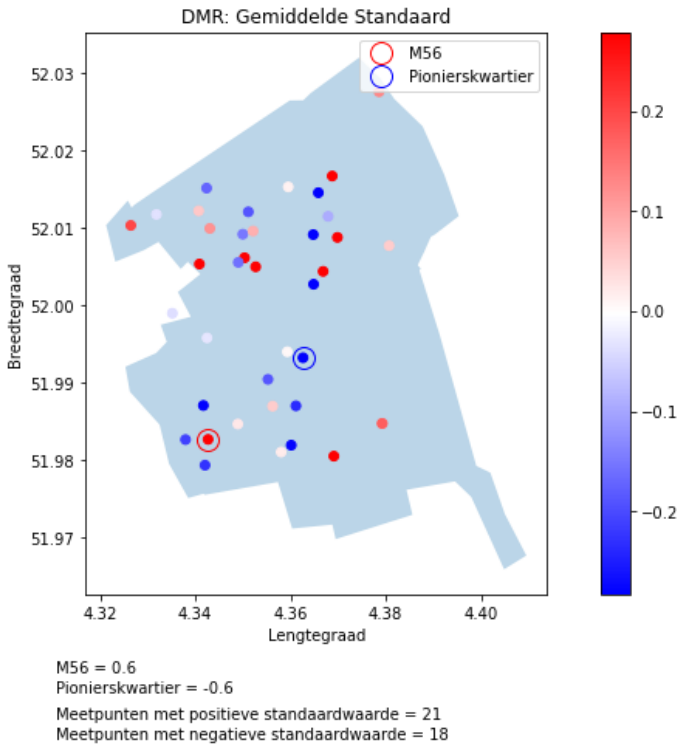
Voor grotere afstanden lijkt deze correlatie voor de beide datasets af te nemen. Echter is duidelijk te zien dat de correlatiecoëfficiënten voor de KNMI data voor alle afstanden, hoger is dan de data van het Delft Meet Regen project. Ook is te zien dat de spreiding van de correlatiecoëfficiënten significant groter is voor de Delft Meet Regen data.

Dit verschil kan een reflectie zijn van de verschillende methode waarop de neerslagwaarden gemeten zijn. De resolutie van de KNMI kan ervoor zorgen dat zeer lokale regenbuien niet gerepresenteerd worden in de metingen. Echter zal de vrijwilligersdata deze lokale intensiteit mogelijk wel kunnen detecteren door de korte afstand tussen meetpunten.

Zeer lokale regenbuien zullen een relatief hoge positieve correlatie hebben op korte afstanden. Deze correlatie zal dan sterk afnemen naarmate de afstand tussen het centrum van een regenbui en een meetpunt groter wordt.

4.5 Standaardwaarden Meetpunten

Figuur 12 kan inzicht geven in hoeveel specifieke meetpunten systematisch een neerslagwaarde meten boven, of onder de gemiddelde dagelijkse neerslag. Deze figuur en de resultaten besproken in deze paragraaf volgen direct uit de methode besproken in paragraaf 3.6.
In Figuur 12 zijn de gemiddelde standaardwaarden van vrijwilligersmeetpunten weergegeven. Deze gemiddelde waarde is gebaseerd op de standaardwaarden van elk meetpunt voor alle geselecteerde regendagen. De meetpunten met de hoogste en laagste gemiddelde standaardwaarde zijn omcirkelt in de figuur.



Figuur 12. Gemiddelde standaardafwijking van Delft Meet Regen metingen.

Opmerkelijk in deze resultaten, is het feit dat sommige nabijgelegen meetpunten een relatief groot verschil in gemiddelde standaardwaarden laten zijn. Een aantal van deze verschillen in de figuur zijn terug te zien in de resultaten van de meetpuntvergelijkingen in paragraaf 4.3.

Het bereik van de gemiddelde standaardwaarden is wellicht te klein te spreken van systematische hoge, of lage metingen. Ook kan door de eerder besproken verschillen geen duidelijk ruimtelijk patroon herkent worden.

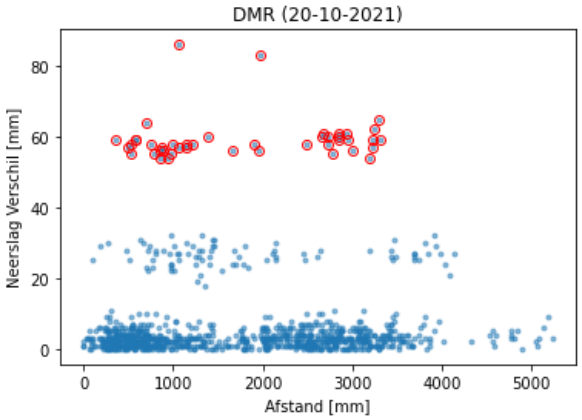
4.6 Invloed van afwijkende neerslagwaarden

In paragraaf 3.4 is besproken dat in de loop van de analyse is gebleken dat er sterk afwijkende neerslagwaarden aanwezig zijn in de vrijwilligersdata. Deze waarden worden in de analyse niet meegenomen. Echter is het belangrijk voor de algemene comprehensie van de vrijwilligersdata, wat voor invloed deze afwijkende waarden hebben op de resultaten van de analyse.

Als een blik geworpen wordt op de metadata van de vrijwilligersdata, wordt een mogelijke oorzaak geven voor deze sterk afwijkende waarden. Uit de opmerkingen van de vrijwilligers blijkt dat meetfouten de oorzaak zijn. In deze gevallen is de gemeten neerslagwaarde in feite een cumulatieve neerslagwaarde die gemeten is over meerdere dagen.

Eén van de manieren dat deze afwijkende waarden een vertekend beeld kunnen geven van de resultaten is in de correlaties van de vrijwilligersdata. De meetpuntcombinaties die een sterk afwijkende waarde bevatten zullen een relatief hoog verschil in neerslag hebben.
Deze situatie is weergegeven in Figuur 13. Deze figuur laat, voor alle unieke meetpuntcombinaties, het verschil zien in de gemeten neerslagwaarden en de afstand tussen meetpunten zien. Hierin zijn

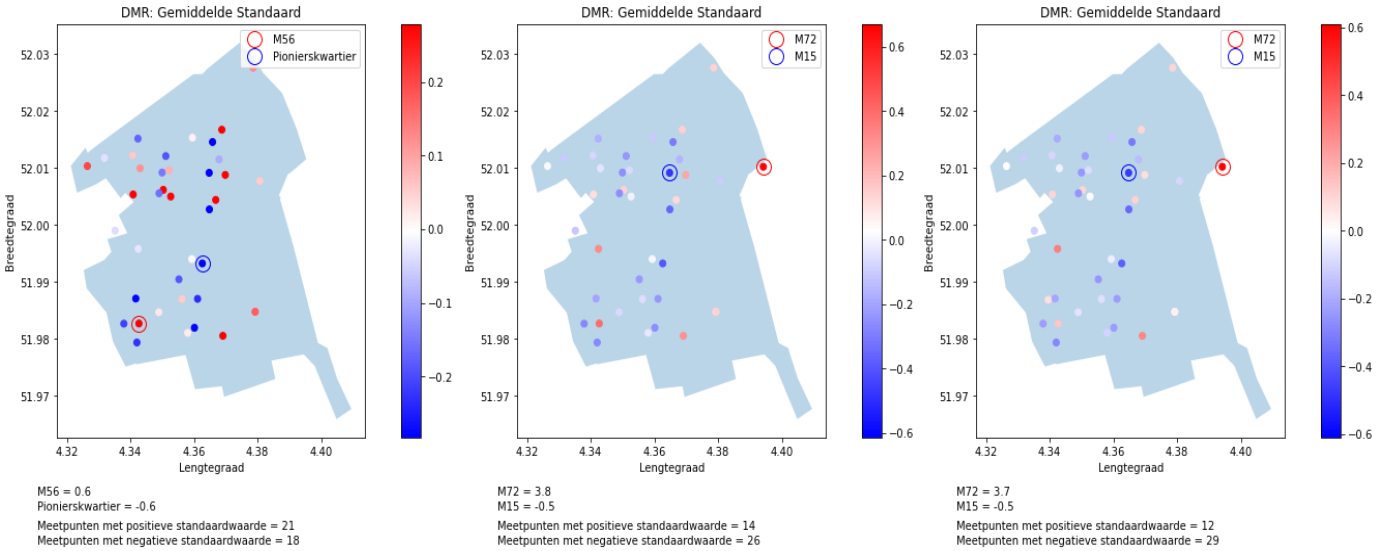
de sterk afwijkende neerslagmetingen meegenomen. De meetpuntcombinaties die, voor deze specifieke regendag, de hoogste (afwijkende) neerslagwaarde bevat zijn omcirkelt.



Figuur 13. Meetpuntcombinaties met sterk afwijkende neerslagwaarden voor 20/10/2021.

De resultaten die wellicht het meest beïnvloed zijn door de sterk afwijkende waarden, zijn de gemiddelde standaardwaarden die besproken zijn in paragraaf 4.5. Dit is te danken aan het feit dat de gemiddelde dagelijkse neerslag sterk beïnvloed kan worden door de afwijkende waarden.

In Figuur zijn drie overzichten van de gemiddelde standaardwaarden weergegeven. Deze laten respectievelijk een overzicht zien zonder afwijkende waarden, een overzicht zonder afwijkende waarden waarbij een opmerking is geplaatst, en een overzicht waarbij alle afwijkende waarden zijn meegenomen.



Figuur 14. Overzicht van gemiddelde standaardwaarden waarbij, respectievelijk geen afwijkende waarden zijn meegenomen, geen afwijkende waarden meegenomen met een opmerking, en alle afwijkende waarden zijn meegenomen.

De overzichten laten zien dat de aanwezigheid van de sterk afwijkende waarden een vertekend beeld geven van de gemiddelde dagelijkse neerslag. Ook is te zien dat het gebruik van opmerkingen voor de identificatie van sterk afwijkende waarden, niet voldoende is om alle afwijkende waarden te vinden. In de metadata is namelijk te zien dat niet alle afwijkende waarden een opmerking hebben.

Extra informatie zal dus nodig zijn om sterk afwijkende waarden ‘eerlijk’ te excluderen. Echter is dit niet mogelijk met de metadata die gebruikt is in deze analyse.

5. Discussie

In dit hoofdstuk zal de kwaliteit en validiteit van de methodes en resultaten in dit document besproken worden. Hierbij zal vooral aandacht besteed worden op de aspecten van de hoofdzakelijke analyse van dit document.

In dit document zijn data, methodes en resultaten besproken. Deze aspecten bouwen voort op de onderzoeken die hebben plaatsgevonden voorafgaand aan dit document. De resultaten van deze onderzoeken zijn besproken in paragraaf Voorafgaande resultaten: Regenmeter vergelijking4.1 en 4.2.

Voor de betrouwbaarheid van de vrijwilligersmetingen zal verwezen moeten worden naar het onderzoek dat zich focust op de kwaliteit van de methode en apparatuur die gebruikt wordt voor de vrijwilligersmetingen (Chen, 2021).
Naast de validiteit van de metingen zijn er een aantal inconsistenties aanwezig in de vrijwilligersdata van het Delft Meet Regen project. Deze inconsistenties zijn aan bod gekomen in paragraaf 2.1 en 3.1. Deze inconsistenties betreffen voornamelijk invoerfouten en meetfouten. Een groot deel van deze fouten zijn gecorrigeerd. Echter zijn deze correcties gedaan op basis van de meest voorkomende fouten die geobserveerd zijn. Hierdoor is alsnog niet alle beschikbare data uiteindelijk gebruikt voor het uitvoeren van de analyse.

Een ander aspect dat in twijfel kan worden gebracht is de selectie van de regendagen, beschreven in paragraaf 3.3. De regendagen die gebruikt worden in de analyse zijn namelijk gebaseerd op de gemiddelde dagelijkse neerslag die verkregen wordt uit de vrijwilligersdata. Echter is in de loop van de analyse duidelijk geworden dat er sterk afwijkende neerslagwaarden zijn die de gemiddelde dagelijkse neerslag beïnvloeden. De invloed van deze afwijkende waarden zijn besproken in paragraaf 4.6.
Er is nog een aspect wat de validiteit van de analyse in twijfel kan brengen, met betrekking tot de sterk afwijkende neerslagwaarden. Dit is namelijk de manier waarop deze waarden geïdentificeerd worden. De manier waarop dat in deze analyse is gedaan kan arbitrair geacht worden, aangezien dit enkel gebaseerd is op de mate waarin een waarde afwijkt van de gemiddelde waarde.
Een betere manier zou gebaseerd kunnen zijn op de metadata van de vrijwilligersdata. Echter is dit niet gerealiseerd voor deze analyse.

De gemiddelde dagelijkse neerslagwaarden worden opnieuw berekend na het excluderen van de sterk afwijkende neerslagwaarden. Dit houdt in dat de afwijkende waarden niet worden gebruikt in de rest van de analyse. Echter is de selectie van de regendagen nog wel gebaseerd op de vrijwilligersdata waarbij de sterk afwijkende neerslagwaarden nog aanwezig waren.
Een betere selectie van de regendagen zou gedaan kunnen worden op basis van de vrijwilligersdata. In deze betere selectie zouden eerst de sterk afwijkende neerslagwaarden verwijdert moeten worden. Ook kan een minimum aantal vrijwilligersmetingen worden gehanteerd om ervoor te zorgen, dat elke geselecteerde regendag een minimum aantal metingen bevat.

Eén van de aspecten die niet behandelt is in dit document betreft de verschillen tussen de neerslagwaarden van sommige nabijgelegen vrijwilligersmeetpunten. In paragraaf 4.3 en 4.5 worden de resultaten gepresenteerd waar deze verschillen naar voren komen. Deze analyse heeft geen duidelijke oorzaak kunnen vinden voor deze verschillen. De enige mogelijke oorzaak die gegeven kan worden op basis van de analyse, is dat de meet- en invoerfouten hiervoor verantwoordelijk zijn.

Ten slotte is in dit document enkel gebruik gemaakt van specifieke radardata van het KNMI. Zoals in paragraaf 2.2 besproken is, zijn er meerdere datavormen beschikbaar gesteld door het KNMI. De

vergijking die gedaan wordt in dit document heeft dus alleen betrekking tot de specifieke datasets die zijn gebruikt.

6. Conclusie

Uit de resultaten van de analyse kan geconcludeerd worden dat er verschillen zijn tussen de neerslagpatronen gemeten door de vrijwilligers van het Delft Meet Regen project en de neerslagpatronen uit de data van het KNMI. Dit zijn verschillen in de data, de ruimtelijke spreiding van de meetpunten, en de correlatie van de neerslagwaarden gemeten in de meetpunten.

Allereerst bieden de data en het ruimtelijk patroon van de meetpunten verschillende informatie. De radardata van het KNMI heeft een vaste hoeveelheid metingen voor iedere dag. Dit is in tegenstelling tot de vrijwilligersdata van het Delft Meet Regen project. Het aantal metingen aanwezig in de vrijwilligersdata is namelijk afhankelijk van de participatie van de vrijwilligers voor een gegeven dag.

De vrijwilligersdata laten instanties zien van relatieve hoge verschillen in neerslagwaarden over relatief korte afstanden. Dit is in tegenstelling tot de radardata van het KNMI. Deze instanties zijn enigszins gereflecteerd in de verschillen in ruimtelijke correlatie tussen de vrijwilligersdata en radardata.
De ruimtelijke correlatiecoëfficiënten van de Delft Meet Regen vrijwilligersdata heeft een significant hogere spreiding dan de coëfficiënten van de KNMI radardata. De radardata lijkt ook algemeen hogere correlatie te hebben dan de vrijwilligersdata. Beide datasets laten wel een positieve correlatie zien voor alle afstandsintervallen.

De instanties van de hoge verschillen op relatief korte afstanden is in kaart gebracht met gemiddelde standaardwaarden van de vrijwilligersmeetpunten. Hierin is geen duidelijk systematisch patroon te herkennen. Echter blijven deze instanties van hoge verschillen op relatief korte afstanden zich voordoen voor zowel de gemiddelde standaardwaarden als de meetpuntvergelijkingen.

Ten slotte zijn de invloeden neerslagwaarden, gemeten door vrijwilligers, die sterk afwijken van de andere vrijwilligersmetingen besproken. Hieruit is duidelijk geworden dat de aanwezigheid van deze waarden een relatief grote impact kunnen hebben op de resultaten van de analyse. Met name de resultaten waarbij het verschil tussen neerslagwaarden of de gemiddelde neerslagwaarde bij betrokken zijn. De aanwezigheid kan voor een vertekend beeld zorgen.

7. Aanbeveling

De analyse beschreven in dit document is te danken aan de vrijwilligersdata van het Delft Meet Regen project. De hoop is dat de obstakels en vindingen die resulteren uit dit document gebruikt kunnen worden voor de voorgang en mogelijke verbetering van het Delft Meet Regen project. Net als de onderzoeken die voorafgingen aan dit document kunnen de middelen en resultaten van dit document gebruikt worden voor nieuwe onderzoeken en analyses.

Een mogelijk aandachtspunt voor toekomstige onderzoeken is het gebruik van de metadata in de vrijwilligersdata om de validiteit van de metingen te evalueren. Echter zal de metadata waarschijnlijk uitgebreid moeten worden om deze vertrouwelijk te kunnen gebruiken.

Verwijzingen

Beek, R. v. (2021, Oktober). Bacheloreindproject Delf Meet Regen. *Onderzoek naar de dagelijkse neerslagspreiding van Delft op basis van de neerslagdata van Delft Meet Regen*.

Chen, K. (2021, Oktober). Bacheloreindproject Delft Meet Regen. *Vergelijking van de regenmeters van Delft Meet Regen met andere regenmeters*.

KNMI. (2022, 4 28). *Precipitation - 5 minute precipitation accumulations from climatological gauge-adjusted radar dataset for The Netherlands (1 km, extended mask) in KNMI HDF5 format.* Opgehaald van dataplatform.knmi.nl: https://dataplatform.knmi.nl/dataset/rad-nl25-rac-mfbs-em-5min-2-0

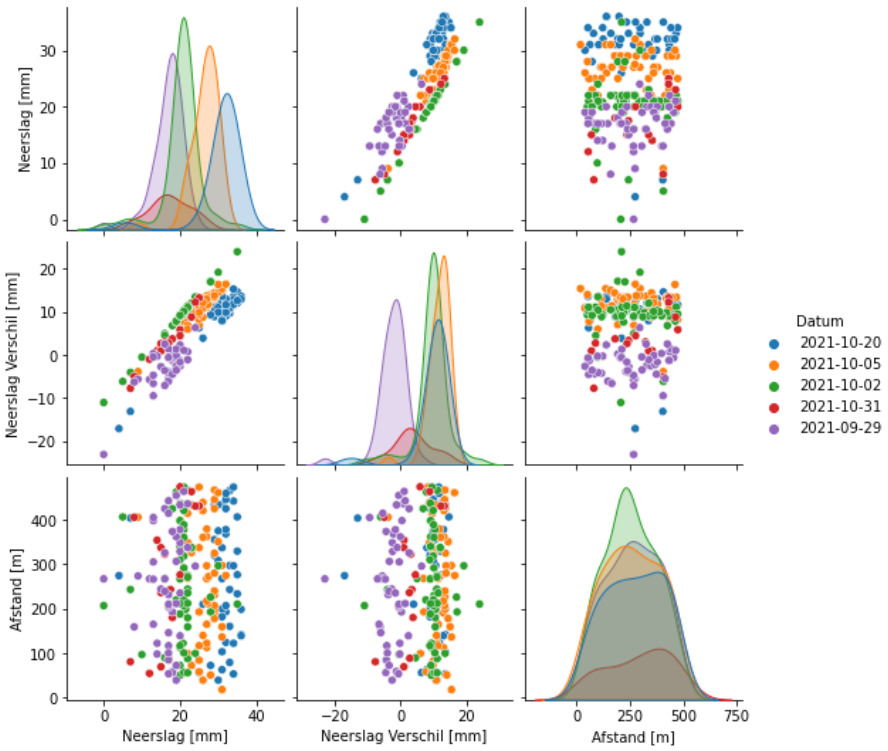
Technische Universiteit Delft. (2022, April). *Project 7: Delft Meet Regen resultaten*. Opgehaald van tudelft.nl: https://www.tudelft.nl/scd/waterlab/doe-mee-aan-onderzoek/project-7-delft-meet-regen

Bijlages

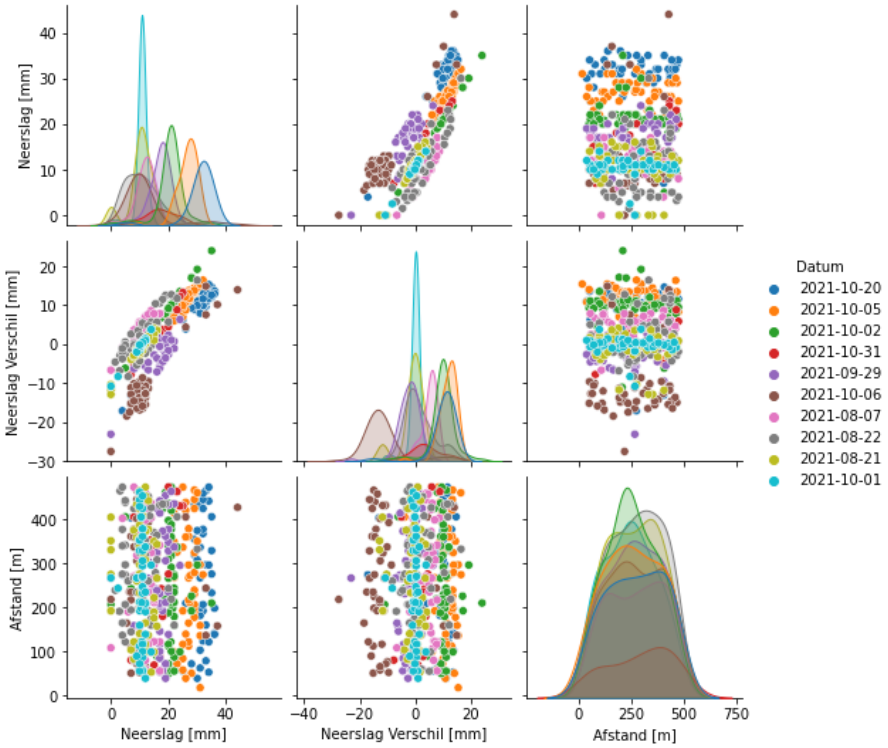
Bijlage 1: Aanvullende plots, tabellen en diagrammen

	Gemiddelde [mm]	Maximum [mm]	Minimum [mm]	Gemiddeld Verschil [mm]	Maximum Verschil [mm]	Minimum Verschil [mm]	Aantal DMR Metingen
2021-10-20	30.750000	36.0	4.0	9.883182	17.07	3.910000e+00	44.0
2021-10-05	26.466667	32.0	9.0	11.631373	16.43	3.850000e+00	51.0
2021-10-02	20.296364	35.0	0.0	9.318727	23.98	4.900000e-01	55.0
2021-10-31	17.147059	25.0	7.0	3.692941	13.26	9.700000e-01	17.0
2021-09-29	16.775510	24.0	0.0	-2.485102	23.07	1.100000e-01	49.0
2021-10-06	12.145652	44.0	0.0	-11.183261	27.49	4.400000e+00	46.0
2021-08-07	11.769231	17.0	0.0	4.708462	8.66	3.600000e-01	39.0
2021-08-22	10.613793	30.0	1.5	1.491034	16.42	1.100000e-01	58.0
2021-08-21	10.109091	16.0	0.0	-1.285818	12.86	6.000000e-02	55.0
2021-10-01	10.470600	14.0	0.0	-0.227000	10.78	3.552714e-15	50.0
2021-10-03	9.183333	35.0	0.0	-9.252500	18.34	7.000000e-01	48.0
2021-10-30	8.673529	11.0	5.0	-2.950588	6.58	8.500000e-01	34.0
2021-10-11	7.976667	11.0	1.0	7.681333	10.71	7.700000e-01	45.0
2021-08-17	7.501887	15.0	1.0	-0.145660	7.36	1.800000e-01	53.0
2021-07-31	7.750000	11.0	4.5	5.280000	8.53	2.030000e+00	2.0
2021-10-21	4.980952	29.0	1.0	-9.825000	14.53	3.000000e+00	42.0
2021-09-27	4.845745	7.0	3.0	-0.143830	2.27	2.000000e-02	47.0
2021-10-12	4.675000	13.0	2.0	-7.333182	10.95	7.000000e-01	44.0
2021-08-09	3.629167	10.0	0.0	0.521250	7.03	0.000000e+00	48.0
2021-10-19	2.494444	8.0	1.0	2.435556	8.00	9.100000e-01	36.0
2021-10-29	2.686207	9.0	1.0	2.255862	8.54	6.100000e-01	29.0
2021-08-15	2.690385	6.0	0.0	2.690385	6.00	0.000000e+00	52.0
2021-11-01	2.500000	5.0	0.0	2.450000	4.97	7.000000e-02	2.0
2021-08-08	1.407609	14.0	0.0	-5.016957	9.02	7.000000e-02	46.0
2021-08-05	2.437143	10.0	0.0	2.406286	9.99	2.000000e-02	35.0
2021-09-28	1.556087	10.0	0.0	1.544565	10.00	0.000000e+00	46.0
2021-08-10	2.112500	7.0	0.0	-3.130625	8.00	8.881784e-16	64.0
2021-10-13	1.927500	5.0	0.9	0.819250	3.99	3.000000e-02	40.0
2021-10-04	0.704444	10.0	0.0	0.095556	9.55	3.000000e-02	45.0
2021-07-30	1.750000	3.0	0.5	-1.070000	2.32	1.800000e-01	2.0

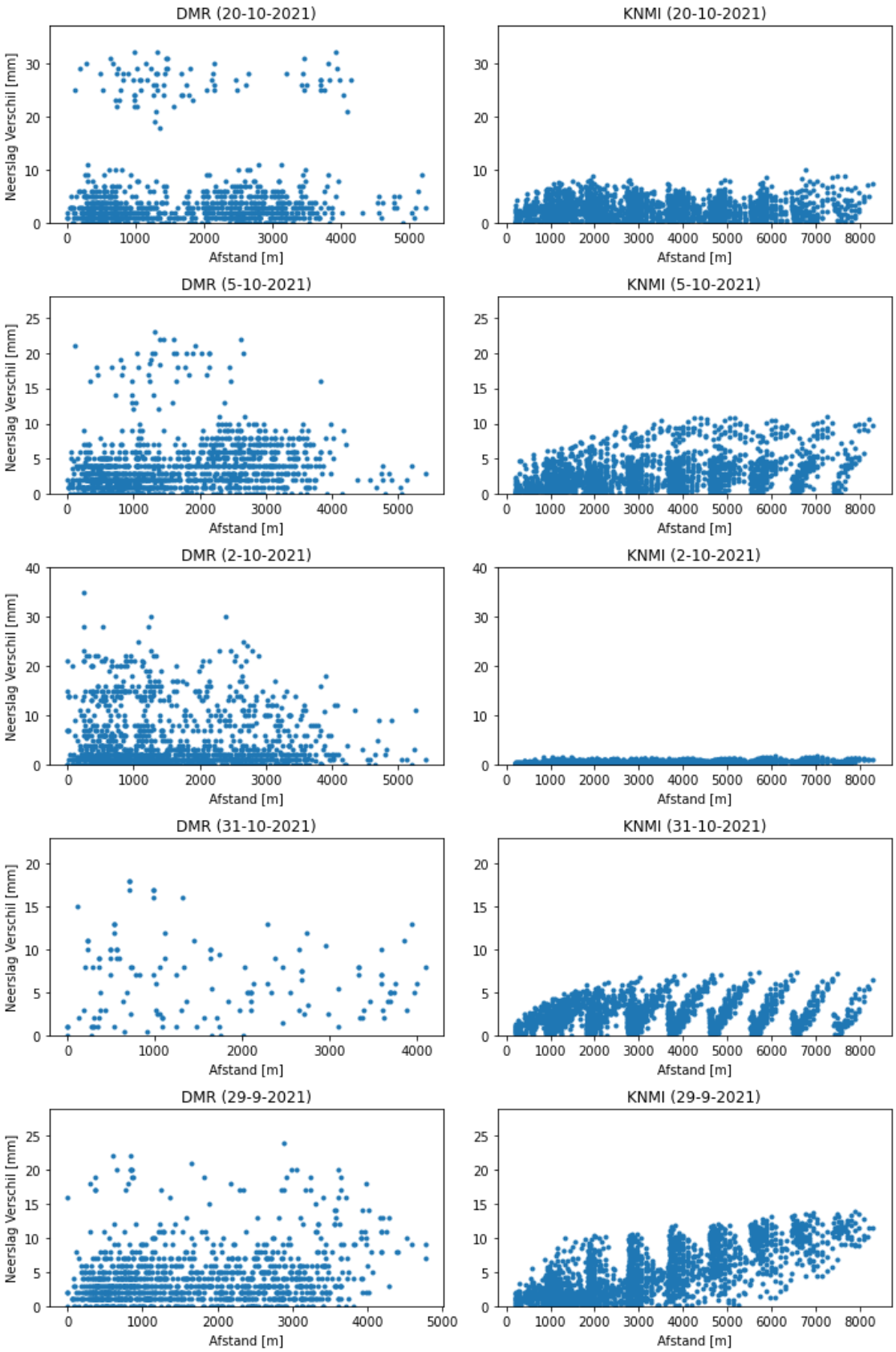
Tabel 5. Delft Meet Regen neerslagdata van geselecteerde regendagen met vergelijkingsdata tussen de Delft Meer Regen neerslagdata en KNMI radardata.



Figuur 15. Crossplot van de vijf regendagen met de hoogste gemiddelde dagelijkse neerslag.



Figuur 16. Crossplot van de vijf regendagen met de hoogste gemiddelde dagelijkse neerslag.



Figuur 17. Delft Meet Regen en KNMI meetpuntcombinaties voor de vijf dagen met de hoogste gemiddelde dagelijkse neerslag.

Bijlage 2: Python script

In [ ]:
```
indir = r"C:\Users\sonny\Documents\Bouwplaats Jaar 3\_BEP\MAIN"
```

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import geopandas as gpd
import h5py
import os
import pyproj
import seaborn as sns
```

Data Ophalen en Preparatie

In [ ]:
```
## Delft shapefile (map)

mp = gpd.read_file(
    r"C:\Users\sonny\Documents\Bouwplaats Jaar 3\_BEP\MAIN\DATA\Data1\Delft_shape.shp")
```

In [ ]:
```
def read_P_from_h5(h5_filename):
    """ This function opens the h5 files containing the KNMI radar rainfall
    estimates. It extracts the rainfall amounts and returns this as a 2D array.

    Parameters
    ----------
    h5_filename: str
      The complete filename (including directory) of the file.

    Returns
    ----------
    P_temp: array-like
        2D array containing the rainfall amounts for this file

    """
    # Open the file
    ds = h5py.File(h5_filename, 'r')
    # Get the values out of the file
    ds_rainfall_values = ds['image1']['image_data'][:]
```

```
    # The values 65535 are NoData.
    ds_rainfall = np.where(ds_rainfall_values == 65535, np.NaN, ds_rainfall_values)
    # Divide the values by 100, as they're saved as value * 100 [mm].
    P_temp = (ds_rainfall / 100.0)

    return P_temp
```

In [ ]:

```
metadata = {}

unit = "mm"
pixelsize = 1.0
accutime = 5.0
f = h5py.File(os.path.join(
    indir,
    "DATA",
    "Data1",
    "RAD_NL25_RAC_MFBS_EM_5min",
    "2021",
    "07",
    "RAD_NL25_RAC_MFBS_EM_5min_202107010000_NL.h5"), 'r')

geographic = f["geographic"]
proj4str = geographic["map_projection"].attrs["projection_proj4_params"].decode()
pr = pyproj.Proj(proj4str)
metadata["projection"] = proj4str

# Get coordinates
latlon_corners = geographic.attrs["geo_product_corners"]
ll_lat = latlon_corners[1]
ll_lon = latlon_corners[0]
ur_lat = latlon_corners[5]
ur_lon = latlon_corners[4]
lr_lat = latlon_corners[7]
lr_lon = latlon_corners[6]
ul_lat = latlon_corners[3]
ul_lon = latlon_corners[2]

ll_x, ll_y = pr(ll_lon, ll_lat)
ur_x, ur_y = pr(ur_lon, ur_lat)
lr_x, lr_y = pr(lr_lon, lr_lat)
ul_x, ul_y = pr(ul_lon, ul_lat)
x1 = min(ll_x, ul_x)
y1 = min(ll_y, lr_y)
```

```
x2 = max(lr_x, ur_x)
y2 = max(ul_y, ur_y)

# Fill in the metadata
metadata["x1"] = x1
metadata["y1"] = y1
metadata["x2"] = x2
metadata["y2"] = y2
metadata["xpixelsize"] = pixelsize
metadata["ypixelsize"] = pixelsize
metadata["cartesian_unit"] = "km"
metadata["yorigin"] = "upper"
metadata["institution"] = "KNMI - Royal Netherlands Meteorological Institute"
metadata["accutime"] = accutime
metadata["unit"] = unit
metadata["zerovalue"] = 0.0
metadata["zr_a"] = 200.0
metadata["zr_b"] = 1.6
```

In [ ]:

```
# +a en +b 1000x groter

metadata['projection'] ='+proj=stere +lat_0=90 +lon_0=0.0 +lat_ts=60.0 +a=6378137 +b=6356752 +x_0=0 +y_0=0'
metadata["x1"] = x1 *1000
metadata["y1"] = y1 *1000
metadata["x2"] = x2 *1000
metadata["y2"] = y2 *1000
metadata["xpixelsize"] = pixelsize *1000
metadata["ypixelsize"] = pixelsize *1000
metadata["cartesian_unit"] = "m"
```

In [ ]:

```
def get_RadarData(Month, Day):
    """ Read the necessary KNMI files and return the total rainfall for the given day"""

    Month = str(Month)
    Day = str(Day)

    ## Account for the format of the files
    if len(Month) < 2:
        Month = f'0{Month}'
    if len(Day) < 2:
        Day = f'0{Day}'

    # Create a list for the reference rainfall and the radar estimate
```

```
    ref_event1 = []
    C = 0

    # Read all hdf5-files and append them to a list
    for filename in sorted(os.listdir(os.path.join(indir, "DATA", "Data1", "RAD_NL25_RAC_MFBS_EM_5min", "2021", Month))):
        if filename == f'RAD_NL25_RAC_MFBS_EM_5min_2021{Month}{Day}0000_NL.h5':
            C = 1
        if C == 1:
            ref_event1.append(read_P_from_h5(os.path.join(indir,
                                                          "DATA",
                                                          "Data1",
                                                          "RAD_NL25_RAC_MFBS_EM_5min",
                                                          "2021",
                                                          Month,
                                                          filename)))

        if filename == f'RAD_NL25_RAC_MFBS_EM_5min_2021{Month}{Day}2355_NL.h5':
            break

    # Calculate the day's total rainfall
    ref_event1_sum = np.nansum(ref_event1, axis=0)

    return ref_event1_sum
```

In [ ]:

```
## Make a tranformer that will transform the KNMI projection to the DMR projection

in2_ = pyproj.crs.CRS('+proj=stere +lat_0=90 +lon_0=0.0 +lat_ts=60.0 +a=6378137 +b=6356752 +x_0=0 +y_0=0')
out2_ = pyproj.crs.CRS('+proj=longlat +datum=WGS84 +no_defs')
T2 = pyproj.Transformer.from_crs(in2_, out2_)
```

In [ ]:

```
def RadarGrid(x1, x2, y1, y2, event_sum):
    """ Make a grid in longlat of the KNMI data for Delft"""

    x_ = np.linspace(x1, x2, num=np.shape(event_sum)[1])
    y_ = np.linspace(y1, y2, num=np.shape(event_sum)[0])

    ## Store latitude, longitude and rainfall for desired area

    ri = []
    lons = []
    lats = []

    bbox = [4.31, 4.42, 51.96, 52.035]
```

```
    for h in np.arange(0, len(y_)):
        for b in np.arange(0, len(x_)):

            lon, lat = T2.transform(x_[b], y_[-(1+h)])

            if lon >= bbox[0] and lon <= bbox[1]:
                if lat >= bbox[2] and lat <= bbox[3]:

                    ri.append(event_sum[h][b])
                    lons.append(lon)
                    lats.append(lat)

    return ri, lons, lats
```

In [ ]:
```
## Read DMR file

df = pd.read_excel(r"C:\Users\sonny\Documents\Bouwplaats Jaar 3\_BEP\MAIN\DATA\Delft_meet_regen 2021_Main_AVGproof.xlsx",
                   index_col=0)
```

In [ ]:
```
## Clean up the 'Neerslag in mm' column

g = df['Neerslag in mm'].replace(to_replace='mm', value='', regex=True)
g = g.replace(to_replace='O', value='0', regex=True)
df['Neerslag in mm'] = g

## Drop the 'ObjectID' column

df = df.drop(labels=['ObjectID'], axis=1)

df['Neerslag in mm'] = pd.to_numeric(df['Neerslag in mm'], errors='coerce')
```

In [ ]:
```
## Exclude any measurements that fall outside of the Delft bounding box

bbox = [4.31, 4.42, 51.96, 52.035]

df = df[(df["x"] > bbox[0]) & (df["x"] < bbox[1])]
df = df[(df["y"] > bbox[2]) & (df["y"] < bbox[3])]
```

In [ ]:
```
## Join the columns 'Datum2' and 'Tijd'

t = df['Datum2'] + ' ' + df['Tijd']
```

```
t2 = pd.to_datetime(t, dayfirst=True, infer_datetime_format=True)

df['ABS_time'] = t2
df = df.drop(labels=['Datum2', 'Tijd'], axis=1)
```

In [ ]:
```
## Round the Dates to days

df['ABS_time'] = df['ABS_time'].round('D')
```

In [ ]:
```
## Set all dates back 1 day
## To account for the fact that the rain collected on the day, was the rainfall of the previous day

df['ABS_time'] = df['ABS_time'] - pd.Timedelta(1, unit='day')
```

In [ ]:
```
## Make a copy to reference later
DF_ref = df
```

In [ ]:
```
## Drop the'op- en aanmerkingen toevoegen' column

df1 = df
df = df.drop(labels=['op- en aanmerkingen toevoegen'], axis=1)

df1['op- en aanmerkingen toevoegen'] = df1['op- en aanmerkingen toevoegen'].fillna('nan')
```

In [ ]:
```
## Calculate the mean rainfall for each day

m = df.groupby(by='ABS_time').mean()
```

In [ ]:
```
m2 = df.groupby(by='ABS_time').count()['Neerslag in mm']
```

In [ ]:
```
m['Count'] = m2
```

In [ ]:
```
## Remove the first 'test' measurement

m = m.drop(index=m.index[0])
m2 = m2.drop(index=m2.index[0])
```

Data Selectie

In [ ]:
```
## Plot an overview of the count and measured mean rainfall for each day
## Skip the first measurement (on 2021-06-28)

plt.figure(figsize=(16,8))

ax1 = plt.subplot2grid((2, 1), (0, 0))
ax1.bar(m.index, m2)
plt.xlabel('Datum')
plt.ylabel('Aantal Metingen')
plt.title('Aantal DMR Metingen per Dag')

ax2 = plt.subplot2grid((2, 1), (1, 0))
ax2.bar(m.index, m['Neerslag in mm'])
# plt.xticks(m.index, rotation=80)

plt.xlabel('Datum')
plt.ylabel('Neerslag [mm]')
plt.title('Gemiddelde Dagelijkse Neerslag')

plt.tight_layout();
```

In [ ]:
```
## Pick the N days with the highest mean rainfall (minimal N=5)

N = 30

## Minimum of 5 days

if N < 5:
    N = 5

highN = m['Neerslag in mm'].nlargest(n=N)
highNd = m['Neerslag in mm'].nlargest(n=N).index
```

In [ ]:
```
plt.figure(figsize=(12,4))

plt.bar(m.index[1:], m['Neerslag in mm'][1:])

plt.xlabel('Datum')
plt.ylabel('Neerslag [mm]')
plt.title('Gemiddelde Dagelijkse Neerslag')

plt.tight_layout();
```

```
plt.figure(figsize=(12,4))

plt.bar(m.index[1:], m['Neerslag in mm'][1:])

for i, v in enumerate(highN.items()):
    _date = v[0]
    plt.bar(v[0], v[1], color='r')


plt.xlabel('Datum')
plt.ylabel('Neerslag [mm]')
plt.title('Gemiddelde Dagelijkse Neerslag')

plt.tight_layout();
```

In [ ]:

```
plt.figure(figsize=(12,4))

plt.bar(m.index[1:], m['Count'][1:])

plt.xlabel('Datum')
plt.ylabel('Aantal Metingen')
plt.title('Aantal DMR Metingen per Dag')

plt.tight_layout();

plt.figure(figsize=(12,4))

plt.bar(m.index[1:], m['Count'][1:])

for i, v in enumerate(highN.items()):
    _date = v[0]
    plt.bar(v[0], m['Count'][v[0]], color='r')


plt.xlabel('Datum')
plt.ylabel('Aantal Metingen')
plt.title('Aantal DMR Metingen per Dag')

plt.tight_layout();
```

In [ ]:

```
pd.DataFrame(data=highN.head(10)).rename(columns={'Neerslag in mm':'Gemiddelde Neerslag [mm]'})
```

In [ ]:
```
## Get the KNMI data from the dates with most rainfall

r = []

for i, v in enumerate(highN.items()):
    _date = v[0]
    rain = get_RadarData(_date.month, _date.day)
    r.append([_date, rain])
    print(f'set {i+1}, date: {_date.year}-{_date.month}-{_date.day}')
```

In [ ]:
```
## Get the KNMI radar grids for the chosen days

ri = []
lons = []
lats = []

for i in np.arange(0, N):

    ri_, lons_, lats_ = RadarGrid(metadata['x1'],
                                  metadata['x2'],
                                  metadata['y1'],
                                  metadata['y2'],
                                  r[i][1])

    ri.append(ri_)
    lons.append(lons_)
    lats.append(lats_)
    print(f'set {i}')
```

Ruimtelijke Plots

In [ ]:
```
## Calculate the mean, max, and min for each subset

KNMIs_means = np.zeros(N)
KNMIs_max = np.zeros(N)
KNMIs_min = np.zeros(N)
KNMIs_count = np.zeros(N)

for i in np.arange(0, len(ri)):
    KNMIs_means[i] = np.mean(ri[i])
```

```
    KNMIs_max[i] = np.max(ri[i])
    KNMIs_min[i] = np.min(ri[i])
    KNMIs_count[i] = len(ri[i])
```

In [ ]:

```
## Plot the KNMI radar measurements for the given days

fig = plt.figure(figsize=(11,8))

cmap = mpl.cm.viridis
vmin = np.mean(ri[0:5]) - 2*np.std(ri[0:5])
vmax = np.mean(ri[0:5]) + 2*np.std(ri[0:5])

plot_grid = (2, 4)

Count_label = 'Aantal Metingen'
Mean_label = 'Gemiddelde'
Max_label = 'Maximum'
Min_label = 'Minimum'

## 1

ax1 = plt.subplot2grid(plot_grid, (0, 0), rowspan=2, colspan=2)
mp.plot(ax=ax1, alpha=0.3)
g1 = plt.scatter(lons[0], lats[0], c=ri[0], cmap=cmap)
plt.clim(vmin, vmax)
plt.title(f'KNMI: {r[0][0].day}-{r[0][0].month}-{r[0][0].year}')
plt.xlabel('Lengtegraad')
plt.ylabel('Breedtegraad')

plt.figtext(0, 0.14, f'{Count_label} = {KNMIs_count[0]}')
plt.figtext(0, 0.12, f'{Mean_label} = {np.round(KNMIs_means[0], 1)} mm')
plt.figtext(0, 0.10, f'{Max_label} = {np.round(KNMIs_max[0], 1)} mm')
plt.figtext(0, 0.08, f'{Min_label} = {np.round(KNMIs_min[0], 1)} mm')

## 2

ax2 = plt.subplot2grid(plot_grid, (0, 2))
mp.plot(ax=ax2, alpha=0.3)
g2 = plt.scatter(lons[1], lats[1], c=ri[1], cmap=cmap)
plt.title(f'KNMI: {r[1][0].day}-{r[1][0].month}-{r[1][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.55, 0.54, f'{Count_label} = {KNMIs_count[1]}')
```

```
plt.figtext(0.55, 0.52, f'{Mean_label} = {np.round(KNMIs_means[1], 1)} mm')
plt.figtext(0.55, 0.50, f'{Max_label} = {np.round(KNMIs_max[1], 1)} mm')
plt.figtext(0.55, 0.48, f'{Min_label} = {np.round(KNMIs_min[1], 1)} mm')

## 3

ax3 = plt.subplot2grid(plot_grid, (0, 3))
mp.plot(ax=ax3, alpha=0.3)
g3 = plt.scatter(lons[2], lats[2], c=ri[2], cmap=cmap)
plt.yticks((),())
plt.title(f'KNMI: {r[2][0].day}-{r[2][0].month}-{r[2][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.8, 0.54, f'{Count_label} = {KNMIs_count[2]}')
plt.figtext(0.8, 0.52, f'{Mean_label} = {np.round(KNMIs_means[2], 1)} mm')
plt.figtext(0.8, 0.50, f'{Max_label} = {np.round(KNMIs_max[2], 1)} mm')
plt.figtext(0.8, 0.48, f'{Min_label} = {np.round(KNMIs_min[2], 1)} mm')

## 4

ax4 = plt.subplot2grid(plot_grid, (1, 2))
mp.plot(ax=ax4, alpha=0.3)
g4 = plt.scatter(lons[3], lats[3], c=ri[3], cmap=cmap)
plt.title(f'KNMI: {r[3][0].day}-{r[3][0].month}-{r[3][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.55, 0.04, f'{Count_label} = {KNMIs_count[3]}')
plt.figtext(0.55, 0.02, f'{Mean_label} = {np.round(KNMIs_means[3], 1)} mm')
plt.figtext(0.55, 0.00, f'{Max_label} = {np.round(KNMIs_max[3], 1)} mm')
plt.figtext(0.55, -0.02, f'{Min_label} = {np.round(KNMIs_min[3], 1)} mm')

## 5

ax5 = plt.subplot2grid(plot_grid, (1, 3))
mp.plot(ax=ax5, alpha=0.3)
g5 = plt.scatter(lons[4], lats[4], c=ri[4])
plt.yticks((),())
plt.title(f'KNMI: {r[4][0].day}-{r[4][0].month}-{r[4][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.8, 0.04, f'{Count_label} = {KNMIs_count[4]}')
plt.figtext(0.8, 0.02, f'{Mean_label} = {np.round(KNMIs_means[4], 1)} mm')
plt.figtext(0.8, 0.00, f'{Max_label} = {np.round(KNMIs_max[4], 1)} mm')
plt.figtext(0.8, -0.02, f'{Min_label} = {np.round(KNMIs_min[4], 1)} mm')
```

```
##

cbar = fig.colorbar(g1, ax=ax1, fraction=0.08)
cbar.set_label('Neerslag [mm]')

plt.tight_layout();
```

In [ ]:
```
## Get the data subset for each day from the DMR data (for the highest N)

subs = []

for i in np.arange(0, N):
    subs.append(df1[df1["ABS_time"] == r[i][0]].dropna(axis=0))
```

In [ ]:
```
## Adjust outcome by using removing all outliers(0), all outliers with comments(1), or include the outliers(2)
outliers = 0

## Calculate the mean, max, and min for each subset

subs_means = np.zeros(N)
subs_max = np.zeros(N)
subs_min = np.zeros(N)
subs_count = np.zeros(N)

for i in np.arange(0, len(subs)):

    subs_means[i] = subs[i]['Neerslag in mm'].mean()
    subs_max[i] = subs[i]['Neerslag in mm'].max()

    ### Remove Outliers

    ## Outliers with comments
    if outliers == 1:
        CON1 = subs_means[i] + 3*np.std(subs[i]['Neerslag in mm'])
        CON2 = (subs[i]['op- en aanmerkingen toevoegen'] != 'nan')
        subs[i] = subs[i].drop(subs[i][(subs[i]['Neerslag in mm'] >= (CON1)) & (CON2)].index)
        subs_means[i] = subs[i]['Neerslag in mm'].mean()
        subs_max[i] = subs[i]['Neerslag in mm'].max()

    ## Outliers
    if outliers == 0:
        subs[i] = subs[i].drop(subs[i][subs[i]['Neerslag in mm'] >= (subs_means[i]+3*np.std(subs[i]['Neerslag in mm']))].index)
```

```
        subs_means[i] = subs[i]['Neerslag in mm'].mean()
        subs_max[i] = subs[i]['Neerslag in mm'].max()

        subs_min[i] = subs[i]['Neerslag in mm'].min()

    subs_count[i] = subs[i]['Neerslag in mm'].count()
```

In [ ]:

```
## Plot the DMR measurements for the chosen days

fig = plt.figure(figsize=(11,8))

plot_grid = (2, 4)
cmap = mpl.cm.viridis
vmin = np.mean(subs_means[0:5]) - 2*np.std(subs_means[0:5])
vmax = np.mean(subs_means[0:5]) + 2*np.std(subs_means[0:5])

Count_label = 'Aantal Metingen'
Mean_label = 'Gemiddelde'
Max_label = 'Maximum'
Min_label = 'Minimum'

### 1

ax1 = plt.subplot2grid(plot_grid, (0, 0), rowspan=2, colspan=2)
mp.plot(ax=ax1, alpha=0.3)
g1 = plt.scatter(subs[0]['x'], subs[0]['y'], c=subs[0]['Neerslag in mm'], cmap=cmap)

plt.title(f'DMR: {r[0][0].day}-{r[0][0].month}-{r[0][0].year}')
plt.clim(vmin, vmax)
plt.xlabel('Lengtegraad')
plt.ylabel('Breedtegraad')

plt.figtext(0, 0.14, f'{Count_label} = {subs_count[0]}')
plt.figtext(0, 0.12, f'{Mean_label} = {np.round(subs_means[0], 1)} mm')
plt.figtext(0, 0.10, f'{Max_label} = {subs_max[0]} mm')
plt.figtext(0, 0.08, f'{Min_label} = {subs_min[0]} mm')

### 2

ax2 = plt.subplot2grid(plot_grid, (0, 2))
mp.plot(ax=ax2, alpha=0.3)
g2 = plt.scatter(subs[1]['x'], subs[1]['y'], c=subs[1]['Neerslag in mm'], cmap=cmap)
```

```
plt.title(f'DMR: {r[1][0].day}-{r[1][0].month}-{r[1][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.55, 0.54, f'{Count_label} = {subs_count[1]}')
plt.figtext(0.55, 0.52, f'{Mean_label} = {np.round(subs_means[1], 1)} mm')
plt.figtext(0.55, 0.50, f'{Max_label} = {subs_max[1]} mm')
plt.figtext(0.55, 0.48, f'{Min_label} = {subs_min[1]} mm')

### 3

ax3 = plt.subplot2grid(plot_grid, (0, 3))
mp.plot(ax=ax3, alpha=0.3)
g3 = plt.scatter(subs[2]['x'], subs[2]['y'], c=subs[2]['Neerslag in mm'], cmap=cmap)

plt.yticks((),())
plt.title(f'DMR: {r[2][0].day}-{r[2][0].month}-{r[2][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.8, 0.54, f'{Count_label} = {subs_count[2]}')
plt.figtext(0.8, 0.52, f'{Mean_label} = {np.round(subs_means[2], 1)} mm')
plt.figtext(0.8, 0.50, f'{Max_label} = {subs_max[2]} mm')
plt.figtext(0.8, 0.48, f'{Min_label} = {subs_min[2]} mm')

### 4

ax4 = plt.subplot2grid(plot_grid, (1, 2))
mp.plot(ax=ax4, alpha=0.3)
g4 = plt.scatter(subs[3]['x'], subs[3]['y'], c=subs[3]['Neerslag in mm'], cmap=cmap)

plt.title(f'DMR: {r[3][0].day}-{r[3][0].month}-{r[3][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.55, 0.04, f'{Count_label} = {subs_count[3]}')
plt.figtext(0.55, 0.02, f'{Mean_label} = {np.round(subs_means[3], 1)} mm')
plt.figtext(0.55, 0.00, f'{Max_label} = {subs_max[3]} mm')
plt.figtext(0.55, -0.02, f'{Min_label} = {subs_min[3]} mm')

### 5

ax5 = plt.subplot2grid(plot_grid, (1, 3))
mp.plot(ax=ax5, alpha=0.3)
g5 = plt.scatter(subs[4]['x'], subs[4]['y'], c=subs[4]['Neerslag in mm'], cmap=cmap)

plt.yticks((),())
```

```
plt.title(f'DMR: {r[4][0].day}-{r[4][0].month}-{r[4][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.8, 0.04, f'{Count_label} = {subs_count[4]}')
plt.figtext(0.8, 0.02, f'{Mean_label} = {np.round(subs_means[4], 1)} mm')
plt.figtext(0.8, 0.00, f'{Max_label} = {subs_max[4]} mm')
plt.figtext(0.8, -0.02, f'{Min_label} = {subs_min[4]} mm')

###

cbar = fig.colorbar(g1, ax=ax1, fraction=0.08)
cbar.set_label('Neerslag [mm]')

plt.tight_layout();
```

In [ ]:

```
def LonLat_Distance(lon1, lat1, lon2, lat2):
    """This function calculates the distance between two coordinates in longitude and latitude"""

    R = 6371000

    dlon = np.deg2rad(lon1-lon2)
    dlat = np.deg2rad(lat1-lat2)

    a = ((np.sin(dlat/2))**2) + np.cos(lat1) *np.cos(lat2)*((np.sin(dlon/2))**2)
    c = 2 * np.arctan2((a**0.5), ((1-a)**0.5))
    d = R * c

    return d
```

In [ ]:

```
def Difference(sub_rain, sub_x, sub_y, ri, lons, lats):
    """This function calculates the difference between a given DMR measurement and the nearest KNMI radar measurement"""
    """The parameters for this function all need to be data of the same day"""

    dis = np.zeros(len(sub_x))
    diff = np.zeros(len(sub_x))

    for i in np.arange(0, len(sub_x)):
        d = 100000
        v = 0
        for n in np.arange(0, len(lons)):
            d_c = LonLat_Distance(sub_x[i], sub_y[i], lons[n], lats[n])
            if d_c < d:
                d = d_c
```

```
                v = sub_rain[i] - ri[n]

        dis[i] = d
        diff[i] = v

    return diff, dis
```

In [ ]:
```
## Calculate the difference of all DMR measurements and their nearest KNMI radar measument

diffs = []
diss = []

for i in np.arange(0, N):
    diffi, disi = Difference(subs[i]['Neerslag in mm'], subs[i]['x'], subs[i]['y'], ri[i], lons[i], lats[i])
    subs[i]['Diff'] = diffi
    subs[i]['Dis'] = disi
    diffs.append(diffi)
    diss.append(disi)

## Calculate the mean, max, and min differences for all subsets

diffs_mean = np.zeros(N)
diffs_max = np.zeros(N)
diffs_min = np.zeros(N)

for i in np.arange(0, N):
    diffs_mean[i] = diffs[i].mean()
    diffs_max[i] = np.abs(diffs[i]).max()
    diffs_min[i] = np.abs(diffs[i]).min()
```

In [ ]:
```
## Plot the calculated difference found for each DMR measurement

fig = plt.figure(figsize=(11,8))

plot_grid = (2, 4)
cmap = mpl.cm.Reds
vmin = np.mean(diffs_mean[0:5]) - np.std(diffs_mean[0:5])
vmax = np.mean(diffs_mean[0:5]) + np.std(diffs_mean[0:5])

Count_label = 'Aantal DMR Metingen'
MeanD_label = 'Gemiddeld Verschil'
MaxD_label = 'Maximum Verschil'
MinD_label = 'Minimum Verschil'
```

```
### 1

ax1 = plt.subplot2grid(plot_grid, (0, 0), rowspan=2, colspan=2)
mp.plot(ax=ax1, alpha=0.3)
g1 = plt.scatter(subs[0]['x'], subs[0]['y'], c=diffs[0], cmap=cmap)
plt.title(f'Neerslag Verschil: {r[0][0].day}-{r[0][0].month}-{r[0][0].year}')
plt.xlabel('Lengtegraad')
plt.ylabel('Breedtegraad')
plt.clim(vmin, vmax)

plt.figtext(0, 0.14, f'{Count_label} = {subs_count[0]}')
plt.figtext(0, 0.12, f'{MeanD_label} = {np.round(diffs_mean[0], 1)} mm')
plt.figtext(0, 0.10, f'{MaxD_label} = {np.round(diffs_max[0], 1)} mm')
plt.figtext(0, 0.08, f'{MinD_label} = {np.round(diffs_min[0], 1)} mm')

### 2

ax2 = plt.subplot2grid(plot_grid, (0, 2))
mp.plot(ax=ax2, alpha=0.3)
g2 = plt.scatter(subs[1]['x'], subs[1]['y'], c=diffs[1], cmap=cmap)
plt.title(f'Neerslag Verschil: {r[1][0].day}-{r[1][0].month}-{r[1][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.55, 0.54, f'{Count_label} = {subs_count[1]}')
plt.figtext(0.55, 0.52, f'{MeanD_label} = {np.round(diffs_mean[1], 1)} mm')
plt.figtext(0.55, 0.50, f'{MaxD_label} = {np.round(diffs_max[1], 1)} mm')
plt.figtext(0.55, 0.48, f'{MinD_label} = {np.round(diffs_min[1], 1)} mm')

### 3

ax3 = plt.subplot2grid(plot_grid, (0, 3))
mp.plot(ax=ax3, alpha=0.3)
g3 = plt.scatter(subs[2]['x'], subs[2]['y'], c=diffs[2], cmap=cmap)
plt.yticks((),())
plt.title(f'Neerslag Verschil: {r[2][0].day}-{r[2][0].month}-{r[2][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.8, 0.54, f'{Count_label} = {subs_count[2]}')
plt.figtext(0.8, 0.52, f'{MeanD_label} = {np.round(diffs_mean[2], 1)} mm')
plt.figtext(0.8, 0.50, f'{MaxD_label} = {np.round(diffs_max[2], 1)} mm')
plt.figtext(0.8, 0.48, f'{MinD_label} = {np.round(diffs_min[2], 1)} mm')

### 4
```

```
ax4 = plt.subplot2grid(plot_grid, (1, 2))
mp.plot(ax=ax4, alpha=0.3)
g4 = plt.scatter(subs[3]['x'], subs[3]['y'], c=diffs[3], cmap=cmap)
plt.title(f'Neerslag Verschil: {r[3][0].day}-{r[3][0].month}-{r[3][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.55, 0.04, f'{Count_label} = {subs_count[3]}')
plt.figtext(0.55, 0.02, f'{MeanD_label} = {np.round(diffs_mean[3], 1)} mm')
plt.figtext(0.55, 0.00, f'{MaxD_label} = {np.round(diffs_max[3], 1)} mm')
plt.figtext(0.55, -0.02, f'{MinD_label} = {np.round(diffs_min[3], 1)} mm')

### 5

ax5 = plt.subplot2grid(plot_grid, (1, 3))
mp.plot(ax=ax5, alpha=0.3)
g5 = plt.scatter(subs[4]['x'], subs[4]['y'], c=diffs[4], cmap=cmap)
plt.yticks((),())
plt.title(f'Neerslag Verschil: {r[4][0].day}-{r[4][0].month}-{r[4][0].year}')
plt.clim(vmin, vmax)

plt.figtext(0.8, 0.04, f'{Count_label} = {subs_count[4]}')
plt.figtext(0.8, 0.02, f'{MeanD_label} = {np.round(diffs_mean[4], 1)} mm')
plt.figtext(0.8, 0.00, f'{MaxD_label} = {np.round(diffs_max[4], 1)} mm')
plt.figtext(0.8, -0.02, f'{MinD_label} = {np.round(diffs_min[4], 1)} mm')

###

cbar = fig.colorbar(g1, ax=ax1, fraction=0.08)
cbar.set_label('Neerslag [mm]')

plt.tight_layout();
```

In [ ]:

```
## Plot Comparisons

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

plot_grid = (1, 4)

frac = 0.05

cmap1 = mpl.cm.viridis
vmin1 = np.mean(ri[0:5]) - 2*np.std(ri[0:5])
```

```
vmax1 = np.mean(ri[0:5]) + 2*np.std(ri[0:5])

cmap2 = mpl.cm.viridis
vmin2 = np.mean(subs_means[0:5]) - 2*np.std(subs_means[0:5])
vmax2 = np.mean(subs_means[0:5]) + 2*np.std(subs_means[0:5])

cmap3 = mpl.cm.Reds
vmin3 = max(np.mean(diffs_mean[0:5] - np.std(diffs_mean[0:5])), 0)
vmax3 = np.mean(diffs_mean[0:5]) + np.std(diffs_mean[0:5])

## KNMI

ax1 = plt.subplot2grid(plot_grid, (0, 0), rowspan=1, colspan=1)
mp.plot(ax=ax1, alpha=0.3)
g1 = plt.scatter(lons[0], lats[0], c=ri[0], cmap=cmap1)

plt.clim(vmin2, vmax2)
plt.title(f'KNMI ({r[0][0].day}-{r[0][0].month}-{r[0][0].year})')
plt.xlabel('Lengtegraad')
plt.ylabel('Breedtegraad')

cbar = fig.colorbar(g1, ax=ax1, fraction=frac)
cbar.set_label('Neerslag [mm]')

## DMR

ax2 = plt.subplot2grid(plot_grid, (0, 1), rowspan=1, colspan=1)
mp.plot(ax=ax2, alpha=0.3)
g2 = plt.scatter(subs[0]['x'], subs[0]['y'], c=subs[0]['Neerslag in mm'], cmap=cmap2)

plt.title(f'DMR ({r[0][0].day}-{r[0][0].month}-{r[0][0].year})')
plt.clim(vmin2, vmax2)
plt.xlabel('Lengtegraad')
plt.yticks((),())

cbar = fig.colorbar(g2, ax=ax2, fraction=frac)
cbar.set_label('Neerslag [mm]')

## DIFF

ax3 = plt.subplot2grid(plot_grid, (0, 2), rowspan=1, colspan=1)
mp.plot(ax=ax3, alpha=0.3)
g3 = plt.scatter(subs[0]['x'], subs[0]['y'], c=diffs[0], cmap=cmap3)
plt.title(f'Verschil ({r[0][0].day}-{r[0][0].month}-{r[0][0].year})')
```

```
plt.xlabel('Lengtegraad')
plt.clim(vmin3, vmax3)
plt.yticks((),())

cbar = fig.colorbar(g3, ax=ax3, fraction=frac)
cbar.set_label('Neerslag Verschil [mm]')

###

print(f'{r[0][0].day}-{r[0][0].month}-{r[0][0].year}')

plt.tight_layout();
```

Data Plots

In [ ]:

```
## Collect the subset calculations in a single dataframe

data = [subs_means,
        subs_max, subs_min,
        diffs_mean,
        diffs_max,
        diffs_min,
        subs_count]

index = ['Mean',
         'Max',
         'Min',
         'Difference_Mean',
         'Difference_Max',
         'Difference_Min',
         'Count']

columns = []

for i, t in enumerate(r):
    columns.append(t[0])

C = pd.DataFrame(data=data, index=index, columns=columns).transpose()

C = C.rename(
    columns={'Mean':'Gemiddelde [mm]',
```

```
            'Max': 'Maximum [mm]',
             'Min': 'Minimum [mm]',
            'Difference_Mean':'Gemiddeld Verschil [mm]',
            'Difference_Max':'Maximum Verschil [mm]',
            'Difference_Min':'Minimum Verschil [mm]',
            'Count':'Aantal DMR Metingen'})
C.head(30)
```

In [ ]:
```
x_vars = ['Aantal DMR Metingen','Gemiddelde [mm]', 'Gemiddeld Verschil [mm]', 'Maximum Verschil [mm]', 'Minimum Verschil [mm]']
y_vars = ['Aantal DMR Metingen']

d = sns.PairGrid(C, x_vars=x_vars, y_vars=y_vars)
d.map_diag(sns.histplot)
d.map_offdiag(sns.scatterplot)
d.add_legend();
```

In [ ]:
```
TOT3 = pd.concat(subs[0:3])
TOT5 = pd.concat(subs[0:5])
TOT10 = pd.concat(subs[0:10])
```

In [ ]:
```
TOT3 = TOT3.reset_index()
TOT3 = TOT3.drop(labels=['ParentGlobalID','GlobalID'], axis=1).rename(
    columns={'Neerslag in mm':'Neerslag [mm]', 'Diff':'Neerslag Verschil [mm]', 'Dis':'Afstand [m]'})

TOT5 = TOT5.reset_index()
TOT5 = TOT5.drop(labels=['ParentGlobalID','GlobalID'], axis=1).rename(
    columns={'Neerslag in mm':'Neerslag [mm]', 'Diff':'Neerslag Verschil [mm]', 'Dis':'Afstand [m]'})

TOT10 = TOT10.reset_index()
TOT10 = TOT10.drop(labels=['ParentGlobalID','GlobalID'], axis=1).rename(
    columns={'Neerslag in mm':'Neerslag [mm]', 'Diff':'Neerslag Verschil [mm]', 'Dis':'Afstand [m]'})
```

In [ ]:
```
TOT3['Datum'] = TOT3['ABS_time'].dt.strftime('%Y-%m-%d')
TOT5['Datum'] = TOT5['ABS_time'].dt.strftime('%Y-%m-%d')
TOT10['Datum'] = TOT10['ABS_time'].dt.strftime('%Y-%m-%d')
```

In [ ]:
```
g3 = sns.pairplot(TOT3, hue='Datum', vars=['Neerslag [mm]', 'Neerslag Verschil [mm]', 'Afstand [m]'])
g5 = sns.pairplot(TOT5, hue='Datum', vars=['Neerslag [mm]', 'Neerslag Verschil [mm]', 'Afstand [m]'])
g10 = sns.pairplot(TOT10, hue='Datum', vars=['Neerslag [mm]', 'Neerslag Verschil [mm]', 'Afstand [m]'])
```

Correlatie

In [ ]:
```
def Subs_cor_(sub, kind='DMR'):

    dists = []
    v1 = []
    v2 = []
    point1 = []
    point2 = []
    comb = []

    sds = sub

    for e, u in enumerate(sub['x']):
        for i, v in enumerate(sds['x']):

            if i > 0:
                d = LonLat_Distance(sds['x'][0], sds['y'][0], sds['x'][i], sds['y'][i])

                dists.append(d)
                v1.append(sds['Neerslag in mm'][0])
                v2.append(sds['Neerslag in mm'][i])

                if kind == 'DMR':
                    point1.append(sds['Meetpunt'][0])
                    point2.append(sds['Meetpunt'][i])
                if kind == 'KNMI':
                    comb.append(f'{e}, {i}')

        sds = sds[1::]
        if kind == 'KNMI':
            sds.reset_index(drop=True, inplace=True)

    if kind == 'KNMI':
        return dists, v1, v2, comb
    if kind == 'DMR':
        return dists, v1, v2, point1, point2
```

In [ ]:
```
def Subs_cor(sub, kind='DMR'):

    dists = []
    differs = []
```

```
    point1 = []
    point2 = []
    comb = []

    sds = sub

    for e, u in enumerate(sub['x']):
        for i, v in enumerate(sds['x']):

            if i > 0:
                d = LonLat_Distance(sds['x'][0], sds['y'][0], sds['x'][i], sds['y'][i])

                rd = abs(sds['Neerslag in mm'][0] - sds['Neerslag in mm'][i])

                dists.append(d)
                differs.append(rd)

                if kind == 'DMR':
                    point1.append(sds['Meetpunt'][0])
                    point2.append(sds['Meetpunt'][i])
                if kind == 'KNMI':
                    comb.append(f'{e}, {i}')

        sds = sds[1::]
        if kind == 'KNMI':
            sds.reset_index(drop=True, inplace=True)

    if kind == 'KNMI':
        return dists, differs, comb
    if kind == 'DMR':
        return dists, differs, point1, point2
```

In [ ]:

```
KNMI_subs = []

for i in np.arange(0, len(ri)):

    sub = pd.DataFrame(data={'Neerslag in mm':ri[i], 'x':lons[i], 'y':lats[i]})
    KNMI_subs.append(sub)
```

In [ ]:

```
fig = plt.figure(figsize=(10, 15))

plot_grid = (5, 2)
```

```
for i in np.arange(0, 5):

    ax11 = plt.subplot2grid(plot_grid, (i, 0), rowspan=1, colspan=1)
    DMR_s1, DMR_d1, p1, p2 = Subs_cor(subs[i])
    g11 = plt.scatter(DMR_s1, DMR_d1, marker='.')
    plt.title(f'DMR ({r[i][0].day}-{r[i][0].month}-{r[i][0].year})')
    plt.ylim(0, (np.max(DMR_d1)+5))
    plt.xlabel('Afstand [m]')
    plt.ylabel('Neerslag Verschil [mm]')

    ax12 = plt.subplot2grid(plot_grid, (i, 1), rowspan=1, colspan=1)
    KNMI_s1, KNMI_d1, KNMI_comb = Subs_cor(KNMI_subs[i], kind='KNMI')
    g12 = plt.scatter(KNMI_s1, KNMI_d1, marker='.')
    plt.title(f'KNMI ({r[i][0].day}-{r[i][0].month}-{r[i][0].year})')
    plt.ylim(0, (np.max(DMR_d1)+5))
    plt.xlabel('Afstand [m]')

plt.tight_layout()
```

In [ ]:
```
### Highest measurements for event0
High_M = subs[0].sort_values('Neerslag in mm', ascending=False).head(5)['Meetpunt']
```

In [ ]:
```
columns=['Distance', 'Difference', 'P1', 'P2']
DMR_s1, DMR_d1, p1, p2 = Subs_cor(subs[0])
DMR1 = pd.DataFrame(data=[DMR_s1, DMR_d1, p1, p2], index=columns).transpose()
DMR1['Ratio'] = pd.to_numeric(DMR1['Difference']/DMR1['Distance'])
DMR1['Distance'] = pd.to_numeric(DMR1['Distance'])
DMR1['Difference'] = pd.to_numeric(DMR1['Difference'])

DMR1_0 = DMR1.sort_values('Ratio', ascending=False)
```

In [ ]:
```
## All pairings with the highest measurement
DMR1_2 = pd.concat([DMR1_0[DMR1_0['P1'] == High_M[0]], DMR1_0[DMR1_0['P2'] == High_M[0]]])
DMR1_2.head(5)
```

In [ ]:
```
DMR_s1, DMR_d1, p1, p2 = Subs_cor(subs[0])
plt.scatter(DMR_s1, DMR_d1, marker='.', alpha=0.5)
plt.title(f'DMR ({r[0][0].day}-{r[0][0].month}-{r[0][0].year})')
plt.ylabel('Neerslag Verschil [mm]')
plt.xlabel('Afstand [mm]')

plt.plot(DMR1_2['Distance'], DMR1_2['Difference'], 'ro', fillstyle='none');
```

In [ ]:
```
### Highest Measured Rainfall

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

plot_grid = (1, 2)

frac = 0.05

cmap2 = mpl.cm.viridis
vmin2 = np.mean(subs_means[0:5]) - 2*np.std(subs_means[0:5])
vmax2 = np.mean(subs_means[0:5]) + 2*np.std(subs_means[0:5])

ax2 = plt.subplot2grid(plot_grid, (0, 0), rowspan=1, colspan=1)
mp.plot(ax=ax2, alpha=0.3)
g2 = plt.scatter(subs[0]['x'], subs[0]['y'], c=subs[0]['Neerslag in mm'], cmap=cmap2)

Mmax = subs[0][subs[0]['Neerslag in mm'] == np.max(subs[0]['Neerslag in mm'])]

g3 = plt.plot(
    Mmax['x'].head(1), Mmax['y'].head(1), 'ro', fillstyle='none', markersize=13, label=Mmax['Meetpunt'].head(1).item())

plt.title(f'DMR ({r[0][0].day}-{r[0][0].month}-{r[0][0].year})')
plt.clim(vmin2, vmax2)
plt.xlabel('Lengtegraad')
plt.ylabel('Breedtegraad')
plt.legend()

cbar = fig.colorbar(g2, ax=ax2, fraction=frac)
cbar.set_label('Neerslag [mm]')
```

In [ ]:
```
Mmax
```

In [ ]:
```
## Take a pairing with a relatively short distance and large difference

DMR1_ = DMR1_0[(DMR1_0['Ratio'] < 1)]
DMR1_ = DMR1_[(DMR1_['Distance'] < 150) & (DMR1_['Distance'] > 5)]
DMR1_ = DMR1_[(DMR1_['Difference'] > 5)]
DMR1_ = DMR1_.sort_values('Difference', ascending=False).head(20)
DMR1_
```

In [ ]:
```
P = subs[0][(subs[0]['Meetpunt'].isin([DMR1_['P1'].head(1).item(), DMR1_['P2'].head(1).item()]))]
P
```

In [ ]:
```
fig = plt.figure(figsize=(15,8))

plot_grid = (1, 2)

frac = 0.05

cmap2 = mpl.cm.viridis
vmin2 = np.mean(subs_means[0:5]) - 2*np.std(subs_means[0:5])
vmax2 = np.mean(subs_means[0:5]) + 2*np.std(subs_means[0:5])

ax2 = plt.subplot2grid(plot_grid, (0, 0), rowspan=1, colspan=1)
mp.plot(ax=ax2, alpha=0.3)
g2 = plt.scatter(subs[0]['x'], subs[0]['y'], c=subs[0]['Neerslag in mm'], cmap=cmap2)

plt.title(f'DMR ({r[0][0].day}-{r[0][0].month}-{r[0][0].year})')
plt.clim(vmin2, vmax2)
plt.ylabel('Breedtegraad')
plt.xlabel('Lengtegraad')

cbar = fig.colorbar(g2, ax=ax2, fraction=frac)
cbar.set_label('Neerslag [mm]')

###

ax3 = plt.subplot2grid(plot_grid, (0, 1), rowspan=1, colspan=1)
mp.plot(ax=ax3, alpha=0.3)
g3 = plt.scatter(P['x'], P['y'], c=P['Neerslag in mm'], cmap=cmap2)

plt.plot(P['x'][0], P['y'][0], 'ro', markersize=14, fillstyle='none', label=P['Meetpunt'][0])
plt.plot(P['x'][1], P['y'][1], 'bo', markersize=14, fillstyle='none', label=P['Meetpunt'][1])

plt.title(f'DMR ({r[0][0].day}-{r[0][0].month}-{r[0][0].year})')
plt.clim(vmin2, vmax2)
plt.xlabel('Lengtegraad')
plt.yticks((),())
plt.legend()

cbar = fig.colorbar(g2, ax=ax3, fraction=frac)
cbar.set_label('Neerslag [mm]')
```

In [ ]:
```
### DMR correlatie data
```

```
S = []
V_1 = []
V_2 = []
P_1 = []
P_2 = []

for i, v in enumerate(subs):
    s_, v_1, v_2, p_1, p_2 = Subs_cor_(v, kind='DMR')

    for e, u in enumerate(s_):
        S.append(s_[e])
        V_1.append(v_1[e])
        V_2.append(v_2[e])
        P_1.append(p_1[e])
        P_2.append(p_2[e])
```

In [ ]:

```
### KNMI correlatie data

S_KNMI = []
V_1_ = []
V_2_ = []
C_KNMI = []

for i, v in enumerate(KNMI_subs):
    s_, v_1_, v_2_, c_ = Subs_cor_(v, kind='KNMI')

    for e, u in enumerate(s_):
        S_KNMI.append(s_[e])
        V_1_.append(v_1_[e])
        V_2_.append(v_2_[e])
        C_KNMI.append(c_[e])
```

In [ ]:

```
CORR = pd.DataFrame(data=[S, V_1, V_2], index=['Distance', 'V1', 'V2']).transpose()
CORR_KNMI = pd.DataFrame(data=[S_KNMI, V_1_, V_2_], index=['Distance', 'V1', 'V2']).transpose()
```

In [ ]:

```
CORR['Distance'] = pd.to_numeric(CORR['Distance'])
CORR['V1'] = pd.to_numeric(CORR['V1'])
CORR['V2'] = pd.to_numeric(CORR['V2'])

CORR_KNMI['Distance'] = pd.to_numeric(CORR_KNMI['Distance'])
CORR_KNMI['V1'] = pd.to_numeric(CORR_KNMI['V1'])
CORR_KNMI['V2'] = pd.to_numeric(CORR_KNMI['V2'])
```

In [ ]:
```
### Divide data into sets that fall into certain intervals

interval_length = 10

CORR_DMR_intervals = []
CORR_KNMI_intervals = []

for i in np.arange(0, 6000, interval_length):

    I_DMR = CORR[(CORR['Distance'] <= (i+interval_length/2)) & (CORR['Distance'] > (i-interval_length/2))]
    I_KNMI = CORR_KNMI[(CORR_KNMI['Distance'] <= (i+interval_length/2)) & (CORR_KNMI['Distance'] > (i-interval_length/2))]

    CORR_DMR_intervals.append(I_DMR)
    CORR_KNMI_intervals.append(I_KNMI)
```

In [ ]:
```
# For all intervalls, find the (pearson) correlation coefficient and average distance of the measurements

CORR_DMR_mDistance = []
CORR_DMR_coef = []

CORR_KNMI_mDistance = []
CORR_KNMI_coef = []

for i, v in enumerate(CORR_DMR_intervals):

    CORR_DMR_mDistance.append(CORR_DMR_intervals[i]['Distance'].mean())
    CORR_DMR_coef.append(
        CORR_DMR_intervals[i].drop(labels='Distance', axis=1).corr(method='pearson', min_periods=10)['V2']['V1'])

    CORR_KNMI_mDistance.append(CORR_KNMI_intervals[i]['Distance'].mean())
    CORR_KNMI_coef.append(
        CORR_KNMI_intervals[i].drop(labels='Distance', axis=1).corr(method='pearson', min_periods=10)['V2']['V1'])
```

In [ ]:
```
DMR_CORR = pd.DataFrame(data=[CORR_DMR_mDistance, CORR_DMR_coef], index=['Distance', 'Coefficient']).transpose().dropna()
KNMI_CORR = pd.DataFrame(data=[CORR_KNMI_mDistance, CORR_KNMI_coef], index=['Distance', 'Coefficient']).transpose().dropna()
```

In [ ]:
```
## Fit a curve with the data

z_DMR = np.polyfit(DMR_CORR['Distance'], DMR_CORR['Coefficient'], 3)
pft_DMR = np.poly1d(z_DMR)

z_KNMI = np.polyfit(KNMI_CORR['Distance'], KNMI_CORR['Coefficient'], 3)
```

```
pft_KNMI = np.poly1d(z_KNMI)

xp = np.linspace(0, 5000, 100)
```

In [ ]:
```
fig = plt.figure(figsize=(15, 5))

plot_grid = (1, 2)

## DMR
plt.subplot2grid(plot_grid, (0, 0))
plt.scatter(CORR_DMR_mDistance, CORR_DMR_coef)
plt.plot(xp, pft_DMR(xp), 'r')
plt.ylim(0.5, 1)
plt.title('DMR Correlatie')
plt.xlabel('Afstand [m]')
plt.ylabel('Pearson coefficiënt [-]')

## KNMI
plt.subplot2grid(plot_grid, (0, 1))
plt.scatter(CORR_KNMI_mDistance, CORR_KNMI_coef)
plt.plot(xp, pft_KNMI(xp), 'r')
plt.ylim(0.5, 1)
plt.title('KNMI Correlatie')
plt.xlabel('Afstand [m]');
```

In [ ]:
```
plt.plot(xp, pft_KNMI(xp), 'b', label='KNMI')
plt.plot(xp, pft_DMR(xp), 'g', label='DMR')
plt.scatter(CORR_KNMI_mDistance, CORR_KNMI_coef, color='b', alpha=0.1)
plt.scatter(CORR_DMR_mDistance, CORR_DMR_coef, color='g', alpha=0.1)


plt.title('Correlatie Vergelijking')
plt.xlabel('Afstand [m]')
plt.ylabel('Pearson coefficiënt [-]')
plt.legend();
```

Standaard

In [ ]:
```
fig = plt.figure(figsize=(15,8))
```

```
plot_grid = (1, 4)

frac = 0.05

cmap1 = mpl.cm.viridis
vmin1 = np.mean(subs_means[0:5]) - 2*np.std(subs_means[0:5])
vmax1 = np.mean(subs_means[0:5]) + 2*np.std(subs_means[0:5])

ax1 = plt.subplot2grid(plot_grid, (0, 0), rowspan=1, colspan=1)
mp.plot(ax=ax1, alpha=0.3)
g2 = plt.scatter(subs[0]['x'], subs[0]['y'], c=subs[0]['Neerslag in mm'], cmap=cmap1)

plt.clim(vmin1, vmax1)
plt.title(f'DMR ({r[0][0].day}-{r[0][0].month}-{r[0][0].year})')
plt.ylabel('Breedtegraad')
plt.xlabel('Lengtegraad')

cbar = fig.colorbar(g2, ax=ax1, fraction=frac)
cbar.set_label('Neerslag [mm]')

ax2 = plt.subplot2grid(plot_grid, (0, 1), rowspan=1, colspan=1)
mp.plot(ax=ax2, alpha=0.3)

d_ = subs[0]['Neerslag in mm']

g2 = plt.scatter(subs[0]['x'], subs[0]['y'], c=((d_ - d_.mean())/(d_.std())), cmap='bwr', vmin=(-d_.std()/2), vmax=(d_.std()/2))

plt.title(f'DMR Standaard ({r[0][0].day}-{r[0][0].month}-{r[0][0].year})')
plt.xlabel('Lengtegraad')
plt.yticks((),())

cbar = fig.colorbar(g2, ax=ax2, fraction=frac)
cbar.set_label('Standaardwaarde [-]')
plt.tight_layout();
```

In [ ]:

```
## Normalize (Standardize) the measured DMR rainfall for each subset

k = []

for i, v in enumerate(subs):

    d = subs[i].groupby(by=['Meetpunt']).mean()['Neerslag in mm']
    dt = (d - d.mean())/(d.std())
```

```
    k.append(dt)
```

In [ ]:
```
## normalized (standardized) rainfals (over all events)

U = pd.DataFrame(data=k).transpose()
U = U.dropna(how= 'all', axis=1)
U = U.dropna(axis=0, thresh=(U.shape[1]*0.5))   ## Each 'Meetpunt' has a measurement for at least half of the events
```

In [ ]:
```
## Mean normal value over all events

TT = U.mean(axis=1)
TT = pd.DataFrame(data = TT, columns=['Norm'])

## Match the data with the x and y coordinates

TT = TT.join(df.groupby(by=['Meetpunt']).mean()['x'])
TT = TT.join(df.groupby(by=['Meetpunt']).mean()['y'])

maxTT = TT[TT['Norm'] == TT['Norm'].max()]
minTT = TT[TT['Norm'] == TT['Norm'].min()]

p_count = TT[TT['Norm'] > 0].count().Norm
n_count = TT[TT['Norm'] < 0].count().Norm
```

In [ ]:
```
## Plot mean normal values for each measuring point

avg = TT['Norm'].mean()  # =0
std = TT['Norm'].std()

fig, ax1 = plt.subplots(figsize=(10, 6))
mp.plot(ax=ax1, alpha=0.3)

g1 = plt.scatter(TT['x'], TT['y'], c=TT['Norm'], cmap='bwr', vmin=(avg-std), vmax=(avg+std))

cbar = fig.colorbar(g1, ax=ax1)

plt.title('DMR: Gemiddelde Standaard')
plt.ylabel('Breedtegraad')
plt.xlabel('Lengtegraad')

plt.figtext(0.35, -0.02, f'{maxTT.index[0]} = {np.round(TT.Norm.max(), 1)}')
```

```
plt.figtext(0.35, -0.05, f'{minTT.index[0]} = {np.round(TT.Norm.min(), 1)}')

plt.figtext(0.35, -0.09, f'Meetpunten met positieve standaardwaarde = {p_count}')
plt.figtext(0.35, -0.12, f'Meetpunten met negatieve standaardwaarde = {n_count}')

plt.plot(maxTT.x, maxTT.y, 'ro', markersize=14, fillstyle='none', label=maxTT.index[0])
plt.plot(minTT.x, minTT.y, 'bo', markersize=14, fillstyle='none', label=minTT.index[0])

plt.legend()

plt.tight_layout()
```

In [ ]:
```
### Look at metadata for large rainfall values

DF_ref
```

In [ ]:
```
DDF = DF_ref.dropna()
DDF
```

In [ ]:
```
DDF.nlargest(5, 'Neerslag in mm')
```

In [ ]:
```
FF = DF_ref.nlargest(100, 'Neerslag in mm')
FF
```