



rijksuniversiteit  
groningen

stadslyceum

lyceum      atheneum      havo

## “Weg Melkweg”

### Wolkendetectie met luchthelderheid

O&O Keuze College Periode 2, Atheneum 4

Tim Wezeman A4F  
Luuk Schouten A4F  
Tom de Groot A4F  
Noa Wagter A4F



H.N. Werkman Stadslyceum  
Sectie O&O: Cees Draaijer  
RijksUniversiteit Groningen  
Faculteit Science and Engineering: Theo Jurriens  
Groningen, Nederland  
1 maart 2024



## Abstract

In dit onderzoek proberen wij het verband tussen de luchthelderheid en de bewolking te achterhalen. Hierin wordt de luchthelderheid uitgedrukt in MSAS ( $mag/arcsec^2$ ). De magnitude is de helderheid van een ster. Hogere magnitude is een zwakkere ster. Als er een hogere MSAS is, zijn er zwakkere sterren zichtbaar en is de hemel relatief donker. We gaan er vanuit dat er een verband is tussen MSAS en bewolking, maar dat er ook andere factoren, waaronder de maan en lichtvervuiling, invloed hebben op luchthelderheid.

We hebben voor ons onderzoek gebruik gemaakt van databestanden van 'Washetdonker.nl'. Hierin staan voor diverse meetstations met name de MSAS waarden. Om de invloed van de maan op MSAS te kunnen bepalen, hebben we maangegevens toegevoegd obv een almanak. Omdat we er vanuit gingen dat ook lichtvervuiling invloed heeft op MSAS, hebben voor elk meetstation een ratio gevonden die de verhouding aangeeft tussen kunstlicht en natuurlijk licht. Ook hebben we bewolkingsegegevens toegevoegd van OpenMeteo.

We hebben met eigen Python scripts deze verschillende data eerst per dag en meetstation in grafieken gevisualiseerd om een eerste beeld te krijgen van de data. Vervolgens hebben wij voor ruim 15 stations deze data samengevoegd en hebben hier een analyse op gedaan met correlatiematrixen, heatmaps, boxplots en regressie.

Door de analyse uiteindelijk toe te spitsen tot alleen de twee meest en minst lichtvervuilde locaties, zien wij een stijgende correlatie tussen MSAS en maan, en MSAS en bewolking. Hieruit maken wij op dat er eigenlijk alleen een goede conclusie te trekken is per locatie, en je niet locaties kan groeperen.

Uit deze analyse valt een redelijk verband te halen tussen MSAS en maan voor niet lichtvervuilde gebieden. Voor wel sterk lichtvervuilde gebieden is dit verband er niet, maar is er wel een redelijk verband tussen MSAS en bewolking. Deze is juist weer niet vinden in niet lichtvervuilde gebieden. We hebben dan ook alleen voor de sterk lichtvervuilde gebieden een model kunnen opstellen waarmee tot op zekere hoogte de aanwezigheid van bewolking kan worden afgeleid uit MSAS.

Wij denken dat het ontbreken van een duidelijker verband tussen MSAS en de bewolking deels te maken heeft met de gebruikte wolkendata. Om dit probleem aan te pakken hebben wij besloten om zelf de hoeveelheid bewolking te meten met zelfgemaakte foto's. Dit werkt goed, maar het kost tijd om een nieuwe dataset te krijgen. Daarom is dit uiteindelijk niet verder toegepast.

---

# Inhoudsopgave

<b>1 Oriëntatie onderzoek</b>	<b>4</b>
1.1 Intro lichtvervuiling . . . . .	4
1.2 Gevolgen lichtvervuiling . . . . .	4
1.3 Het meten van luchthelderheid . . . . .	6
1.4 WasHetDonker . . . . .	6
1.5 Theo Jurriens . . . . .	7
1.6 Eerdere onderzoeken . . . . .	7
1.7 Onderzoeksvraag en hypothese . . . . .	8
1.8 Benadering experts + onderzoek RIVM . . . . .	8
1.9 Kennis sterrenhemel en lichtvervuiling . . . . .	9
<b>2 Factoren MSAS</b>	<b>10</b>
2.1 Invloed maan . . . . .	10
2.2 Invloed locatie en lichtvervuiling . . . . .	11
2.3 Invloed jaargetij . . . . .	12
<b>3 Visuele representatie factoren</b>	<b>13</b>
<b>4 Data analyse</b>	<b>15</b>
4.1 Verzamelen data . . . . .	15
4.2 Correlatie van factoren . . . . .	15
4.3 Maan en lichtvervuiling . . . . .	18
4.4 Effect maand . . . . .	21
4.5 Invloed maan op verband MSAS en bewolking . . . . .	22
4.6 De bewolking voorspellen . . . . .	25
<b>5 Wolkendetetectie met foto's</b>	<b>26</b>
5.1 HSP berekening . . . . .	26
5.2 Verschillende camera's en resultaten . . . . .	27
5.3 Raspberry PI . . . . .	28
<b>6 Conclusie</b>	<b>29</b>
<b>7 Discussie</b>	<b>31</b>
<b>8 Nwoord</b>	<b>32</b>
8.1 Noa . . . . .	32
8.2 Luuk . . . . .	32
8.3 Tom . . . . .	32
8.4 Tim . . . . .	33
<b>9 Algemeen logboek</b>	<b>34</b>
<b>Bijlagen</b>	<b>35</b>
A Vragen enquête . . . . .	35
B Resultaten enquête . . . . .	37

C	Programmatuur externe data downloaden . . . . .	39
C1	Washeddonker data downloaden . . . . .	40
C2	Data uitlezen . . . . .	41
C3	Plot limitatie . . . . .	43
C4	Maan data . . . . .	44
C5	bewolking . . . . .	45
D	Programmatuur grafieken . . . . .	47
E	Programmatuur data analyse . . . . .	50
F	Programmatuur foto's . . . . .	57
F1	Verwerking foto's . . . . .	58
	<b>Literatuur</b>	<b>59</b>

## 1 Oriëntatie onderzoek

### 1.1 Intro lichtvervuiling

Sinds 100 jaar is er steeds meer artificieel licht gekomen. De afgelopen 25 jaar is de lichtvervuiling nog met 50 tot 270% [1] wereldwijd toegenomen. Dit komt omdat kantoorpanden, kassen en straatlantarens 's nachts hun lampen aanhouden. Niet overal is de lichtvervuiling even sterk. Zo zie je dat de randstad in Nederland veel lichter is dan bijvoorbeeld Ameland. Dit heeft te maken met de economische. Er zijn daar veel meer bedrijven. Elk jaar stijgt de lichtvervuiling in Nederland tussen de 5 en de 9% [2].

Tussen landen is ook goed een verschil te zien. Neem als voorbeeld Noord en Zuid Korea. Zuid Korea is goed ontwikkeld, en is erg licht. Noord Korea aan de ander kant is bijna helemaal donker. Dit soort verschillen meten wij met satellietbeelden. Elk jaar stijgt de lichtvervuiling wereldwijd met 2%.

### 1.2 Gevolgen lichtvervuiling

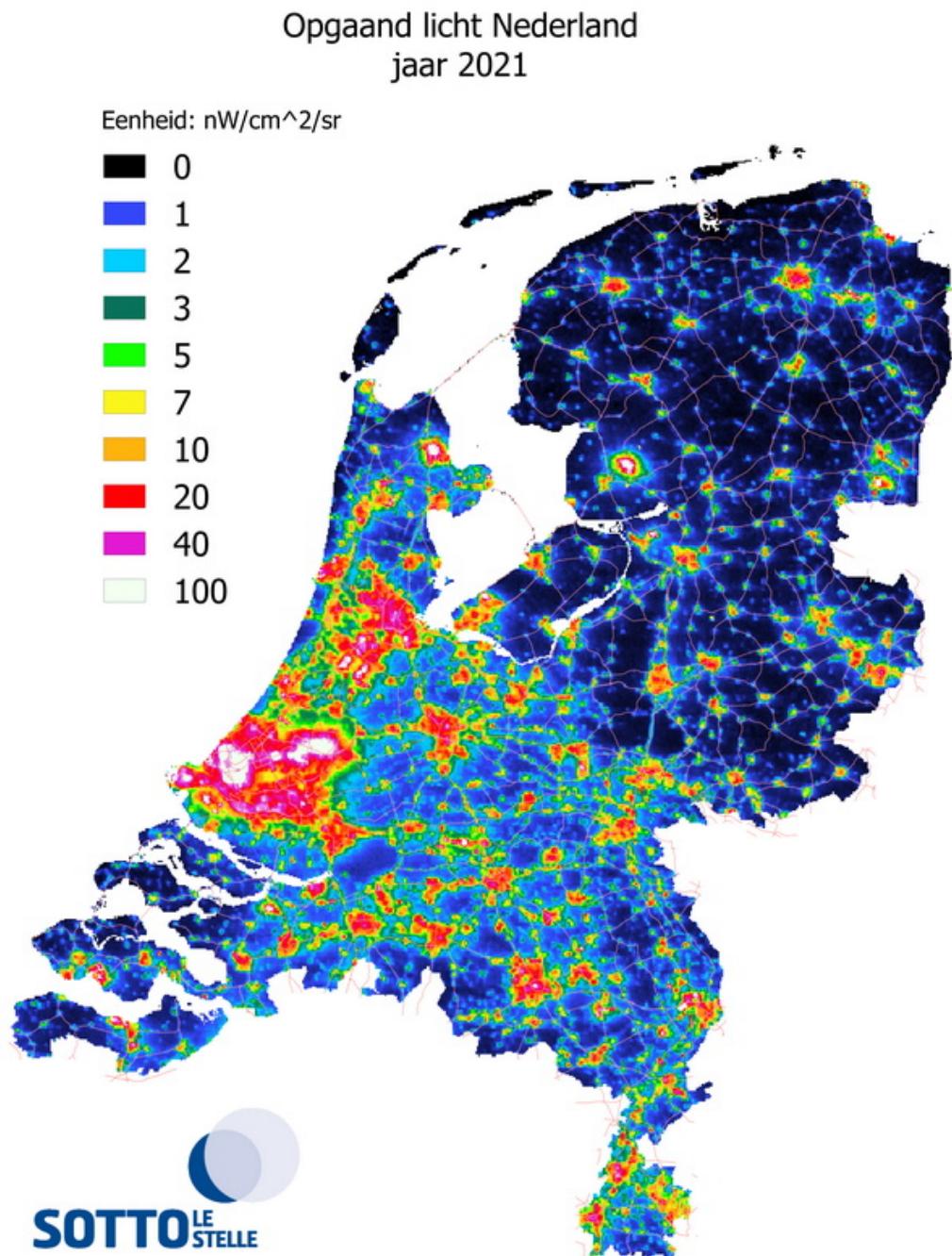
Lichtvervuiling heeft grote gevolgen voor dieren. Zo nageren vogels meestal aan de hand van zonlicht. Maar als er opeens andere lichtbronnen bijkomen, raken zij gedesoriënteerd. Insecten zoals nachtvlinders en -kevers worden aangetrokken tot licht. Andere dieren vermijden juist licht, waardoor grote lichtbronnen zoals steden of straatverlichting als een barrière van hun leefomgeving functioneren [3].

Vissen, zoals de Europese aal en de paling ondervinden op een gelijksoortige manier last van lichtvervuiling. Bij een heldere maan stoppen ze met migreren [3]. Artificiële lichtbronnen geven hetzelfde effect als een heldere maan, waardoor de aal en paling veel minder reizen dan voorheen.

Ook de voortplanting wordt verstoord door lichtvervuiling. Zo veranderen de broedtijden, en paren sommige dieren zoals kikkers een stuk sneller. En juist stadsmerels, die elke nacht in de stad te maken hebben met lichtvervuiling, hebben hier last van. Ze worden bijvoorbeeld minder vruchtbaar door een afname van testosteron door het artificiële licht [4].

Maar naast dieren hebben mensen ook flink last van lichtvervuiling. Zo wordt het slaappatroon flink verstoord. Herhaalde blootstelling aan kunstlicht vermindert de aanmaak van het slaaphormoon melatonine. Dit leidt tot slapeloosheid en een verstoring van het dag-nachtritme. Dit komt omdat licht een negatieve terugkoppeling heeft op de productie van melatonine. Lichtvervuiling kan ook leiden tot depressie [5].

Hobbyisten ervaren ook last van lichtvervuiling. Met name geïnteresseerden in de sterren. De laatste jaren is de Melkweg ook steeds moeilijker te zien, en amateurs met een telescoop moeten steeds verder reizen naar donkere gebieden [6].



Figuur 1: Mate van lichtvervuiling in Nederland

### 1.3 Het meten van luchthelderheid

De helderheid van de nachtelijke hemel wordt gemeten met zogenaamde "Sky Quality Meters", afgekort SQM [7]. De eenheid van de metingen wordt uitgedrukt in een zogenaamde MSAS of MpSAS waarde, welke staat voor Magnitude per Square Arc Second ( $mag/arcsec^2$ ) [8]. De rest van het verslag zullen wij over de MSAS waarden spreken.

Magnitudes zijn een maat voor de helderheid van een object. Hoe lager de magnitude, hoe feller het object. Zo is een 6de magnitude ster helderder dan een 12de magnitude ster.  $Arcsec^2$  is een maat voor een deel van de hemel. Een cirkel bevat 360 graden (degrees). Elke graad is opgebouwd uit 60 minuten, die weer stuk voor stuk zijn opgebouwd uit 60 seconden. 1 graad is dus gelijk aan  $\frac{1}{3600}$  seconden. Een square arcsecond heeft dus het oppervlak van een seconde \* een seconde [9].

De term MSAS betekent dat de helderheid van een bepaalde magnitude is uitgespreid over een hele  $arcsec^2$ . Als de SQM bijvoorbeeld een waarde van 21 geeft, dat is alsof al het licht van een 18de magnitude ster is uitgespreid over 1  $arcsec^2$  van de hemel [9].

Als je spreekt van een bepaalde magnitude nacht, kan je alleen sterren van deze magnitude zien of helderder. Bij een 8ste magnitude nacht kan je bijvoorbeeld geen sterren zien met een magnitude van 12. Deze term is wel subjectief. Niet iedereen heeft even goede ogen. Figuur 2 geeft een goede indicatie hoeveel sterren je kan zien bij een bepaalde MSAS waarde.

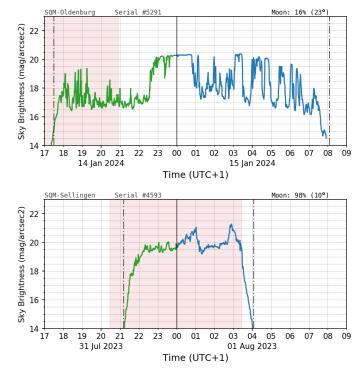


Figuur 2: Indicatie waarneming hoeveelheid sterren bij een gegeven MSAS waarde.

### 1.4 WasHetDonker

De site "Washetdonker.nl" toont elke dag de MSAS over verschillende plekken in het land. Dit doen ze met een Unihedron SQM-LU meter, een variant op de eerder genoemde SQM meter. De SQM meter wordt aangestuurd met een Raspberry Pi met de software van Mireia [10]. De data wordt opgeslagen in .dat bestanden, die dagelijks op de site gepubliceerd worden. Washetdonker levert ook grafieken met maandata en histogrammen.

Washetdonker is een onderdeel van het project "De Donkerte van het Waddengebied" en is een initiatief van het Kapteyn Instituut en ScienceLinX van de Faculteit Science and Engineering van de Rijksuniversiteit Groningen. Hier is Theo Jurriens verantwoordelijk voor.



Figuur 3: Voorbeelden grafieken WasHetDonker

## 1.5 Theo Jurriens

Voor dit onderzoek is Theo Jurriens onze opdrachtgever. In zijn kindertijd was hij al flink bezig met de sterren, maar zijn interesse in de sterrenkunde startte pas echt toen hij een telescoop kocht van het geld van zijn eerste bijbaantje [11].

Tegenwoordig werkt Theo al minstens 40 jaar bij de RUG, aan Faculteit Science and Engineering, voor “Science LinX, Jouw Energie van Morgen, Jouw Technologie van Morgen”. In zijn carrière heeft hij twee luisterboeken voor kinderen gemaakt: “Hoe plast een astronaut? 50 superslimme kindervragen over ruimtevaart” en “Kunnen vogels op de kop vliegen? 50 superslimme kindervragen”.

## 1.6 Eerdere onderzoeken

Wij hebben twee bruikbare onderzoeken gevonden over de detectie van bewolking met nachtelijke luchthelderheid. Beiden gebruiken eenzelfde soort SQM meter als WasHetDonker.

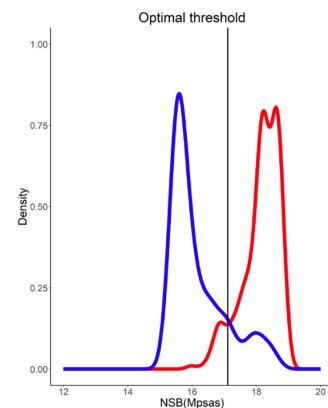
De eerste [12] kijkt voornamelijk naar het effect van maan en lichtvervuiling op een eventueel verband tussen MSAS en bewolking. De conclusies van het onderzoek luiden als volgt:

- Wel lichtvervuild: Hemel wordt helderder met bewolking. Dit komt omdat het artificiële licht door de bewolking weerkaatst wordt. Hierdoor krijgt de hemel ook een lichte gloed.
- Niet lichtvervuild: Hemel wordt donkerder met bewolking. Hier is geen artificieel licht dat weerkaatst kan worden. Door de bewolking is ook het bestaande licht van de maan en de sterren minder goed te zien.

In het tweede onderzoek [13] wordt de invloed van de nachtelijke lucht helderheid, de temperatuur van de lucht en het verschil in temperatuur tussen grond en lucht gebruikt om de aanwezigheid van bewolking te meten.

Zij hebben een bimodale verdeling gebruikt voor de analyse [14]. Deze verdeling bestaat uit twee pieken, in dit geval wel en geen bewolking. Hierbij is de y as de bewolking en de x as de MSAS. De x waarde waarbij de 2 pieken snijden, kan gebruikt worden als grenswaarde om te bepalen of er wel of geen bewolking was. Ze maken hierbij onderscheid tussen wel of geen maan.

Ze vonden dat een wolkloze nacht 1,99 MSAS helderder was dan een bewolkte nacht zonder maan, en 1,75 MSAS helderder met maan. Uiteindelijk hebben ze met een hoge accuratesse bewolking kunnen detecteren.



Figuur 4: NSKY-CD bimodale verdeling

## 1.7 Onderzoeksvergroot en hypothese

In ons onderzoek proberen wij een antwoord te vinden op de volgende vragen:

- Welke factoren spelen een rol bij de MSAS, en is er een verband tussen deze factoren onderling?
- Is er een model te maken voor wolken detectie op basis van MSAS?

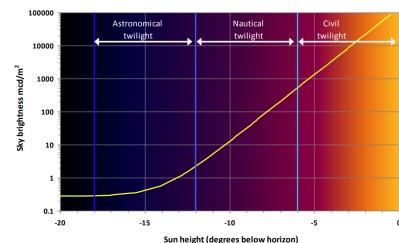
Onze hypothese is dat de maan en lichtvervuiling voornamelijk invloed hebben op de MSAS. Beide zijn bronnen van licht, en de MSAS is een eenheid van licht. We verwachten dan ook een verband tussen MSAS en maan te zien. Een verband tussen de MSAS en de aanwezigheid van bewolking is ook aannemelijk, aangezien bewolking artificieel licht kan weerkaatsen, of juist licht van de maan kan tegen houden. Of dit verband meetbaar is, proberen wij te achterhalen. In tegenstelling tot eerdere onderzoeken, visualiseren wij de data zoveel mogelijk.

## 1.8 Benadering experts + onderzoek RIVM

Voor dit onderzoek hebben wij Dorien Lolkema van het RIVM benaderd. Zij heeft onder andere meegewerk aan het rapport “Effects of atmospheric conditions on night sky brightness” [15]. Hierbij wordt ingegaan op het effect van de zon, de maan en de verschillende vormen van bewolking.

Wij hebben Dorien onze huidige hypotheses voorgelegd, en ons voorlopige plan voor de data analyse. Hier hebben wij uitgebreid antwoord op gekregen. De volgende stukken tekst zijn een verwerking van haar mail.

Om de invloed van de MSAS op de bewolking te onderzoeken, moet je eerst rekening houden met omgevingslicht. Hierbij is niet alleen lichtvervuiling van belang, maar ook de maan en de zon. Voor sterrenkundige waarnemingen wordt de astronomische nacht gebruikt. Hierbij is de zon minimaal  $18^\circ$  onder de horizon. Voor het onderzoek van het RIVM worden metingen gedaan vanaf  $15^\circ$  onder de horizon. Het RIVM doet alleen metingen waar de maan onder de horizon is. Wij wijken hier van af om ook het effect van de maan te onderzoeken.



Figuur 5: Definities nacht

Voor het onderzoek van Dorien komen alle metingen van 1 plaats in het westen van het land. De data van de bewolking worden uit een dataset van het KNMI gehaald. Dit is voor ons niet praktisch, aangezien deze dataset zich focust op 1 plaats.

Met lineaire regressie (dezelfde soort manier van data-analyse die wij haar voorlegde), was een duidelijk verband zichtbaar tussen MSAS en bewolking. Onze hypothese is hiermee redelijk in lijn met de conclusie van RIVM, ook wat betreft de invloed van de maan op MSAS.

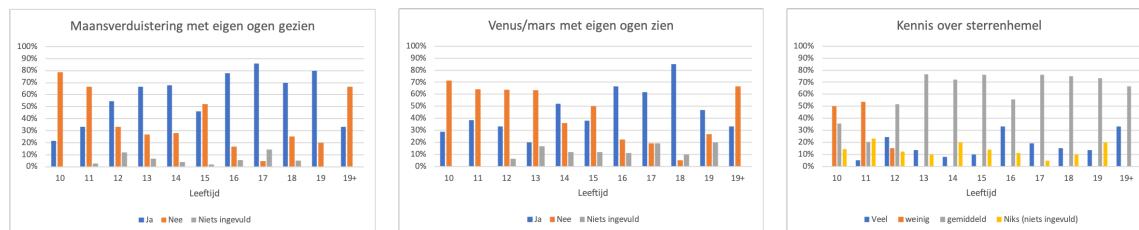
## 1.9 Kennis sterrenhemel en lichtvervuiling

Voor dit onderzoek hebben wij ook de kennis van mensen over de sterrenhemel en lichtvervuiling gemeten met een enquête. Het doel was om de verschillen in kennis tussen mensen van verschillende leeftijd te meten. De vragen en antwoorden zijn te vinden in bijlage A & B.

Als eerste hebben wij alle jaargangen van het Stadslyceum geënquéteerd. Gemiddeld genomen hebben wij per jaargang 2 klassen. De docent heeft ook de enquête ingevuld. Hierna zijn we naar basisschool O.D.S. De Starter gegaan, en hebben daar de groepen 6, 7 en 8 geënquéteerd. Ook hierbij heeft de docent de enquête ingevuld.

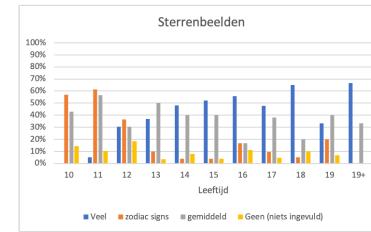
Verder hebben wij weinig ouderen geënquéteerd. Dit komt omdat er niet echt een algemene plaats is om mensen van deze doelgroep te enquêteren. Er zijn natuurlijk wel verenigingen en andere plaatsen, maar meestal komen hier mensen met dezelfde interesse. Zo hadden wij eerst het idee om naar het bezoekerscentrum van Staatsbosbeheer in het Lauwersmeer gebied te gaan. Maar omdat hier waarschijnlijk meer mensen zullen zijn met een interesse voor de sterrenhemel, zou een eventuele conclusie niet correct zijn.

Bij de antwoorden zien wij dat de vragen die gaan over de kennis relatief gelijkmatig zijn beantwoord over alle leeftijden. Voor de sterrenhemel is het wel duidelijk te zien dat kinderen van 10 en 11 jaar oud hier weinig kennis over hebben. De meeste mensen weten de basis, en voornamelijk 16 jarigen weten veel over het onderwerp. Er zit een groot verschil in kennis tussen de paar die veel weten, en de mensen die alleen de basis kennen.



Bij de actieve vragen, dus ergens naar toe gaan, of zelf de sterren bekijken, zien wij een duidelijk verband tussen leeftijd en antwoord. Oudere kinderen hebben meestal vaker Venus/Mars /maansverduistering gezien dan jongere.

We zien ook dat dat de kinderen van 10 en 11 jaar voornamelijk de sterrenbeelden kennen van de zogenaamde zodiac signs. Hierbij worden bepaalde eigenschappen aan een bepaalde geboortedatum toegekend, aangeduid met een sterrenbeeld.



Figuur 6: Interessante grafieken enquête (overige in bijlage B)

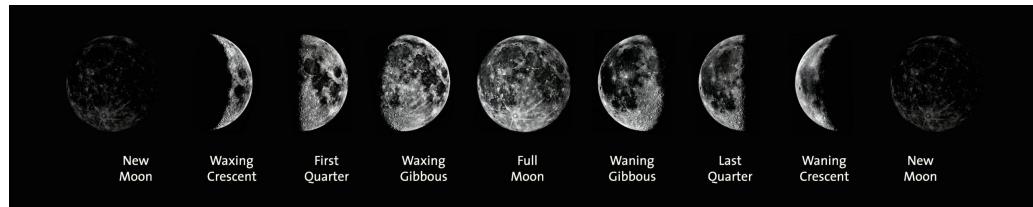
## 2 Factoren MSAS

Voordat wij kunnen beginnen aan het analyseren van data, moeten wij eerst weten welke factoren een rol spelen bij ons onderzoek. Er wordt voornamelijk gekeken naar de invloed op de MSAS.

### 2.1 Invloed maan

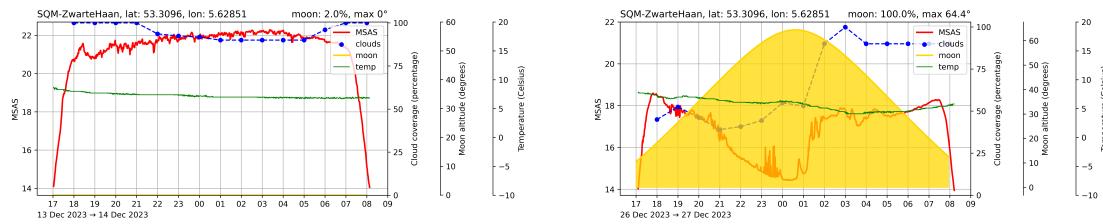
Als wij naar alle grafieken van een maand kijken, valt ons op dat er soms een duidelijk dal in de MSAS te zien is, en soms niet. Dit valt goed te zien bij ZwartheHaan in december en lijkt meestal in het midden van de maand voor te komen.

De verklaring hiervoor ligt bij de maanfase. Bij volle maan (100%), ligt de MSAS waarde laag. Bij een nieuwe maan (0%), ligt de MSAS waarde over het algemeen hoog. Dit verschil heeft dus niets te maken met de bewolking. De maan fase moet dus als een extra factor worden meegerekend. Dat de maan effect heeft, komt omdat de maan de hemel verlicht, en dus de MSAS doet afnemen.



Dit verschil is het beste zichtbaar om 12 uur 's nachts. Er is dan een versterkte verlaging van de MSAS. Als wij een paar dagen na volle maan kijken, zien wij dat dit dal opschuift. De locatie van het dal heeft te maken met de maans-opkomst, en de hoek aan de hemel van de maan. Na meervoudig testen blijkt dat de piek van de maan hoogte, overeenkomt met het dal van de MSAS. De hoek van de maan aan de horizon is dus ook een belangrijke factor.

Dit dal is alleen aanwezig bij een redelijk grote maan. Bij een maan percentage kleiner dan 40%, is er nagenoeg geen dal in MSAS meer, maar wel nog steeds een piek in maan altitude. Dit komt omdat de maan in deze fase niet meer groot/krachtig genoeg is om invloed op de MSAS te hebben. Als er geen maan is, is er veel minder verschil in MSAS in een specifieke nacht. Is er wel een groot percentage maan, dan fluctueren de MSAS waarden over het algemeen veel meer.



## 2.2 Invloed locatie en lichtvervuiling

Door ook grafieken van andere locaties te bekijken, kwamen wij er achter dat de maan niet overal even veel effect heeft. Lichtvervuilde locaties zoals Rijswijk, hebben een veel kleiner verschil in MSAS bij wel/geen maan dan de donkere locaties zoals ZwartheHaan en Moddergat.

In ZwartheHaan was het verschil in MSAS in december ongeveer 8 tussen 100% en 0% maan. In dezelfde tijd was er in Rijswijk maximaal een verschil van 4. Ook in Groningen was er maar een verschil van 4 te zien in de maand december.

Bij de meer lichtvervuilde gebieden is er ook veel minder sterk een verband tussen de hoogte van de maan en de MSAS te vinden. Meestal is er geen dal zichtbaar meer. Als er nog wel een dal is, is deze veel kleiner dan in bijvoorbeeld ZwartheHaan. De locatie is dus een belangrijke factor voor de relatie tussen de maan en MSAS.

Dit komt omdat de maan een soortgelijk effect heeft als de lichtvervuiling op de MSAS. Beide maken de hemel lichter, waardoor de MSAS daalt. Een lichtvervuild gebied kan je het beste voorstellen als een gebied met een constante volle maan. Door deze verlichting heeft het extra licht van een maan veel minder effect.

Washetdonker.nl publiceert elke dag een ranglijst van de verschillende meetstations in het land. Deze meetstations produceren elke dag data. Voor ons onderzoek hebben we hieruit een selectie gemaakt waarbij we de verschillende mate van lichtvervuiling zo evenredig mogelijk hebben meegenomen. Deze selectie is te zien in figuur 17.

Via de site “lichtvervuiling’s atlas” van David Lorenz [16] kunnen wij de gemiddelde MSAS bekijken over de hele wereld. De kaart is gebaseerd op satellietdata. Naast de MSAS geeft deze site ook een ‘Ratio’ aan. Deze Ratio is  $\frac{\text{Artificiële helderheid}}{\text{Natuurlijke helderheid}}$  [17], en geeft dus de mate van lichtvervuiling aan. De data is gebaseerd op 2020, maar is nog steeds een goede indicator en factor voor ons onderzoek.

Locatie	MSAS	Ratio
Spiekeroog	21,77	0,24
Aldeboarn	21,67	0,35
Borkum-Ostland	21,67	0,36
Vlieland-Oost	21,66	0,37
Schiermonnikoog-dorp	21,64	0,40
Moddergat	21,51	0,57
Noordpolderzijl	21,48	0,61
Lauwersoog-haven	21,44	0,68
Borkum	21,38	0,77
Burlage	21,32	0,87
ZwartheHaan	21,32	0,87
Sellingen	21,22	1,06
Westhoek	21,17	1,15
t Zandt	21,1	1,29
Texel	21,01	1,48
Gorredijk	20,82	1,96
Hippolytushoef	20,81	2,00
Katwijk	20,68	2,39
Oostkapelle	20,68	2,39
Tolbert	20,62	2,58
Lochem	20,47	3,10
Heerenveen-Station	19,94	5,70
Emden	19,94	5,70
Groningen-DeHeld	19,94	5,70
Weerribben	19,24	11,70
DeZilk	19,13	13,10
Leiden-Sterrewacht	18,07	36,20
Rijswijk	16,89	110,1
Delft	16,78	121,4

Tabel 1: Mate van lichtvervuiling bij actief bijgewerkte stations

### 2.3 Invloed jaargetij

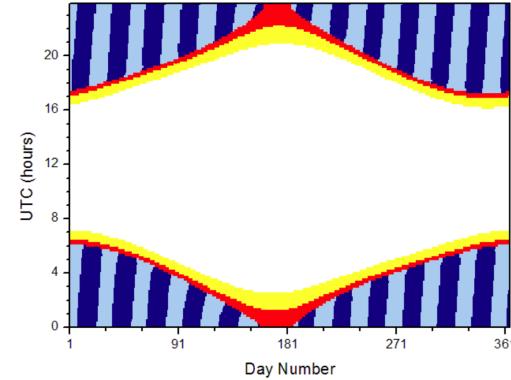
Voor het jaargetij maken wij onderscheid tussen de zomer en de winter periode. Dit verschil heeft op 2 manieren invloed op de MSAS waarden.

In de zomer zijn de dagen langer en de nachten korter. Dit komt omdat de aardas ongeveer een hoek van 23,5 graden maakt, waardoor het noordelijk halfrond in de zomer iets naar de zon gedraaid is, en in de winter iets van de zon af is gedraaid [18]. Een volle maan staat dan tegenovergesteld aan de zon. Hierdoor staat in de winter een volle maan hoger aan de hemel, waardoor de MSAS daalt. In de zomer staat een volle maan lager aan de hemel, waardoor de MSAS weinig verandert.

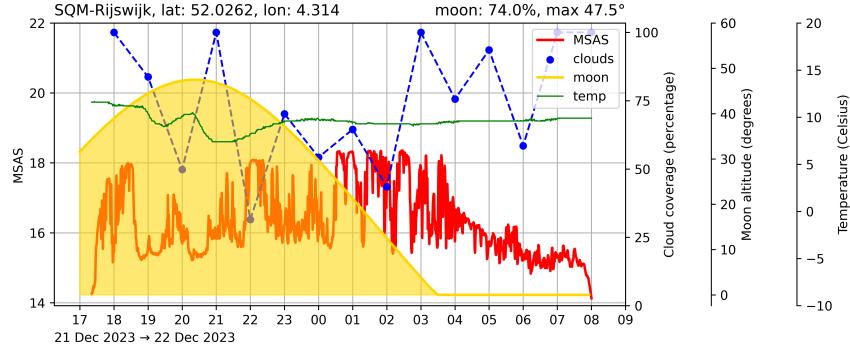
Doordat de maan zo hoog aan de hemel staat, is hij 's nachts juist laag aan de hemel. Volgens Dorien van het RIVM kunnen er pas goede resultaten behaald worden als de zon  $15^{\circ}$  onder de horizon staat. Volgens het RIVM gebeurt dit niet altijd tijdens de zomer maanden [19]. Hierdoor wordt de hemel belicht door de zon, ook al is deze onder de horizon. Dit extra licht kan een lagere MSAS geven in de zomer.

Om goed te zien wanneer er metingen gedaan kunnen worden, hebben ze een grafiek gemaakt. Hierbij is de x-as de tijd in de dag, en de y-as de dagen in het jaar. Rood = zon  $<-12$ . Geel = zon  $<-6$ . Bij de donkerblauwe strepen is de zon lager dan  $15^{\circ}$  en is er geen maan. Bij de lichtblauwe strepen is er wel maan. Hier valt goed te zien dat er volgens het RIVM geen metingen mogelijk zijn rond dag 180 (juni/juli).

Opnieuw heeft de lichtvervuiling invloed op hoe groot dit verschil is. In Rijswijk is de MSAS in de zomer hoger dan in de winter, maar een stuk lager dan de zomerse MSAS van Zwartehaan. Het jaargetij heeft weer invloed op de invloed van de maan. Deze lijkt nog steeds effect te hebben op de MSAS, maar in de zomer is deze invloed veel kleiner dan 's winters.



Figuur 7: Wanneer zijn er metingen mogelijk

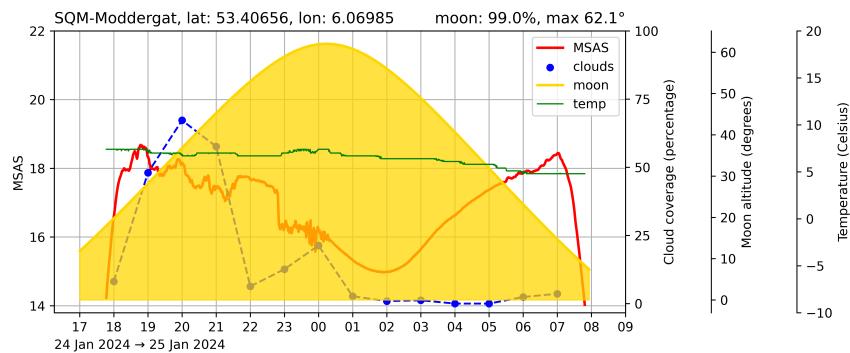


Figuur 8: Rijswijk winter

### 3 Visuele representatie factoren

Al deze factoren kunnen wij per dag en meetstation makkelijk in een grafiek weergeven. We maken voor de MSAS data gebruik van Washetdonker.nl, evenals voor de temperatuur. De bewolking wordt gedownload via een API genaamd Open Meteo [20]. De maan altitude wordt via de skyfield API [21] opgehaald. Het maan percentage wordt berekend door middel van een Almanak [22]. De code voor het downloaden van de data en het maken van de grafieken staan in bijlage C en D.

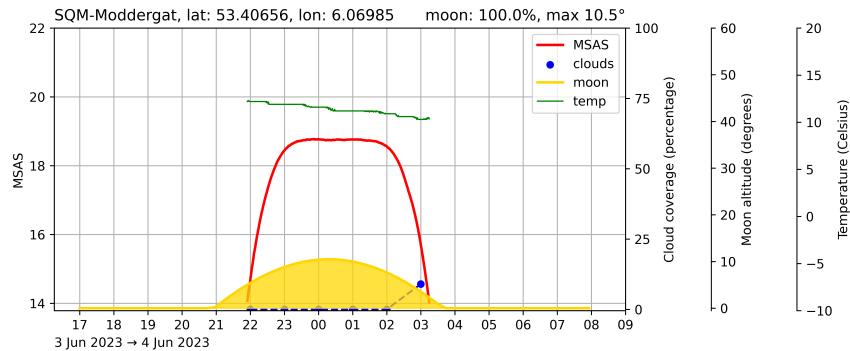
Bij deze grafieken kunnen wij al een paar verbanden zien. Later kunnen wij in de analyse kijken hoe sterk dit verband is. In de grafiek hieronder is bijvoorbeeld heel goed te zien dat als de maan altitude een piek bereikt, de MSAS een dal bereikt.



Figuur 9: Invloed maan Moddergat winter

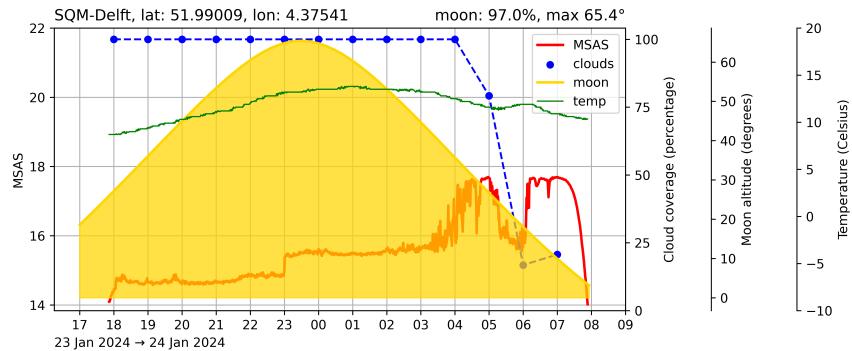
Het effect van de maan wordt hier extra versterkt door het feit dat er vanaf 12 uur 's nachts veel minder bewolking is. Het maanlicht wordt nu niet tegengehouden.

Als we dezelfde locatie in de zomer bekijken, zien wij dat de maan veel lager aan de horizon staat. Hierdoor daalt de MSAS veel minder sterk.

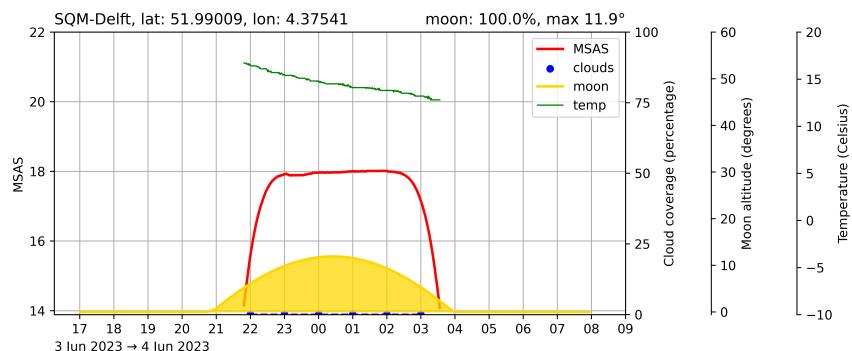


Figuur 10: Invloed maan Moddergat zomer

We kunnen dezelfde grafieken maken voor Delft, een sterk lichtvervuiled gebied. Wij zien dat de piek in maan altitude nu een veel kleiner effect heeft op de MSAS. Het dal is kleiner, maar de MSAS over de hele nacht is een stuk lager. Dit komt door de (redelijk constante) lichtvervuiling. Dit effect werd deze nacht versterkt omdat het flink bewolkt was. De wolken lichten op door de lichtvervuiling en weerkaatsen dit licht.



Figuur 11: Invloed maan Delft winter



Figuur 12: Invloed maan Delft zomer

Wij hebben ook animaties gemaakt van de grafieken van een bepaalde locatie. Hierbij wordt elke grafiek van een locatie (Zwartehaan, Noordpolderzijl, Groningen-DeHeld, Rijswijk) in een jaar achter elkaar afgespeeld. Hier is pas echt goed te zien hoe de maan invloed heeft op de MSAS. Deze animaties zijn te vinden op onze Github, net zoals al onze databestanden en code.



Figuur 13: QR code Github

## 4 Data analyse

### 4.1 Verzamelen data

Voor data analyse heb je eerst data nodig. Hiervoor is 1 dag en 1 meetstation niet genoeg. Daarom verzamelen we alle data van 17 meetstations over 2023 van website Washetdonker, aangevuld met maangegevens van het meetmoment en de mate van bewolking. We slaan de data op in csv bestanden. Tijdens de analyse kan Pandas [23] de data uit het csv bestand halen.

Voor het automatisch opslaan van veel data hebben wij een script geschreven. Dit script vult een csv bestand met alle data van een bepaalde periode van een bepaalde list of SQM meters. Wij gebruiken de locaties van de ranglijst van washetdonker.nl om te achterhalen welke stations dagelijks worden geupdate. Deze stations kunnen gebruikt worden voor analyse. Per station hebben wij per jaar zo'n 3000 metingen. De hoeveelheid metingen per maand varieert (figuur 7).

### 4.2 Correlatie van factoren

De verbanden tussen twee verschillende factoren, ook wel de correlatie genoemd, kunnen we makkelijk berekenen met Pearson correlation coefficient (PCC) [24]. Dit wordt voornamelijk toegepast bij een linear verband, het verband dat wij hopen te vinden. Pearson coefficient is een waarde tussen de 1 en -1. Hierbij is 1 een volmaakt positief verband, 0 geen verband en -1 een volmaakt negatief verband [25]. De waarde is de gemiddelde afstand tussen een meting en de trendlijn, en wordt uitgedrukt met  $\rho$  of  $r_{xy}$ . De PCC is als volgt te berekenen:

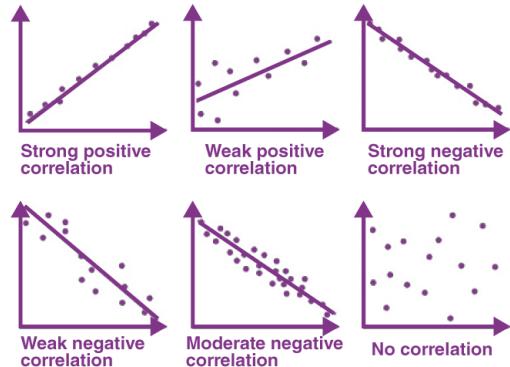
$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Hierbij geldt:

$n$  = Aantal metingen

$x_i, y_i$  = Individuele metingen

$\bar{x}, \bar{y}$  = Gemiddelde van de metingen



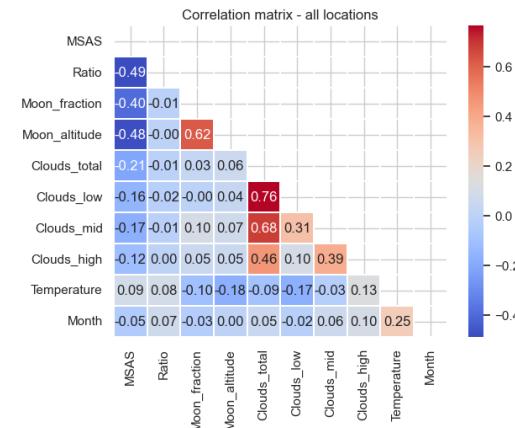
Figuur 14: Voorbeelden verschillende correlaties

Met de python libraries Pandas, Seaborn, Matplotlib, Sklearn en NumPy kunnen we deze berekeningen automatisch laten uitvoeren. Hiervoor wordt een .csv als input gebruikt, en een heatmap gegenereerd. Bij een heatmap wordt de correlatie van alle factoren in een matrix weergegeven [26]. Alle code voor de data analyse is te vinden in de bijlage.

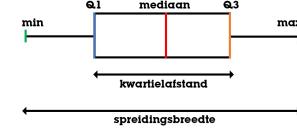
In de heatmap van figuur 15 wordt een sterker verband weergegeven met een donkerdere kleur. Een blauwe tint is een negatief verband en een rode tint is een positief verband.

Uit de heatmap trekken wij de volgende conclusies:

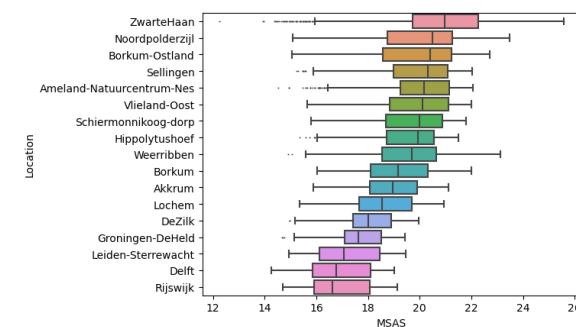
- Er is een redelijk negatief verband tussen MSAS en Ratio. Logisch, want meer lichtvervuiling geeft een lagere MSAS.
- Er is een redelijk negatief verband tussen MSAS en maan fractie, en MSAS en maan altitude. Logisch, want meer maan geeft meer licht, wat een lagere MSAS geeft.
- Er is een redelijk sterk positief verband tussen maan altitude en maanfractie.
- De totale bewolking wordt verklaard door de individuele lagen van bewolking, wat terug te zien is in de correlatie.
- Er is een zwak negatief verband tussen de MSAS en de totale bewolking.



Figuur 15: Heatmap correlatie factoren

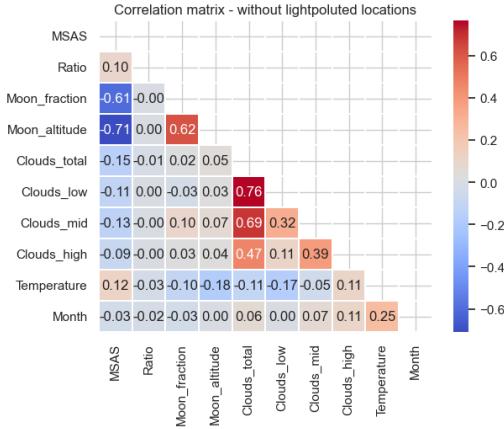


Figuur 16: Boxplot

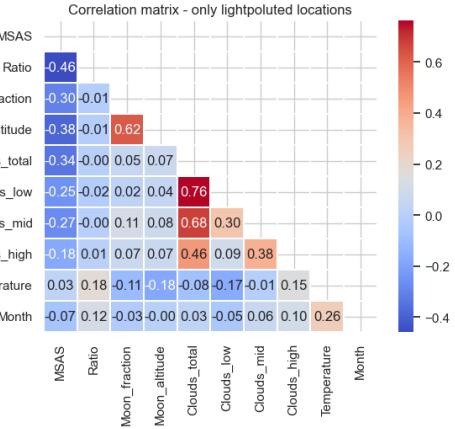


Figuur 17: Boxplot MSAS per locatie

Als eerste hebben wij opnieuw een heatmap geplot. Hierbij maken wij onderscheid tussen wel en niet lichtvervuiled. Wij definiëren lichtvervuiling vanaf nu als volgt: Bij een lichtvervuiled gebied is de hoeveelheid artificieel licht groter dan de hoeveelheid natuurlijk licht: Ratio > 1.



Figuur 18: Heatmap zonder lichtvervuiling

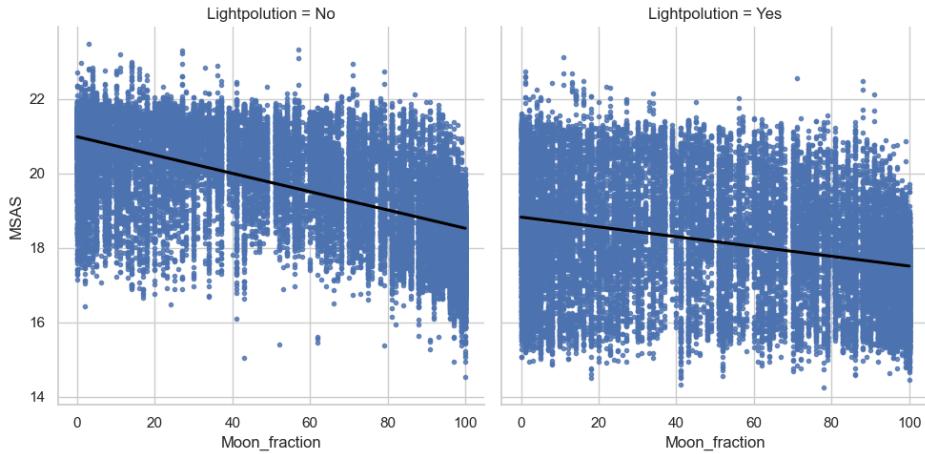


Figuur 19: Heatmap met lichtvervuiling

We kunnen nu goed het verschil zien tussen wel en geen lichtvervuiling. Bij de heatmap zonder lichtvervuiling (figuur 18) is het verband tussen de maan en de MSAS het grootste.

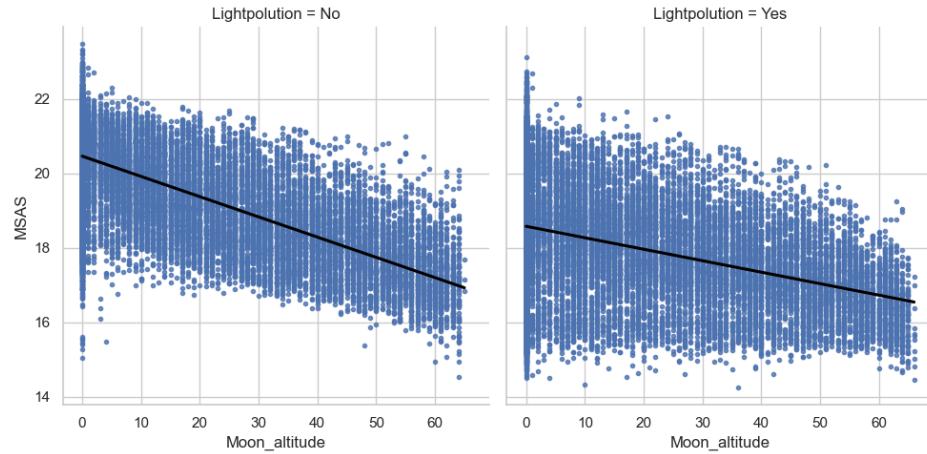
Bij de heatmap met lichtvervuiling (figuur 19) is het verband tussen de bewolking en de MSAS het grootste. Dit komt door het verminderde effect van de maan.

Beide situaties hebben dezelfde correlatie tussen maan altitude en maan factor, tussen temperatuur en maand en tussen de verschillende soorten bewolking. Dit is ook logisch, aangezien artificieel licht hier geen effect op heeft. Het verschil in correlatie tussen maan en MSAS kunnen we het beste weergeven in een scatterplot:



Figuur 20: Scatterplot maan percentage

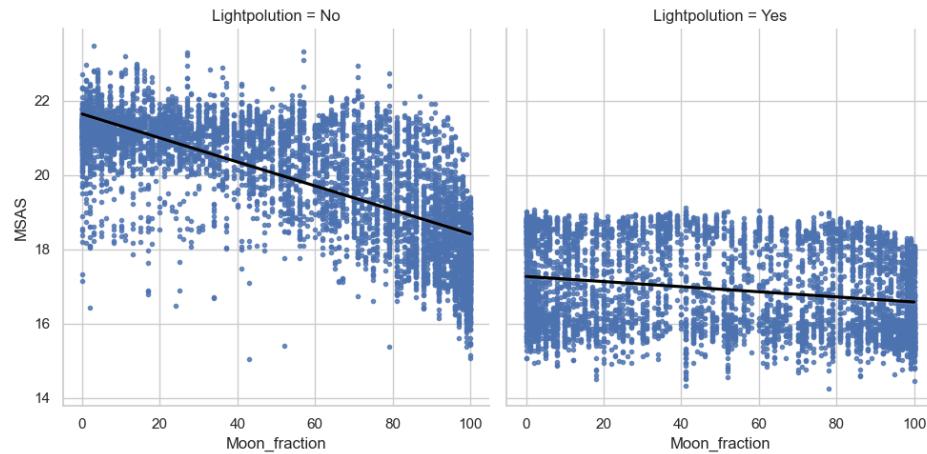
Voor de maan altitude (hoek boven horizon) kunnen wij hetzelfde doen. In deze twee grafieken is het verschil in correlatie tussen wel en niet lichtvervuild nu goed zichtbaar. Bij geen lichtvervuiling liggen alle waarden iets dichter bij elkaar.



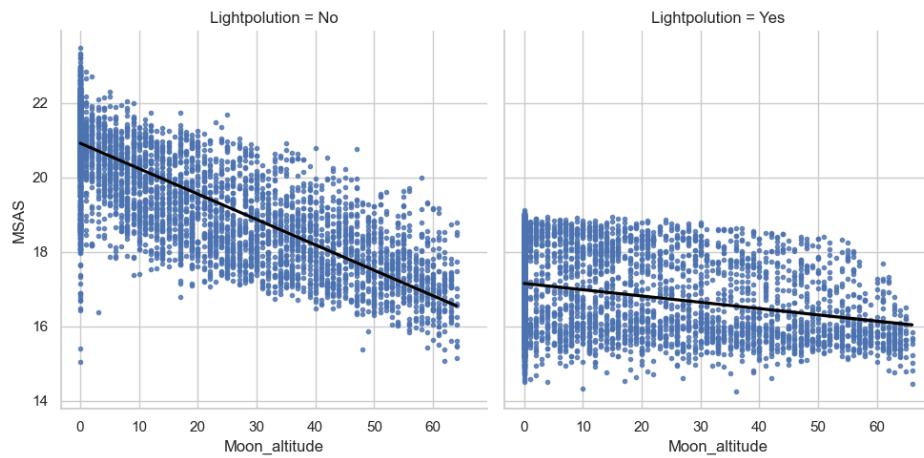
Figuur 21: Scatterplot maan altitude

### 4.3 Maan en lichtvervuiling

Om een betere conclusie over de maan en de bewolking te vormen, kiezen wij ervoor om vanaf nu alleen nog maar de 2 minst en 2 meest lichtvervuilde stations te gebruiken. In figuur 17 zien wij dat dit Noordpolderzijl, Borkum-Ostland, Delft en Rijswijk zijn. Hiervoor plotten we de correlatie tussen MSAS en maan opnieuw:

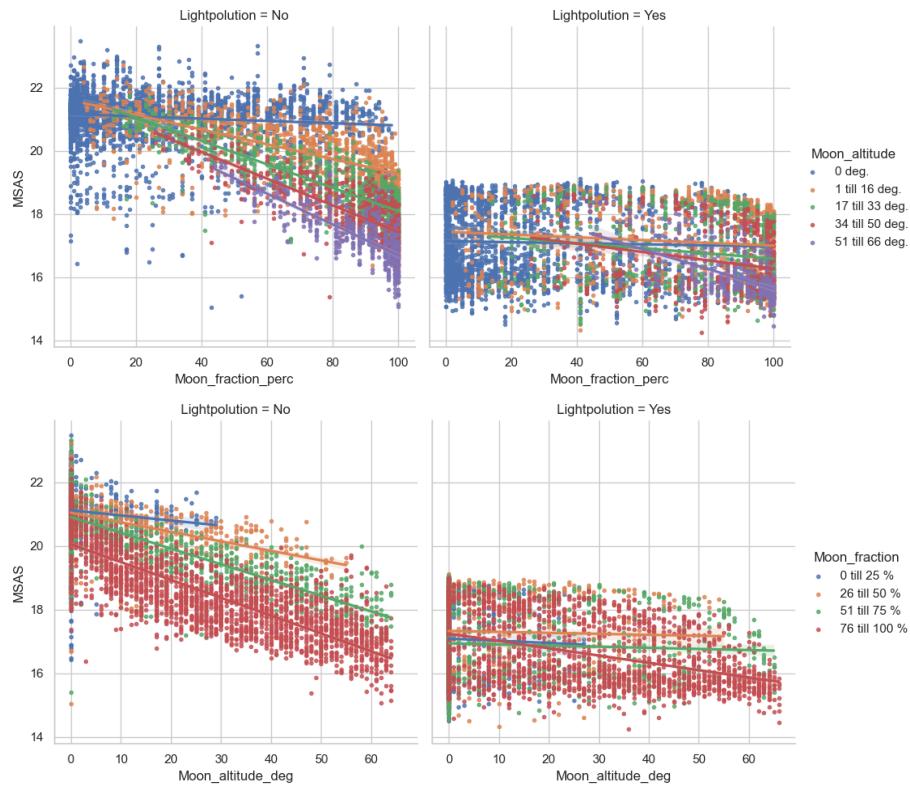


Figuur 22: Scatterplot maan percentage



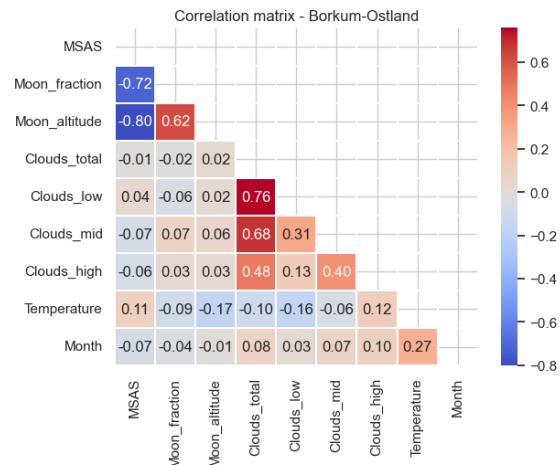
Figuur 23: Scatterplot maan altitude

In figuur 24 is de invloed van maan altitude meegegeven als groepeering bij de maan fractie, en anders om. Hier is duidelijk zichtbaar dat beide onderling een groot verband hebben. Ook is er duidelijk zichtbaar dat er alleen een hogere maan altitude mogelijk is bij een hoger percentage.

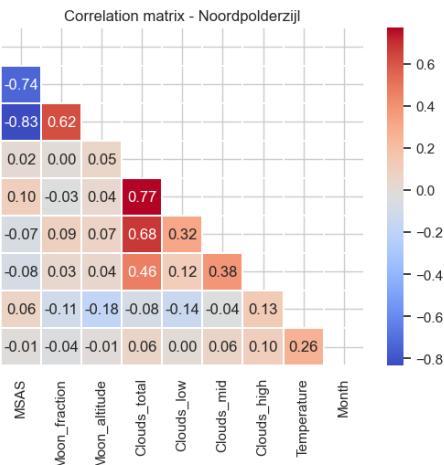


Figuur 24: Covariaat maan altitude en maan percentage

We kunnen hier nu per locatie ook makkelijk een heatmap plotten:

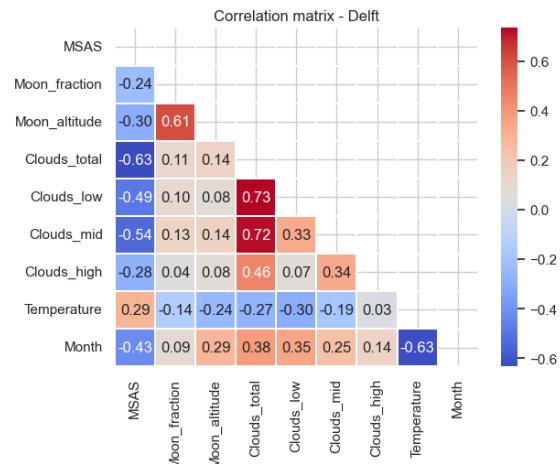


Figuur 25: Borkum-Ostland

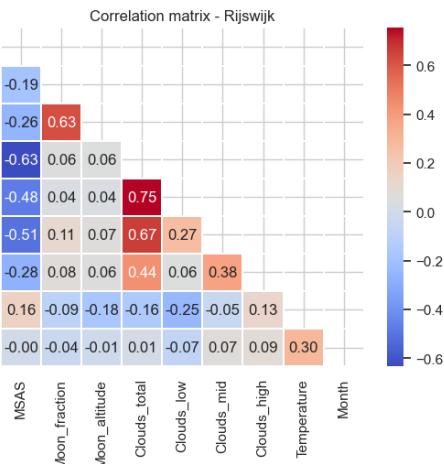


Figuur 26: Noordpolderzijl

Bij de twee minst lichtvervuilde locaties is een grotere correlatie zichtbaar tussen maan en MSAS dan in figuur 18. Toch is dit verschil relatief klein. Hieruit concluderen wij dat de stations Noordpolderzijl en Borkum-Ostland een goede representatie zijn voor niet lichtvervuilde gebieden.



Figuur 27: Delft



Figuur 28: Rijswijk

Bij deze heatmaps is een verdubbeling van correlatie tussen MSAS en bewolking zichtbaar ten opzichte van figuur 19. Onze verklaring is dat de heatmap van figuur 19 stations bevatten met een te hoge mate van lichtvervuiling om de correlatie van MSAS en maan te verminderen, maar een te lage mate van lichtvervuiling voor een goede correlatie tussen MSAS en bewolking. Deze stations moeten dus eigenlijk in een 3e categorie worden opgenomen: die met een gemiddelde lichtvervuiling.

#### 4.4 Effect maand

Wij hebben ook een boxplot gemaakt van de MSAS tegen de verschillende maanden. Hier is iets bijzonders te zien. Bij de twee meest lichtvervuilde stations zien wij een veel kleinere kwartielafstand in de maanden mei en juni. Hierbij is de MSAS juist hoog.

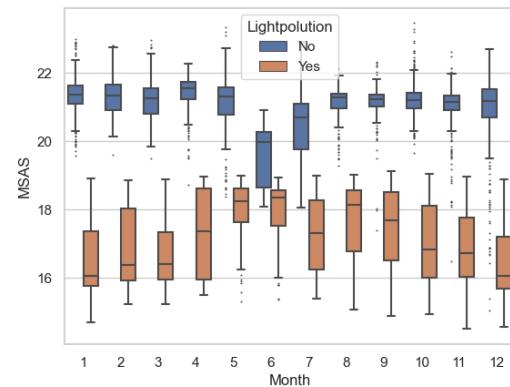
In juni gebeurt precies het tegenovergestelde in de twee minst lichtvervuilde stations. Hier is de kwartielafstand het grootst, en is de MSAS juist laag. Voor de andere maanden is de kwartielafstand heel klein.

Wat hier onder andere een rol speelt is de aanwezigheid van zon in de zomer. Deze komt in de maand juni niet onder de  $15^{\circ}$ , waardoor de MSAS lager kan zijn. Dit verklaart echter niet waarom de lichtvervuilde stations een hogere MSAS hebben.

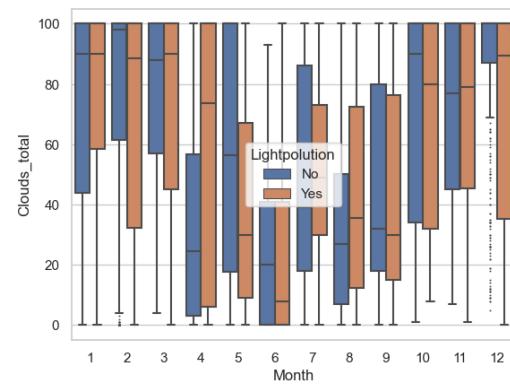
Wij hebben voor dezelfde stations ook een boxplot gemaakt van de bewolking tegen de verschillende maanden. Wij zien hier dat er in de maand juni veel minder bewolking is ten opzichte van de andere maanden. Dit lijkt de grootste reden te zijn dat de MSAS tegengesteld afwijkt bij wel en niet lichtvervuilde stations.

Als er geen bewolking is boven een niet lichtvervuild gebied, zorgt de eventuele aanwezigheid van maan ervoor dat de MSAS daalt. Als er geen bewolking is boven een wel lichtvervuild gebied, kan het artificiële licht niet weerkaatst worden. Hierdoor stijgt de MSAS.

Dit verklaart ook waarom lichtvervuilde gebieden in alle maanden grofweg dezelfde max hebben, en in april tot september hetzelfde derde quartiel. Hier is totaal geen bewolking. Dit is dus de hoogste MSAS die de maan kan aannemen. In andere maanden is het derde quartiel zo veel lager dan in juni, omdat daar een hoger percentage bewolking mogelijk is.



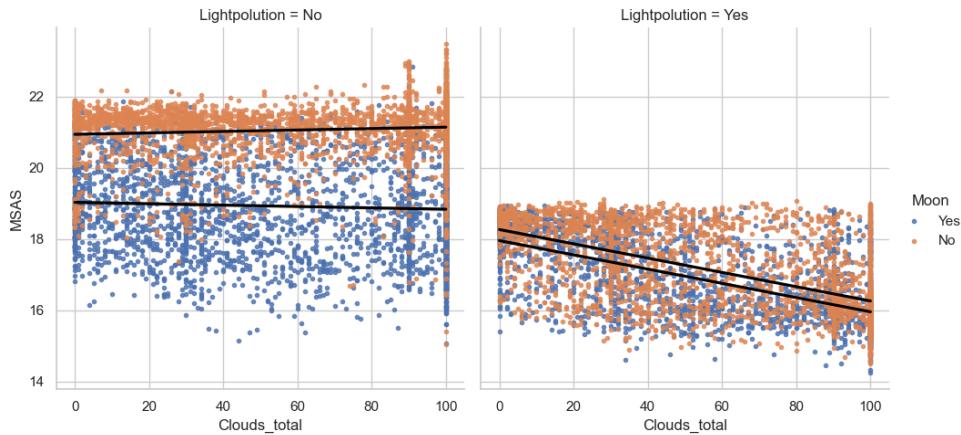
Figuur 29: Boxplot MSAS per maand



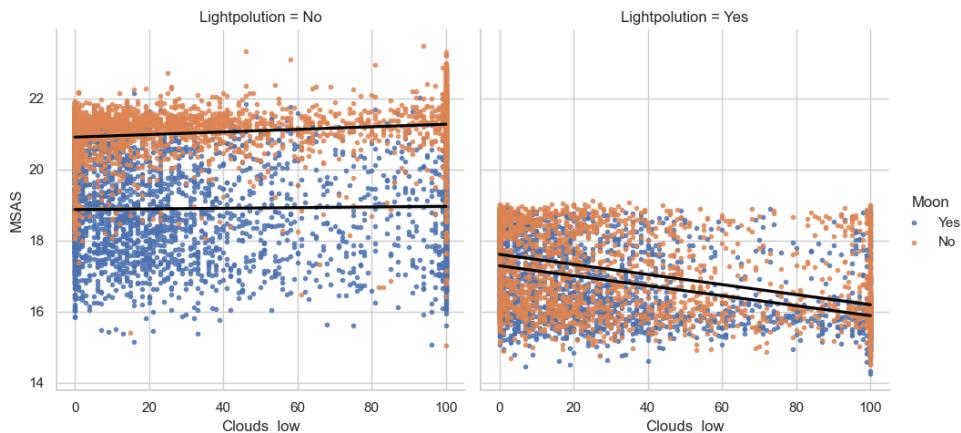
Figuur 30: Boxplot bewolking per maand

## 4.5 Invloed maan op verband MSAS en bewolking

Tot nu toe hebben wij voornamelijk de maan data geanalyseerd met maar 1 factor per grafiek. Voor de bewolkingsdata doen wij iets anders. Per bewolkingshoogte maken wij twee grafieken, met de twee minst en twee meest lichtvervuilde stations. In elke grafiek wordt de bewolkingsdata weergegeven met de aanwezigheid van maan en geen maan. Per categorie wordt een losse trendlijn getekend. In zo'n soort grafiek noemen wij de maan de covariaat. Dit principe heet Ancova [ 29].

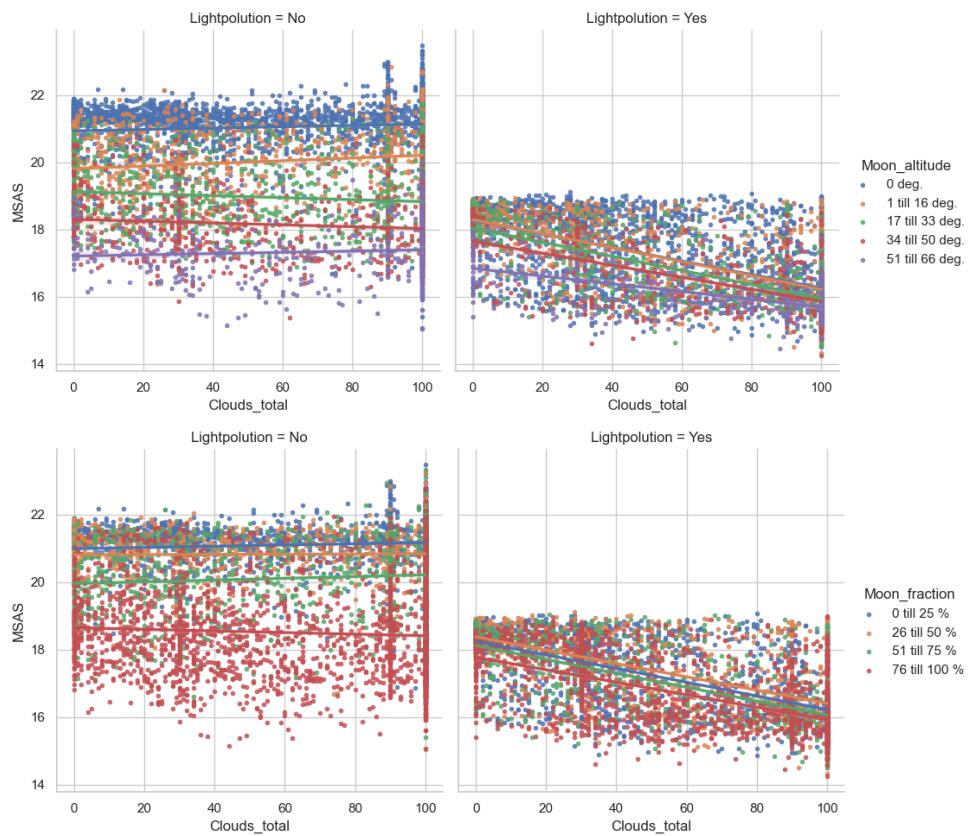


Figuur 31: Ancova bewolking totaal



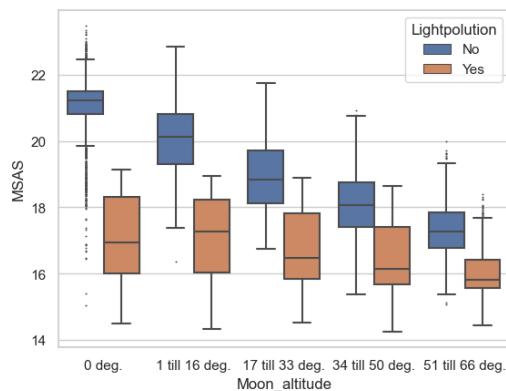
Figuur 32: Ancova bewolking laag

In deze grafieken is pas echt goed de invloed van de maan op de MSAS te zien. Bij geen lichtvervuiling zie je een redelijk geconcentreerde oranje wolk van metingen (geen maan), en een brede blauwe wolk van metingen (wel maan). De wolk van metingen is zo groot omdat de metingen van een MSAS = 16 niet gelijk zijn aan de metingen van een MSAS = 22. Bij de eerste is er waarschijnlijk een hoog percentage maan, die hoog aan de hemel staat. Bij het tweede geval is de maan waarschijnlijk kleiner en maar een paar graden boven de horizon. Dit kunnen wij ook plotten:

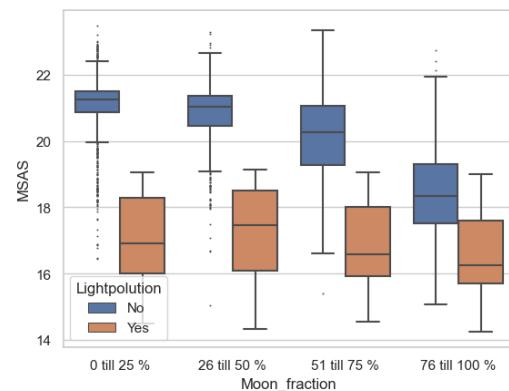


Figuur 33: Invloed maan altitude en percentage op de MSAS

De voorheen blauwe wolk stippen (wel maan) is nu opgedeeld in groeperingen van maan altitudes en fractie. Hier valt goed te zien dat een hogere altitude en fractie een lagere MSAS veroorzaken. De verbanden (lineair en exponentieel) zijn het beste te zien in een boxplot:

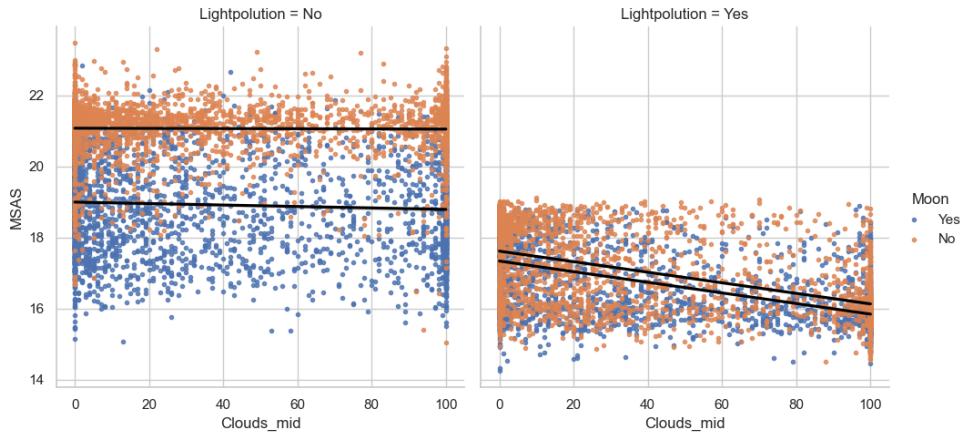


Figuur 34: Maan altitude

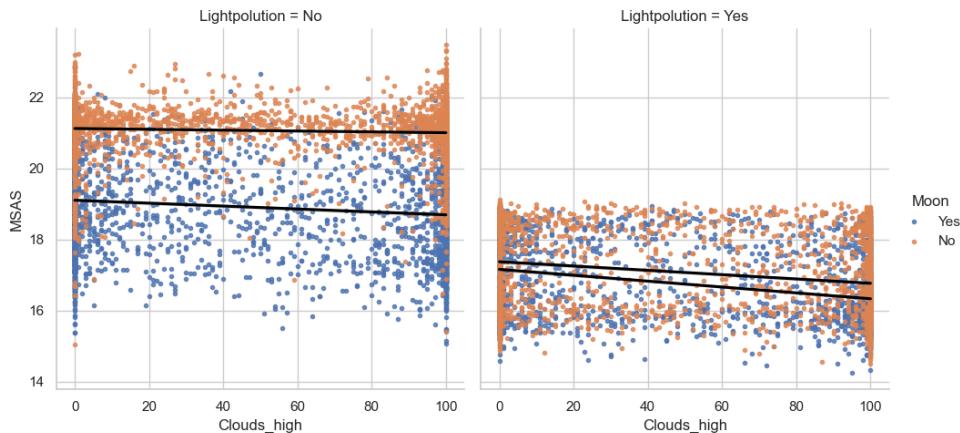


Figuur 35: Maan percentage

In de eerdere ancova grafieken keken we naar totale en lage bewolking. Als we kijken naar het verband tussen MSAS en middelhoge bewolking (tussen 2 en 6 km.) en hoge bewolking (boven 6 km.) [30], zien we bij lichtvervuiling en hoge bewolking geen verband in tegenstelling tot middelhoge en lage bewolking met lichtvervuiling. De hoge bewolking heeft praktisch een horizontale lijn.



Figuur 36: Ancova bewolking mid



Figuur 37: Ancova bewolking hoog

## 4.6 De bewolking voorspellen

Tot dusver hebben we het verband tussen de factoren en de MSAS vooral in grafieken weergegeven. Dit is echter alleen voor visualisatie en om bekend te worden met de verbanden. Als wij toekomstige voorspellingen willen doen over de MSAS aan de hand van de bewolkingsdata, moeten wij echter een model maken. Deze produceert de vergelijking van de trendlijn. Op deze manier geef je in de formule een verklarende waarde mee en krijg je een verklaarde waarde terug. Met één verklarende variabele heet zo'n model enkelvoudige lineaire regressie [31]. Het functie voorschrijf van lineaire regressie luidt als volgt:

$$Y = \alpha + \beta x$$

De constante  $\alpha$  is hierbij het snijpunt met de  $y$ -as van de trendlijn.

De constante  $\beta$  is hierbij de richtingscoëfficiënt van de trendlijn.

Bij lineaire regressie heb je geen correlatiestaart, maar een determinatie coëfficiënt:  $R^2$ . Deze waarde ligt tussen de 0 en de 1, waarbij een hoger getal een betere passendheid van de data bij de trendlijn is. Bij  $R^2 = 0.8$  kan 20% van de data variatie niet worden verklaard door het model [32].

Hieronder geven we de uitkomsten van de regressieanalyses die we hebben gedaan op verschillende combinaties van factoren. Voor berekeningen zie bijlage E.

Als de bewolking de verklarende variabele is voor MSAS, en we kijken alleen naar de twee meest lichtvervuilde stations, dan is de vergelijking als volgt:

$$Y = 18.12 - 0.0201x \quad R^2 = 0.40 \quad (1)$$

Als we kijken naar de middelmatig lichtvervuilde stations (Ratio groter dan 1 en kleiner dan 100), dan krijgen we de volgende vergelijking met correlatie:

$$Y = 19.21 - 0.0124x \quad R^2 = 0.11 \quad (2)$$

Deze daling in  $R^2$  is dusdanig dat we vanaf nu alleen nog maar kijken naar de twee meest lichtvervuilde locaties. Als we de bewolking willen voorspellen, dan moet de MSAS de verklarende variabele worden voor bewolking.  $x$  is dan de MSAS en  $y$  de totale bewolking in procenten. De vergelijking van onze definitieve model voor de wolkendetectorie op basis van MSAS, is dan:

$$Y = 395.43 - 19.86x \quad R^2 = 0.40 \quad (3)$$

Als we meerdere verklarende variabelen gaan gebruiken, heet dit principe meervoudige lineaire regressie [33]. Hierbij wordt per variabele de richtingscoëfficiënt bepaald:

$$Y = \alpha + \beta_1 x_1 + \cdots + \beta_m x_m$$

Wij hebben hier bijvoorbeeld de verklarende variabelen Ratio, maan altitude, maan fractie en Maand gebruikt. Maar omdat de Ratio nauwelijks verschilt bij de twee meest lichtvervuilde locaties, en de maan geen rol speelt bij een lichtvervuilde locatie, was het verschil in determinatie coëfficiënt zo klein (0.06) dat het praktischer is om gewoon enkelvoudige lineaire regressie te gebruiken.

## 5 Wolkendetectie met foto's

Wij denken dat één van de oorzaken voor het matige verband tussen MSAS en bewolking ligt bij de data van Open Meteo. We weten niet zeker of de data het bewolkspercentage van dat hele uur is, of dat het de gemiddelde bewolking is van dat voorgaande uur. In het laatste geval is de data een slechte representatie van de bewolking op het meetmoment. Als het bijvoorbeeld het hele uur onbewolkt is, behalve bij de MSAS-meting, koppelen wij een laag percentage bewolking aan een bewolkte MSAS-meting. Dit geeft uiteraard een fout verband. Dit is de reden dat wij onze eigen data willen verzamelen. Hiervoor willen we zelf foto's van de hemel maken.

Als er namelijk bewolking boven een lichtvervuild gebied is, weerkaatst dit het artificiële licht. Hierdoor kunnen wolken oranje of lichtgrijs oplichten tegen de donkere nachthemel. Door met een camera met lange sluitertijd foto's van de lucht te maken, kunnen wij dit kleurverschil waarnemen. Vervolgens kunnen wij aan de hand van de kleur van de foto de hoeveelheid bewolking berekenen.

### 5.1 HSP berekening

Voor de hoeveelheid bewolking wordt de helderheid van de foto berekend, aan de hand van de RGB-waarden. De berekeningen zijn gebaseerd op greyscale conversies van Adobe Photoshop door Darel Rex Finley [34] [35]:

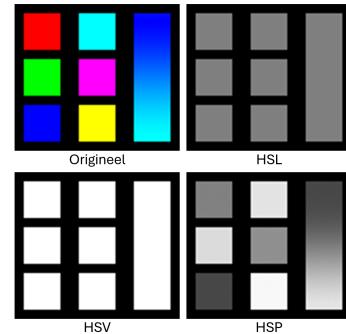
$$\sqrt{(0.241 \cdot (r^2) + 0.691 \cdot (g^2) + 0.068 \cdot (b^2)}$$

Hierbij zijn de waarden  $r$ ,  $g$ ,  $b$  de RGB waarden van de foto. Deze worden individueel ermenigvuldigd met een constante. Deze constantes zijn de mate van invloed op de helderheid van de RGB waarden, en zijn bij elkaar 1. Finley noemt dit principe HSP, gebaseerd op de principes HSV en HSL.

HSV (Hue, Saturation, Value) en HSL (Hue, Saturation, Lightness) zijn beide representaties van een kleur, net zoals RGB. Hierbij zijn de Lightness en Value beide maatstaven voor de helderheid, al zijn ze totaal anders gedefinieerd.

Bij HSV wordt de helderheid berekend van het V component. Deze is gedefinieerd als de grootste waarde van de RGB. De andere twee worden niet gebruikt. Dit geeft hierdoor een slecht beeld van de daadwerkelijke helderheid.

Bij HSL wordt de helderheid berekend van het L component. Deze is gedefinieerd als het gemiddelde van de grootste en de laagste waarde van de RGB. De middelste waarde wordt niet gebruikt [36].



Figuur 38: Voordeel HSP boven HSV en HSL

## 5.2 Verschillende camera's en resultaten

We hebben meerdere camera's geprobeerd, waaronder 2 modellen van Canon en 1 van Panasonic. Aangezien de Canon modellen een zeer beperkte sluitertijd hebben, zijn we verder gegaan met de Panasonic DC-TZ200. Deze camera kan tot 60 seconden sluitertijd, en kan ook via een app aangestuurd worden. Hier hebben wij de eerste echt bruikbare foto's mee gemaakt. Voor weergave is hier met PIL een brightness transform van x3.5 gebruikt. Voor de HSP is de oorspronkelijke helderheid gebruikt. Volledig bewolkt:  $HSP \approx 150$ .

Figuur 39:  $HSP = 43.0$ Figuur 40:  $HSP = 13.4$ Figuur 41:  $HSP = 12.4$ Figuur 42:  $HSP = 5.2$ 

Deze foto's zijn nog handmatig gemaakt. Als wij de hele nacht door foto's willen maken, moet dit proces geautomatiseerd worden. Wij hebben dit op 3 manieren geprobeerd met de DC-TZ200.

Ons eerste idee was om met een solenoïde de sluiterknop van de camera te activeren. Een solenoïde is een spoel van koperdraad, met daarin een metalen pin. Als de spoel op een spanningsbron wordt aangesloten, wordt er een magnetisch veld gecreëerd. Dit magnetisch veld stoot de metalen pin voor een deel uit de behuizing.

Wij hebben de solenoïde aangestuurd met een ESP32. Dit is een microcontroller, die GPIO (Global Purpose Input Output) pinnen bevat om externe circuits aan te sturen. De gebruikte solenoïde heeft een spanningsbron van 12V. Door een externe spanningsbron op een MOSFET aan te sluiten (elektronische schakelaar), kunnen wij met een signaal van de ESP32 de solenoïde aan of uit zetten.

Na de solenoïde op de camera te hebben gezet, kwamen wij er achter dat de gebruikte solenoïde niet krachtig genoeg was om de sluiterknop in te drukken. De gebruikte solenoïde is relatief zwak, en er zijn veel sterkere modellen te verkrijgen. Deze zijn echter duur.

We hebben met de ESP32 ook geprobeerd om een servo aan te sturen, en hiermee de sluiterknop in te drukken. Deze bleek ook niet krachtig genoeg te zijn.

Om toch foto's te kunnen maken, hebben we een camera statief van Lego gemaakt. De sluiterknop wordt nu ingedrukt door een Large EV3 motor. Na de EV3 te hebben geprogrammeerd met Python, is het gelukt om automatisch foto's te maken.

Na een test voor het raam kwamen wij er achter dat de accu van de EV3 en de DC-TZ200 niet lang genoeg mee gaan. Vooral de accu van de camera was snel leeg. Dit kwam waarschijnlijk door het denoisen van de lange sluitertijd foto's. We zouden dus dus een camera moeten gaan gebruiken die op netstroom kan werken , bijvoorbeeld een camera aangesloten op een Raspberry Pi.

## 5.3 Raspberry PI

Met een Raspberry Pi kunnen wij een camera aansturen. De Raspberry Pi is een kleine computer, aangesloten op netstroom, en de Raspberry Pi voorziet de camera weer van stroom. Zo zitten we dus niet met het probleem van een lege accu. Omdat de camera rechtstreeks op de Raspberry Pi wordt aangesloten, kan de camera gewoon programmatuur-technisch bediend worden. Dit moet ook, omdat er geen fysieke sluiterknop meer is.

Voor onze tests is een Raspberry Pi 4b gebruikt. Wij hebben twee camera's geprobeerd. De eerste was een goedkope nachtzicht camera, maar aangezien deze een maximale sluitertijd van 2 seconden heeft, werd dit hem niet. De tweede camera is de camera module v2. Deze kan een maximale sluitertijd van 10 seconden aan. Hier is de volgende code op gebaseerd, die foto's maakt met een bepaald interval tussen twee tijdstippen:

```
1 from picamera2 import Picamera2, Preview  
2 import time  
3  
4 picam2 = Picamera2()  
5 camera_config = picam2.create_still_configuration(main={"size":(1920, 1080)},  
6                                     lores={"size":(640, 480)}, display="lores")  
7 picam2.configure(camera_config)  
8 picam2.set_controls({"ExposureTime": 10000000, "AnalogueGain": 8, "ColourGains": (2, 1.81)})  
9  
10 start_hour = 17  
11 end_hour = 2  
12 photos_per_hour = 6  
13  
14 total_hours = 24 - start_hour + end_hour  
15  
16 for i in range(photos_per_hour * total_hours):  
17     picam2.start_preview(Preview.QTGL)  
18     picam2.start()  
19     time.sleep(2)  
20     picam2.capture_file(f"RPI {i}")  
21     time.sleep(((60/photos_per_hour)*60)-2)
```

Voor een vervolg hierop willen wij de Raspberry Pi HQ camera gebruiken. Deze kan ruim 670 seconden sluitertijd en is een stuk lichtgevoeliger dan de V2. Hier moet nog wel een losse lens op. Hiervoor willen wij een lens met een redelijk grote invalshoek gebruiken, om een zo groot mogelijk beeld te krijgen van de bewolking [37].



Figuur 43: RPI 4B



Figuur 44: Module V2



Figuur 45: HQ camera

## 6 Conclusie

Voor het beantwoorden van onze eerste onderzoeksvraag ('Welke factoren spelen een rol bij MSAS, en is er een verband tussen deze factoren onderling?'), kijken we eerst naar de correlatiematrix / heatmap die gemaakt is voor alle meetstations in de analyse (figuur 15). De conclusie is dan dat er vooral een negatieve correlatie is ( $\rho < -0.40$ ) tussen MSAS en lichtvervuiling ('Ratio'), maanhoogte en maangrootte. Tussen maanhoogte en maangrootte onderling is overigens een aanzienlijk positief ( $\rho = 0.62$ ) verband. Voor MSAS is bewolking een geringe factor ( $\rho$  tussen -0.12 en -0.21 voor de bewolkingsvariabelen totaal, laag, midden en hoog). Tussen de bewolkingshoechten onderling is een duidelijker verband, wat ook wel logisch is aangezien dit elkaar kan overlappen.

Omdat lichtvervuiling een duidelijke factor is, hebben we gekeken welke invloed de factoren hebben wanneer we de meetstations opdelen in niet lichtvervuiled (Ratio  $\leq 1$ ) en wel lichtvervuild (Ratio  $> 1$ ). Hierbij zagen we (figuur 18 en 19) dat de invloed van de maan op MSAS groter werd in niet lichtvervuilde gebieden en kleiner in wel lichtvervuilde gebieden. De invloed van bewolking op MSAS werd juist kleiner in niet lichtvervuilde gebieden en groter in wel lichtvervuilde gebieden.

Om te kijken hoe sterk de invloed van maan, bewolking en lichtvervuiling op MSAS kan zijn, hebben we de analyse verder toegespitst op de twee minst lichtvervuilde stations (ratio resp. 0.36 en 0.61) en de twee meest lichtvervuilde stations (ratio 110.1 en 121.4). Dit gaf voor het verband tussen MSAS en maan een grote negatieve correlatie ( $\rho = -0.70$  tot  $-0.83$ ) voor de niet lichtvervuilde stations en een zwakke negatieve correlatie ( $\rho > -0.30$ ) voor de wel lichtvervuilde gebieden. Voor het verband tussen MSAS en bewolking was er feitelijk geen correlatie ( $\rho$  tussen -0.1 en 0.1) voor de niet lichtvervuilde stations en een duidelijk negatieve correlatie ( $\rho \approx -0.60$ ) voor de wel lichtvervuilde stations.

Om te bepalen hoeveel invloed de factoren maan, lichtvervuiling en bewolking hebben op de hoogte van de daadwerkelijke MSAS waarden, kunnen uit de Ancova grafieken de volgende conclusies getrokken worden. De MSAS in niet lichtvervuilde gebieden ligt globaal tussen 17 en 22 en in sterk lichtvervuilde gebieden tussen 15.5 en 19. In niet lichtvervuilde gebieden geeft de afwezigheid van maan een MSAS van gemiddeld 21 en een hoge en/of volle maan een MSAS van 17. Door de invloed van de maan is hier dus een grotere variatie in MSAS dan in sterk lichtvervuilde gebieden.

Door het ontbreken van een correlatie tussen bewolking en MSAS in niet lichtvervuilde gebieden, kan hiervoor geen wolkendetectie model worden opgesteld. Dit geldt eigenlijk ook voor gebieden met een gemiddelde lichtvervuiling (Ratio tussen 1 en 100). Deze regressievergelijking heeft een  $R^2$  van 0.11 en heeft hiermee geen voorspellend vermogen. Alleen een enkelvoudige regressievergelijking tussen (totale) bewolking en MSAS in sterk lichtvervuilde gebieden geeft met een  $R^2 = 0.40$  een redelijke mate van voorspelling. Het model is dan  $Y = 395.43 - 19.86x$ . Het toevoegen van de factoren maan en lichtvervuiling aan deze vergelijking, blijkt de  $R^2$  niet te vergroten.

Samengevat:

- De maan heeft zeker invloed op MSAS in niet lichtvervuilde gebieden, maar heeft nauwelijks invloed op MSAS in sterk lichtvervuilde gebieden.
- De maan brengt MSAS gemiddeld met een waarde 2 omlaag in niet lichtvervuilde gebieden. In sterk lichtvervuilde gebieden is er geen invloed van de maan omdat de continue lichtvervuiling dit overschaduwt.
- De maximale MSAS is in niet lichtvervuilde gebieden een waarde 3 hoger dan die in sterk lichtvervuilde gebieden.
- De variatie in MSAS is in niet lichtvervuilde gebieden ruim anderhalf keer groter dan in sterk lichtvervuilde gebieden.
- Bewolking heeft geen invloed op MSAS in niet lichtvervuilde gebieden, maar heeft wel invloed op MSAS in sterk lichtvervuilde gebieden. Hierbij heeft lage bewolking met name invloed en hoge bewolking nauwelijks.
- Een model / regressievergelijking voor het bepalen van de aanwezigheid van bewolking op basis van MSAS, kan alleen voor sterk lichtvervuilde gebieden worden toegepast, en heeft een redelijke mate van verklarend vermogen.

Onze hypothese dat MSAS vooral wordt beïnvloed door de factoren maan, lichtvervuiling en bewolking, blijkt op basis van deze conclusies juist te zijn.

## 7 Discussie

Achteraf gezien is het ook eigenlijk best logisch dat de bewolking alleen in lichtvervuilde gebieden gedetecteerd wordt. Hiervoor kunnen we het beste de twee conclusies er bij halen van het eerst bekeken onderzoek. Hier wordt gesteld dat bewolking maanlicht tegenhoudt in niet lichtvervuilde gebieden, waardoor de MSAS hoger wordt. Bij wel lichtvervuilde gebieden weerkaatsen de wolken het artificiële licht, waardoor de MSAS lager wordt. In beide gevallen heb je dus een externe lichtbron nodig.

Bij lichtvervuilde gebieden is de maan te variabel om als goede lichtbron te gebruiken. De hoeveelheid licht verandert naarmate het maan percentage en de maan altitude toe of afnemen. Hierdoor verandert de MSAS ook, terwijl de bewolking daar niet verantwoordelijk voor is. Zonder maan is er überhaupt geen licht voor de wolken om tegen te houden, en is er dus geen verschil in MSAS te meten.

Bij lichtvervuilde gebieden is de hoeveelheid artificiële licht zoveel groter dan het natuurlijk maanlicht (Ratio > 100), dat deze geen effect meer heeft bij veranderingen. Hier is de maan dus helemaal geen factor meer. Omdat de hoeveelheid lichtvervuiling per gebied anders is, maar in een gebied meestal hetzelfde blijft, kan er per gebied wel een uitspraak over de hoeveelheid bewolking worden gemaakt.

Toch hebben wij maar een correlatie van -0.63 gevonden tussen bewolking en MSAS. Zoals eerder gezegd denken wij dat dit deels komt door een verkeerde bron voor bewolkingsdata. Daarom willen wij onze eigen data gaan verzamelen met een camera. Dit werkt goed, maar om opnieuw een dataset op te bouwen ben je weer een jaar bezig. Daarom is dat niet uitgewerkt, maar is het wel een interessante insteek voor een vervolgonderzoek.

Daarnaast hebben wij gemerkt dat de maan data van skyfield API een kleine achterstand heeft volgens deze maankalender [38]. Gemiddeld genomen bleek dit zo'n acht minuten te zien, dus wij denken niet dat dit de grootste oorzaak is. Wel hebben we gemerkt dat de zon onder de horizon toch een grotere invloed heeft gehad dan wij hebben gehoopt. Onze manier voor plot limitatie is wel correct, maar de parameters moeten iets worden bijgewerkt. De meest veilige manier is om toch tussen twee tijdstippen in te plotten, die worden bepaald door de hoogte van de zon (altitude < -15°).

Er zijn echter ook een paar kleine correlaties opgedoken waar wij geen verklaring voor hebben. Zo is er bij Rijswijk een correlatie van 0.30 tussen temperatuur en maand, terwijl er bij Delft juist een correlatie van -0.63 is tussen temperatuur en maand. Hier is ook een correlatie van -0.43 tussen maand en MSAS, terwijl hier absoluut geen verband zichtbaar is in Rijswijk.

## 8 Nwoord

### 8.1 Noa

Ik vond het een heel leuk en leerzaam project. Ik heb veel geleerd over de sterrenhemel en lichtvervuiling. Voor dit project wist ik nog helemaal niks over lichtvervuiling. Er was goede samenwerking en iedereen wist goed wat ze moesten doen. Er is goed gewerkt in de lessen waardoor er weinig thuis gedaan moest worden. Het werk is verdeeld maar iedereen wist wel waar de anderen mee bezig waren. Ik vond het nakijken van de enquêtes en er staafdiagrammen bijmaken een leuk deel van het project. Het was veel werk maar wel leuk om te doen. Aan het begin van het project zou ik bezig gaan samen met Tom om het python programma te maken. Voor dit project dacht ik dat ik goed was in programmeren aangezien ik wel goed met microbit kan programmeren, maar zodra ik zag wat we moesten schrijven voor dit python programma zag ik dat ik hier helemaal niks vanaf wist. Om deze reden heb ik de enquêtes nagekeken en daar staafdiagrammen en cirkeldiagrammen bij gemaakt. Over het algemeen vond ik het een erg leuk project.

### 8.2 Luuk

Mijn eerst indruk van dit project was erg positief omdat ik erg geïnteresseerd ben in dit onderwerp, maar achteraf bleek dat je redelijk wat ervaring moet hebben in het programmeren. Dat heb ik niet, dus ik was een beetje sceptisch toen we begonnen, maar gelukkig had de rest van het team wel de nodige ervaring, en ik kon me storten op het enquêteren. Ik denk dat de samenwerking over het algemeen wel soepel verliep, we hadden de taken goed verdeeld. Soms had ik wel het idee dat de een iets meer deed dan de andere, maar dat was minimaal. Ik kijk tevreden terug op dit project en ben erg tevreden met het eindresultaat, evenals met de positieve reactie die we van Theo Jurriens (opdrachtgever) kregen.

### 8.3 Tom

Ik vond het een erg leuk project om te maken omdat ik veel nieuwe dingen heb geleerd in onder andere python. Het maken van de python code was moeilijk aan het begin maar later begon ik het te begrijpen en werd het makkelijker. De samenwerking in het groepje ging goed en ik had het gevoel dat we elke les allemaal nuttig bezig waren. Luuk en Noa hebben de enquêtes nagekeken en diagrammen met de resultaten gemaakt. Dat is allemaal heel netjes geworden. Tim en ik zijn bezig gegaan met onderzoek en python. Zelf vond ik het leukste deel van het project de python module omdat dat goed bij mij past en ik was zelf ook aan het leren dus het heeft mij geholpen om mijn python kennis te vergroten.

## 8.4 Tim

Ik heb oorspronkelijk een ander project gekozen, namelijk “Math has the best curves”. Ik had het idee dat ik hier het beste een maquette bij kon maken, waar nog steeds techniek in te verwerken valt. Bij dit project valt niet direct een maquette te maken, en dat vond ik eerst jammer. Maar direct na de presentatie van Theo besefte ik dat hier ook genoeg leuks mee te doen is, en eenmaal thuis zat ik al direct bomvol met ideeën. Aangezien ik al eerdere ervaring had met python, ben ik voornamelijk met de code bezig geweest. Toch was de complete data analyse en api kant compleet nieuw voor mij, en heb ik extreem veel geleerd. We hebben de sterkste kant van iedereen goed kunnen benutten. Nu ik op dit project terug blik spreekt juist de theoretische kant mij erg aan, en dat wij daadwerkelijk een verband hebben kunnen vinden. In volgende projecten zou ik deze onderzoeksrand van O&O vaker willen toepassen, in plaats van alleen de ontwerp kant.



Figuur 46: Noa - Luuk - Tom - Tim

## 9 Algemeen logboek

Wij hebben niet per dag of per les een logboek bijgehouden, wel hebben wij het project in periodes opgedeeld, en de bezigheden in die periodes genoteerd.

Tussen de eerste projectdag bij de RUG en de Kerstvakantie:

Na de eerste projectdag zijn wij begonnen met de oriëntatie. Welke onderzoeken zijn er en welke factoren spelen een rol. Dit hebben Tom en Noa voornamelijk gedaan. Luuk heeft een opzet voor de enquêtes gemaakt, en Tim is bezig geweest met wolken fotograferen. (Dit was bedoeld als een tweede manier om wolken te detecteren, nog niet voor een dataset voor een analyse).

Tijdens de Kerstvakantie:

In de tweede week van de vakantie heeft Tim een opzet gemaakt voor de code van de grafieken. De verschillende API's zijn hier gebruikt en de maan data.

Tussen de Kerstvakantie en de toetsweek:

Luuk heeft de enquêtes afgemaakt en vervolgens op het Stadslyceum afgenoemt. Tim heeft de enquêtes op zijn oude basisschool afgenoemt. Tim en Tom verder met de grafieken. Noa bezig met achtergrond Theo.

Tussen de toetsweek en de tweede projectdag bij de RUG:

Luuk en Noa hebben alle enquêtes nagekeken. Noa heeft de resultaten in grafieken verwerkt. Noa heeft ook een opzet gemaakt met de presentatie, die verder is gemaakt door Tim. Tom heeft een tweede wolken API gevonden en probeert met temperatuur de bewolking te voorspellen.

Tussen de tweede projectdag bij de RUG en de voorjaarsvakantie:

Met zijn allen de feedback van Theo verwerken. Ook hebben wij samen een kleuren pallet uitgekozen voor de poster. Luuk heeft de textgrote testmatig uitgekozen. Samen opzet gemaakt poster.

Tijdens de voorjaarsvakantie tot de eindpresentatie:

Tim heeft het verslag geschreven met de input van de anderen. Uiteindelijke data analyse is uitgevoerd. Noa, Luuk en Tom maken de poster verder af.

Uiteindelijk zijn Noa en Luuk voornamelijk bezig geweest met de enquêtes, en Tom en Tim met de code. Tom is specifiek bezig geweest met de eerst genoemde wolken API en een eventueel verband tussen temperatuur en MSAS. Tim is voornamelijk begig geweest met het plotten van de grafieken en de algemene verbanden. Luuk heeft de enquêtes meer gemaakt, terwijl Noa de resultaten voornamelijk heeft verwerkt. Daarnaast is Luuk met de poster bezig geweest en Noa meer met de vooronderzoeken.

## A Vragen enquête

### **Sterrenhemel & lichtvervuiling**

De enquête gaat over de waarneembare sterrenhemel en lichtvervuiling. Deze enquête is 2 pagina's lang en bestaat uit 19 vragen. Als er een meerkeuze vraag is, zet u een cirkel om het puntje. Geef indien mogelijk extra toelichting bij het gegeven antwoord.

**1. Wat is je leeftijd? (en klas indien van toepassing)**

**2. In welke wijk, stad of dorp woon je?**

**3. Vind je het leuk of interessant om naar de sterrenhemel te kijken?**

Open antwoord: \_\_\_\_\_

**4. Wat weet je allemaal over de sterrenhemel?**

Open antwoord: \_\_\_\_\_

**5. Welke sterrenbeelden ken je?**

Open antwoord: \_\_\_\_\_

**6. Kun je deze sterrenbeelden vinden bij je woonplaats? Zo ja, hoe?**

Open antwoord: \_\_\_\_\_

**7. Heb je het idee dat je vroeger meer sterren kon zien?**

- Ja

- Nee

• Toelichting: \_\_\_\_\_

**8. Kun je de Poolster vinden bij je woonplaats?**

- Ja, hoe: \_\_\_\_\_

- Nee

• Waarom wel / niet: \_\_\_\_\_

**9. Heb je ooit met eigen ogen de Melkweg gezien?**

- Ja, waar?: \_\_\_\_\_

- Nee

• Waarom wel / niet: \_\_\_\_\_

**10. Heeft u ooit met eigen ogen Venus en/of Mars gezien?**

- Ja, waar?: \_\_\_\_\_

- Nee

• Waarom wel / niet: \_\_\_\_\_

De enquête gaat verder op de achterzijde.

**11. Heb je ooit met eigen ogen een maansverduistering gezien?**

- Ja, waar?: \_\_\_\_\_
- Nee
- Waarom wel / niet: \_\_\_\_\_

**12. Heb je ooit een telescoop gebruikt?**

- Ja
- Nee
- Waarom wel / niet: \_\_\_\_\_

**13. Wat weet je over lichtvervuiling?**

Open antwoord: \_\_\_\_\_  
\_\_\_\_\_

**14. Heb je het idee dat er lichtvervuiling is in je omgeving?**

- Ja
- Nee
- Toelichting: \_\_\_\_\_

**15. Ervaar je last van lichtvervuiling? (in je omgeving)**

- Ja, hoe?: \_\_\_\_\_
- Nee
- Toelichting: \_\_\_\_\_

**16. Welke dieren hebben volgens jou last van lichtvervuiling en waarom?**

Open antwoord: \_\_\_\_\_  
\_\_\_\_\_

**17. Heb je ooit de sterrenwacht op het Zernike complex bezocht?**

- Ja, waarom: \_\_\_\_\_
- Nee

**18. Hoe kom je aan uw kennis over de sterrenhemel en lichtvervuiling?**

- Boeken / media
- Familieleden
- Eigen interesse
- School
- Lezingen van JWG (Jongeren Werkgroep Groningen, sterrenwacht)
- Overig: \_\_\_\_\_

**19. Is er nog iets dat je wil / kan vertellen over dit onderwerp?**

Open antwoord: \_\_\_\_\_  
\_\_\_\_\_

## B Resultaten enquête





## C Programmatuur externe data downloaden

Wij hebben ervoor gekozen om alle functies in een class te zetten. Hierdoor kunnen wij makkelijk verschillen de instanties maken van een andere datum en locatie. In totaal bestaat de class uit 9 functies, plus de initialisatie functie. De class wordt in een los Python bestand aangeroepen, waar de grafieken worden geproduceerd.

### Gebruikte libraries

Python 3.11

```

1 import requests
2 from datetime import datetime, timedelta
3 from skyfield.api import Topos, load
4 from almanac import *
5 import matplotlib.pyplot as plt
6 import openmeteo_requests
7 import requests_cache
8 from retry_requests import retry
9 import pandas as pd
10 import csv
11 import sys
12 import math

```

### Opbouw class

Python 3.11

```

1 class Analser:
2     def __init__(self, date, location, path):
3
4         def get_file(self):
5
6             def find_location(self):
7
8                 def find_prefix_auto(self, first_msas, spacing):
9
10                    def read_file(self):
11
12                        def hourly_data(self, date_objects, data):
13
14                            def csv_file(self, filepath, action, rows, header):
15
16                                def is_moon_above_horizon(self, datetime_input):
17
18                                    def moon_list(self, date_list):
19
20                                        def weather_api(self, start_hour, end_hour):

```

Stel we willen het meest recente bestand downloaden, dan moeten wij de datum van gisteren gebruiken. Dat komt omdat het bestand in de nacht wordt aangemaakt. Met de python `datetime` kunnen wij de huidige datum ophalen.

Aangezien je deze datum als string kan formatteren, deden wij eerst:

`str(int(datetime.now().date().strftime('%Y%m%d')))` – 1) Na de jaarwisseling kwamen wij er al vrij snel achter dat dit tijdens een wisseling van een maand of jaar niet werkt. Je start dan

weer bij dag 1, en dag 1-1 bestaat niet. Daarom dat wij de `timedelta(days=1)` functie gebruiken. Voor verschillende delen in de code, hebben wij een ander tijd formaat nodig, daarom initializeren wij 4 formaten:

Initialisatie functie	Python 3.11
<pre> 1 def __init__(self, date, location, path): 2     self.location = location 3     self.file_path = path 4 5     self.today = date 6     self.yesterday = date - timedelta(days=1) 7 8     self.date1 = self.yesterday.strftime('%Y/%m/') 9     self.date2 = self.yesterday.strftime('%Y%m%d') </pre>	

## C1 Washetdonker data downloaden

Op de site [washetdonker.nl](http://washetdonker.nl) kan je de MSAS waarden van verschillende locaties bekijken. Standaard geeft deze site ze weer in grafieken, maar de losse .dat bestanden kan je ook downloaden. Een deel van zo'n bestand is hieronder te zien.

Voorbeeld .dat bestand	Terminal
	<pre> 1 2024-01-23T16:50:51.000;2024-01-23T17:50:51.000;7.64;26995.600;17.127;16.850 </pre>

Na meerdere bestanden te hebben geopend ter oriëntatie, bleek dat alle bestanden een op dezelfde manier geformateeerde downloadlink hebben. Door deze uniforme opmaak kunnen wij de bestanden met Python downloaden. Dit doen wij met de `requests` library.

Data downloaden	Python 3.11
<pre> 1 def get_file(self): 2     url = "https://www.washetdonker.nl/data/" + self.location + "/" + self.date1 \ 3         + self.date2 + "_120000_SQM-" + self.location + ".dat" 4 5     try: 6         response = requests.get(url) 7         response.raise_for_status() 8         with open(self.file_path, "wb") as file: 9             file.write(response.content) 10    except: 11        print('File not found, check if you are connected to wifi, ' 12              'or if you are using a valid URL') 13        sys.exit(1) </pre>	

## C2 Data uitlezen

In het .dat bestand staat de lokale tijd (UTC+1), de UTC tijd, de temperatuur, frequentie en MSAS. Wij hebben er voor gekozen om de gehele code te baseren op lokale tijd.

De tweede waarde is hierin de lokale tijd, en de laatste de MSAS. De andere waarden negeren wij eerst. Er zijn nu meerdere manieren om deze data aan afzonderlijke lists toe te voegen, wij hebben er twee uitgewerkt.

De eerste script een specifiek aantal karakters op een bepaalde plaats uit de lijn. Dit leek eerst goed te werken. Meestal worden er voorloop nullen gebruikt om lege waarden op te vullen bijvoorbeeld 009. De datum moet wel geformatteerd worden. We doen deze actie per lijn, vanaf lijn 36. De lijnen hiervoor bevatten informatie over het meetstation, of zijn headers.

Data uitlezen v1	Python 3.11
<pre> 1 def read_file(path, first, last): 2     time = [] 3     MSAS = [] 4 5     with open(path, 'r') as file: 6         for current_line_number, line in enumerate(file, start=1): 7             if current_line_number &gt;= 36: 8 9                 data1 = float(line[-7:].strip().lstrip(';')) 10                MSAS.append(data1) 11 12                data2 = line[-54:43].strip().lstrip('0;') 13                datetime_object = datetime.strptime(data2, '%Y-%m-%dT%H:%M:%S') 14 15                output_string = datetime_object.strftime('%Y:%m:%d %H:%M:%S') 16                time.append(output_string) 17 18 19    return MSAS, time </pre>	

Na dat we deze aanpak met meerdere data bestanden hebben getest, kwamen we er achter dat niet elke lijn even lang is. We moeten dus een generiekere aanpak toepassen.

Door beter naar het voorbeeld te kijken, zien wij dat de afzonderlijke data per lijn is gescheiden met een ;. Met de python `.strip()` en `.split()` functie kunnen wij zo elke lijn in de afzonderlijke data opsplitsen. Dit geeft een per lijn een list van de afzonderlijke waarden.

Lijn voor het splitsen	Terminal
<pre> 1 2024-01-23T16:50:51.000;2024-01-23T17:50:51.000;7.64;26995.600;17.127;16.850 </pre>	

Lijn na het splitsen	Terminal
<pre> 1 ['2024-01-23T16:50:51.000', '2024-01-23T17:50:51.000', '7.64', '26995.600', '17.127', '16.850'] </pre>	

## Data uitlezen v2

Python 3.11

```

1 def read_file(self):
2     # Initializing lists
3     time = []
4     msas = []
5     temp = []
6
7     # Iterate through the lines between two prefixes
8     with open(self.file_path, 'r') as file:
9         for line_num, line in enumerate(file, start=1):
10             if self.start_line <= line_num <= self.end_line: # defined in find_prefix auto
11                 values = line.strip().split(';') # split the line in the separate values
12
13                 second_value = values[1].replace('.000', '')
14                 datetime_object = datetime.strptime(second_value, '%Y-%m-%dT%H:%M:%S')
15                 time.append(datetime_object.strftime('%Y-%m-%d %H:%M:%S'))
16
17                 third_value = values[2].replace('.000', '')
18                 temp.append(float(third_value))
19
20                 sixth_value = values[5].replace('.000', '')
21                 msas.append(float(sixth_value))
22
23     return time, temp, msas

```

In lijn 10 specificeren wij de data die we willen ophalen uit het .dat bestand. Oorspronkelijk was dit: `if line_num >= 36`. Al snel bleek dat dit niet een praktische benadering is, aangezien de SQM meter al begint met meten voordat de zon onder gaat. Daarom hebben wij een `self.start_line` en `self.end_line` gedefinieerd. Dit doen wij in 2.3 Plot limitatie.

De latitude en longitude (locatie) van de SQM meter staan ook vermeld in lijn 9 van het .dat bestand. Aangezien de locatie nodig is voor het ophalen van maan en bewolkings data, schrijven wij deze ook weg. Door de `.split()` functie meerdere keren te gebruiken kunnen wij de lat en lon scheiden van de rest van de regel. Vervolgens returnen wij ze als twee variabelen.

## Locatie ophalen

Python 3.11

```

1 def find_location(self):
2     with open(self.file_path, 'r') as file:
3         for current_line_number, line in enumerate(file, start=1):
4             if current_line_number == 9:
5                 a, b, c = line.split(',')
6                 d, e = a.split(':')
7                 self.lat = float(e)
8                 self.lon = float(b)
9
10    return float(e), float(b)

```

### C3 Plot limitatie

De variabelen 'first' en 'last' worden gebruikt om de grafiek te limiteren. Wij willen alleen de data gebruiken als de grafiek afvlakt en de MSAS niet meer wordt beïnvloed door de schemering. Wij hebben gemerkt dat dit een uur nadat de MSAS een waarde van 14 heeft bereikt, en een uur voordat de MSAS weer lager dan 14 wordt. Dit wordt bereikt door de tijden te nemen waarbij de MSAS groter is als 14. Vervolgens wordt bij deze tijd een uur opgeteld. Daarna zoeken wij de lijnnummers met de nieuwe tijden.

Plot limitatie

Python 3.11

```
1 def find_prefix_auto(self, first_msas, spacing):
2     start_msas, end_msas = None, None
3
4     with open(self.file_path, 'r') as file:
5         for line_num, line in enumerate(file, start=1):
6             if line_num >= 36: # 36 is the first data line in .dat file
7
8                 # Split each line into values using semicolon as the delimiter
9                 values = line.strip().split(';')
10
11             if values and float(values[-1]) > first_msas:
12                 if not start_msas:
13                     start_msas = line_num
14                 end_msas = line_num
15
16             if line_num == start_msas:
17                 original_time_str = values[0]
18                 original_time = datetime.strptime(original_time_str, "%Y-%m-%dT%H:%M:%S.%f")
19
20                 new_time = original_time + timedelta(minutes=spacing)
21                 begin_time_str = new_time.strftime("%Y-%m-%dT%H:%M")
22
23             if line_num == end_msas:
24                 original_time_str = values[0]
25                 original_time = datetime.strptime(original_time_str, "%Y-%m-%dT%H:%M:%S.%f")
26
27                 new_time = original_time - timedelta(minutes=spacing)
28                 end_time_str = new_time.strftime("%Y-%m-%dT%H:%M")
29
30     with open(self.file_path, 'r') as file:
31         for start_line_number, line in enumerate(file, start=1):
32             if line.startswith(begin_time_str):
33                 self.start_line = start_line_number
34
35     with open(self.file_path, 'r') as file:
36         for end_line_number, line in enumerate(file, start=1):
37             if line.startswith(end_time_str):
38                 self.end_line = end_line_number
```

Dit doen wij als vervanging van de astronomische nacht. Op deze manier verkrijgen wij zoveel mogelijk metingen, zonder de invloed van de zon.

## C4 Maan data

Met de skyfield api kunnen wij precies de hoogte van de maan berekenen. Hiervoor hebben wij een date object nodig, en de locatie. Vervolgens wordt er True/False gereturned als er wel/geen maan is. Is er wel maan, dan wordt ook de hoogte meegegeven. Wij hebben gemerkt dat skyfield meestal een offset van 8 minuten heeft. Hier houden wij rekening mee.

Hoogte maan	Python 3.11
<pre> 1 def is_moon_above_horizon(self, datetime_input): 2     ts = load.timescale() 3     observer = Topos(latitude_degrees=self.lat, longitude_degrees=self.lon) 4     t = ts.utc(datetime_input.year, datetime_input.month, datetime_input.day, 5                datetime_input.hour - 1, datetime_input.minute, datetime_input.second) 6 7     planets = load('de421.bsp') 8     earth, moon = planets['earth'], planets['moon'] 9 10    astrometric = (earth + observer).at(t).observe(moon) 11    _, _, _ = astrometric.apparent().altaz() 12 13    if alt.degrees &gt; 0: 14        return alt.degrees &gt; 0, alt.degrees 15    else: 16        return alt.degrees &gt; 0, 0 </pre>	

Om de data te plotten in een grafiek, moet de data in een list staan. In de functie moon list wordt de hoogte van de maan per datum instantie van washetdonker berekend.

De maan factor wordt met een almanac berekend. Deze is geprogrammeerd door Daniel Minor, en wordt hier dus niet weergegeven. De berekeningen zijn gebaseerd op: Astronomical Algorithms, 2nd Edition by Jean Meeus, published by Willmann-Bell, 1998. Wij hebben de code gemodificeerd om te werken met lokale tijd (UTC + 1).

Maan list	Python 3.11
<pre> 1 def moon_list(self, date_list): 2     moon = [] 3     moon_alt = [] 4     moon_frac = [] 5 6     year, moth, day = self.today.year, self.today.month, self.today.day 7     frac = round(illuminated_fraction_of_moon(year, moth, day), 2) 8 9     # Get the moon altitude and fraction for every date object 10    for date in date_list: 11        moon.append(self.is_moon_above_horizon(date)[0]) 12        moon_alt.append(self.is_moon_above_horizon(date)[1]) 13        moon_frac.append(round(illuminated_fraction_of_moon(year, moth, day), 2)) 14 15    return moon, moon_alt, moon_frac, frac </pre>	

C5 Bewolking

Om een verband tussen de bewolking en de MSAS te vinden, hebben wij ook bewolkings data nodig. KNMI publiceert bewolkingsdata van de afgelopen 2 weken op hun site ADAGUC Viewer [39]. Hoewel deze data niet te downloaden valt, is het wel een goede controle voor andere data bronnen.

De site weatherapi publiceert wel downloadbare bewolkingdata via een json bestand. Deze kan tussen twee tijdstippen gedownload worden via python:

```
1 city = "Groningen"
2 station = "Groningen-DeHeld"
3 time1 = "2024-01-20"
4 url = f"https://api.weatherapi.com/v1/history.json?
5     key=d0d0567147184e79884125607242601&q={city}&dt="
6
7 def Get_Cloud(url, date1, date2):
8     Clouds = []
9     NewUrl = url + f'{date1}'
10    response = requests.get(NewUrl)
11    if response.status_code == 200:
12        data = response.json()
13        for hour in range(start_hour, 23):
14            cloud_data = data['forecast']['forecastday'][0]['hour'][hour]['cloud']
15            Clouds.append(cloud_data)
16    else:
17        print(f'Error: Unable to fetch data. Status code: {response.status_code}')
18
19    NewUrl = url + f'{date2}'
20    response = requests.get(NewUrl)
21    if response.status_code == 200:
22        data = response.json()
23        for hour in range(end_hour + 1):
24            cloud_data = data['forecast']['forecastday'][0]['hour'][hour]['cloud']
25            Clouds.append(cloud_data)
26    else:
27        print(f'Error: Unable to fetch data. Status code: {response.status_code}')
28
29 return Clouds
```

Helaas blijkt deze data na testen niet overeen te komen met de ADAGUC viewer. Daarnaast wordt alleen de data van de afgelopen 2 weken verstrekt. Dit is niet genoeg om een betrouwbaar model te maken, aangezien je alleen data van een specifiek jaargetij kan verkrijgen.

Een andere site die bewolkingsdata publiceert is Open Meteo. De gegevens van Open Meteo zijn als .csv bestand te downloaden door middel van een API. Deze data lijkt wel overeen te komen met die van de ADAGUC Viewer. Daarnaast bevatten de archieven van Open Meteo data van tot wel 80 jaar terug. Lang genoeg dus om een model te voeden.

De site van Open Meteo geeft onder andere python code voor het ophalen van de data. Deze code is door ons lichtelijk aangepast, zodat hij werkte met lokale tijd (UTC+1).

## Data bewolking v2

Python 3.11

```

1 def weather_api(self, start_hour, end_hour):
2     cache_session = requests_cache.CachedSession('.cache', expire_after=-1)
3     retry_session = retry(cache_session, retries=5, backoff_factor=0.2)
4     openmeteo = openmeteo_requests.Client(session=retry_session)
5
6     url = "https://archive-api.open-meteo.com/v1/archive"
7     params = {
8         "latitude": self.lat,
9         "longitude": self.lon,
10        "start_date": self.yesterday.strftime('%Y-%m-%d'),
11        "end_date": self.today.strftime('%Y-%m-%d'),
12        "hourly": "cloud_cover",
13        "timezone": "Europe/Berlin"
14    }
15    responses = openmeteo.weather_api(url, params=params)
16    response = responses[0]
17    hourly = response.Hourly()
18    hourly_data = {
19        "date": pd.date_range(
20            start=pd.to_datetime(hourly.Time(), unit="s"),
21            end=pd.to_datetime(hourly.TimeEnd(), unit="s"),
22            freq=pd.Timedelta(seconds=hourly.Interval()),
23            inclusive="left"),
24        "cloud_cover": hourly.Variables(0).ValuesAsNumpy()
25    }
26    time = []
27    clouds = []
28
29    for i, date, cloud_cover in zip(range(len(hourly_data["date"])), hourly_data["date"],
30                                     hourly_data["cloud_cover"]):
31        time.append(date)
32        cloud_percentage = cloud_cover
33        clouds.append(cloud_percentage)
34
35    date1 = self.yesterday.strftime('%Y-%m-%d')
36    date2 = self.today.strftime('%Y-%m-%d')
37
38    find_start = datetime.strptime(f'{date1} {start_hour}', '%Y-%m-%d %H:%M:%S')
39    find_end = datetime.strptime(f'{date2} {end_hour}', '%Y-%m-%d %H:%M:%S')
40
41    start = time.index(find_start - timedelta(hours=1))
42    end = time.index(find_end - timedelta(hours=1))
43    start2 = time.index(find_start)
44    end2 = time.index(find_end)
45    stripped_cloud_cover = clouds[start:end + 1]
46    stripped_time = time[start2:end2 + 1]
47
48    if any(math.isnan(x) for x in stripped_cloud_cover):
49        print("Failed to retrieve cloud data")
50
51    return stripped_cloud_cover, stripped_time

```

## D Programmatuur grafieken

Met de opgehaalde data kunnen grafieken geplot worden. Dit doen wij met Matplotlib. Wij hebben er voor gekozen om lijn grafieken te maken van de MSAS, in plaats van een histogram. Wij denken dat hier makkelijker eventuele verbanden zichtbaar zijn. De volgende stukken code zijn één geheel.

We initializeren onze class eerst als WHD (Was Het Donker). Vervolgens wordt de MSAS, de lokale tijd, de verschillende maan waarden en de bewolking naar lists weggeschreven. Deze worden later geplot. Eerst was dit proces zeer sloom. Het ophalen van de data kostte 12 tot 18 seconden. Dit komt omdat bij elke MSAS-meting de hoogte van de maan opgehaald wordt. Door om de 20 metingen de hoogte van de maan te bepalen, duurt het ophalen van de data nog maar 1 tot 2 seconden. Dit doen we met: `moon_date_objects = moon_long_date_objects[::20]`

Ophalen data
Python 3.11

```

1 from module import *
2 import time
3
4 date = datetime(2023, 12, 22)
5 location = 'ZwarteHaan'
6 file = "data.dat"
7
8 yesterday = date-timedelta(days=1)
9
10 WHD = Analyser(date, location, file)
11
12 WHD.get_file()
13 WHD.find_prefix_auto(14, 0) # Plot if MSAS > 14. No time limitation for the plot
14 lat, lon = WHD.find_location()
15
16 time_data, temp_data, msas_data = WHD.read_file()
17 date_objects = [datetime.strptime(timestamp, '%Y:%m:%d %H:%M:%S') for timestamp in time_data]
18
19 start_date = datetime(yesterday.year, yesterday.month, yesterday.day, 17)
20 moon_long_date_objects = [start_date + timedelta(seconds=42 * i) for i in range(1286)]
21
22 # Only get the cloud data every 20 date objects. This greatly reduces runtime
23 moon_date_objects = moon_long_date_objects[::20]
24 is_moon, moon_alt, moon_frac, frac = WHD.moon_list(moon_date_objects)
25 frac = frac*100
26
27 new_msas_datetimes, new_msas = WHD.hourly_data(date_objects, msas_data)
28
29 # Get the first and last datetime. This is used to get the cloud data
30 first_datetime = new_msas_datetimes[0]
31 last_datetime = new_msas_datetimes[-1]
32
33 # Format time component of the datetime objects
34 first_hour_str = first_datetime.replace(minute=0, second=0, microsecond=0).strftime('%H:%M:%S')
35 last_hour_str = last_datetime.replace(minute=0, second=0, microsecond=0).strftime('%H:%M:%S')
36
37 clouds, cloud_time = WHD.weather_api(first_hour_str, last_hour_str)

```

De 4 verschillende dingen die wij willen plotten (MSAS, temp, maan, bewolking), gebruiken allemaal een andere schaal. Daarom dat er 4 y-assen gedefinieerd worden. Eén links en drie rechts.

## Initialisatie plot

Python 3.11

```

1 # Initialize plot
2 fig, ax = plt.subplots(figsize=(10, 4))
3 fig.subplots_adjust(right=0.75)
4
5 # Create multiple y axes
6 ax2 = ax.twinx()
7 ax3 = ax.twinx()
8 ax4 = ax.twinx()
9
10 # Define the distance of the y axes
11 ax3.spines.right.set_position(("axes", 1.15))
12 ax4.spines.right.set_position(("axes", 1.3))

```

Hierna worden de verschillende data lijsten geplot. MSAS, temperatuur en maan hoogte zijn lijn grafieken. Het oppervlakte onder de maan hoogte wordt geel gekleurd. Hoe donkerder deze kleur, hoe voller de maan. Als laatste wordt de bewolking uitgedrukt in een scatterplot. Dit komt omdat de data van Open Meteo maar eens per uur is. Hiertussen is een stippellijn getekend.

## Plotten data

Python 3.11

```

1 # Plot moon data
2 ax3.plot(moon_date_objects, moon_alt, color='gold', linewidth=2, label='moon')
3 ax3.fill_between(moon_date_objects, 0, moon_alt, color='gold', alpha=(frac/100)*0.75)
4
5 # Plot temp en MSAS data
6 ax4.plot(date_objects, temp_data, color='green', linewidth=1, label='temp')
7 ax4.plot(date_objects, msas_data, color='red', linewidth=2, label='MSAS')
8
9 # Plot clouds data
10 ax2.scatter(cloud_time, clouds, color='blue', marker='o', s=25, label='clouds')
11 ax2.plot(cloud_time, clouds, linestyle='--', color='blue')

```

Vervolgens worden de labels van de 4 y-assen en de x-as gedefinieerd.

## Definiëren labels

Python 3.11

```

1 # Set the title of the plot
2 plt.title(f'SQM-{location}, lat: {lat}, lon: {lon}', loc='left')
3 plt.title(f'moon: {str(frac)}%, max {round(max(moon_alt),1)}%', loc='right')
4
5 ax.set_xlabel(f'{int(yesterday.strftime("%d"))} {(yesterday.strftime("%b"))} '
6 f'{(yesterday.strftime("%Y"))} → f'{int(date.strftime("%d"))} '
7 f'{(date.strftime("%b"))} {(date.strftime("%Y"))}', loc='left')
8
9 ax.set_ylabel('MSAS')
10 ax2.set_ylabel('Cloud coverage (percentage)')
11 ax3.set_ylabel('Moon altitude (degrees)')
12 ax4.set_ylabel('Temperature (Celsius)')

```

De intervallen op de x-as en y-as worden statisch meegegeven. Voor de y-as zijn dit gewone integers, maar voor de x-as is dit iets lastiger. De waarden op de x-as zijn namelijk datum objecten. Deze worden voor de x-as per uur aangemaakt en tot integer geconverteerd, zodat ze gebruikt kunnen worden als interval (ticks).

## Definiëren x en y ticks

Python 3.11

```

1 # Set the ticks of the x axis
2 start_date = datetime(int(yesterday.strftime("%Y")),
3                         int(yesterday.strftime("%m")),
4                         int(yesterday.strftime("%d")), 17)
5 hours = range(17)
6 x_ticks = [start_date + timedelta(hours=hour) for hour in hours]
7
8 ax.set_xticks(x_ticks)
9 ax.set_xticklabels(['17', '18', '19', '20', '21', '22', '23', '00',
10                     '01', '02', '03', '04', '05', '06', '07', '08', '09'])
11
12 # Set the ticks of the y axis
13 y_ticks = [14, 16, 18, 20, 22]
14 y2_ticks = [0, 25, 50, 75, 100]
15 y3_ticks = [0, 10, 20, 30, 40, 50, 60]
16 y4_ticks = [-10, -5, 0, 5, 10, 15, 20]
17
18 ax.set_yticks(y_ticks)
19 ax2.set_yticks(y2_ticks)
20 ax3.set_yticks(y3_ticks)
21 ax4.set_yticks(y4_ticks)

```

Om elk plot in dezelfde legende weer te geven, worden ze als variabele geplot. Per variabele wordt de lijn en label opgehaald. Deze worden vervolgens in een legenda in de linkerbovenhoek weergegeven.

## Legenda en grid

Python 3.11

```

1 # Add alle the plots to the legend
2 lines, labels = ax.get_legend_handles_labels()
3 lines2, labels2 = ax2.get_legend_handles_labels()
4 lines3, labels3 = ax3.get_legend_handles_labels()
5 lines4, labels4 = ax4.get_legend_handles_labels()
6 ax2.legend(lines + lines2 + lines3 + lines4, labels +
7             labels2 + labels3 + labels4, loc='upper right')
8
9 ax.grid()
10
11 plt.show()

```

De grafiek kan gedownload worden met `plt.savefig`. Aangezien deze standaard een lage resolutie afbeelding produceert, verhogen wij de dpi tot 600. Dit is een vervangende actie voor `plt.show()`.

## Downloaden grafiek

Python 3.11

```
1 plt.savefig(filepath, dpi=600)
```

## E Programmatuur data analyse

De data analyse hebben wij gedaan in Jupyter Notebook. Dit is een programma waarbij je onder andere python code kan schrijven in verschillende cellen. Variabelen en andere definities zijn in de ene cel te definiëren en in de andere cel te gebruiken. Elke cel is wel los van elkaar te starten, en heeft een eigen output.

Gebruikte libraries	Yupyter Notebook
<pre> 1 # Gebruikte libraries 2 import pandas as pd 3 import seaborn as sns 4 import matplotlib.pyplot as plt 5 import numpy as np 6 from sklearn.linear_model import LinearRegression </pre>	

Code figuur 15	Yupyter Notebook
<pre> 1 # Heatmap obv alle locaties (excl. Zwarde Haan) 2 df1 = pd.read_csv("c:/anaconda3/databestanden/all.csv") 3 df1 = df1.loc[:, ~df1.columns.isin(['Location', 'Year', 'Day', 'Moon', 'Cloud_time', 4                                     'Lightpolution'])] 5 df_corr = df1.corr('pearson') 6 mask = np.triu(np.ones_like(df_corr, dtype=bool)) 7 sns.set(style="whitegrid") 8 sns.heatmap(df_corr,mask=mask,annot=True,fmt=".2f", linewidth=0.1, linecolor='white', 9             cmap='coolwarm') 10 plt.title("Correlation matrix - all locations", y=1) </pre>	

Code figuur 17	Yupyter Notebook
<pre> 1 # Boxplots van MSAS voor alle locaties (inc. ZH) 2 df2 = pd.read_csv("c:/anaconda3/databestanden/all_incZH.csv") 3 sns.boxplot(data=df2,x='MSAS',y='Location',fliersize=0.5, 4              order=['ZwartheHaan','Noordpolderzijl','Borkum-Ostland','Sellingen', 5                      'Ameland-Natuurcentrum-Nes','Vlieland-Oost','Schiermonnikoog-dorp', 6                      'Hippolytushoef','Weerribben','Borkum','Akkum','Lochem','DeZilk', 7                      'Groningen-DeHeld','Leiden-Sterrewacht','Delft','Rijswijk']) </pre>	

Code figuur 18	Yupyter Notebook
<pre> 1 # Heatmap obv locaties zonder lichtvervuiling (excl. ZH) 2 df1a = pd.read_csv("c:/anaconda3/databestanden/all_polution_no.csv") 3 df1a = df1a.loc[:, ~df1a.columns.isin(['Location', 'Year', 'Day', 'Moon', 'Cloud_time', 4   'Lightpolution'])] 5 df_corr = df1a.corr('pearson') 6 mask = np.triu(np.ones_like(df_corr, dtype=bool)) 7 sns.set(style="whitegrid") 8 sns.heatmap(df_corr,mask=mask,annot=True,fmt=".2f", linewidth=0.1, linecolor='white', 9             cmap='coolwarm') 10 plt.title("Correlation matrix - without lightpoluted locations", y=1) </pre>	

Code figuur 19

Yupyter Notebook

```

1 # Heatmap obv locaties met lichtvervuiling
2 df1b = pd.read_csv("c:/anaconda3/databestanden/all_polution_yes.csv")
3 df1b = df1b.loc[:, ~df1b.columns.isin(['Location','Year','Day','Moon','Cloud_time',
4                                         'Lightpollution'])]
5 df_corr = df1b.corr('pearson')
6 mask = np.triu(np.ones_like(df_corr, dtype=bool))
7 sns.set(style="whitegrid")
8 sns.heatmap(df_corr,mask=mask,annot=True,fmt=".2f", linewidth=0.1, linecolor='white',
9             cmap='coolwarm')
10 plt.title("Correlation matrix - only lightpoluted locations", y=1)

```

Code figuur 20

Yupyter Notebook

```

1 # LMplots van Moon_fraction en MSAS met onderscheid Lightpollution
2 df5 = pd.read_csv("c:/anaconda3/databestanden/all.csv")
3 sns.set(style="whitegrid")
4 sns.lmplot(x="Moon_fraction",y="MSAS",col="Lightpollution",line_kws={"color":"black"}, 
5             data=df5,markers='o')

```

Code figuur 21

Yupyter Notebook

```

1 # LMplots van Moon_altitude en MSAS met onderscheid Lightpollution
2 df6 = pd.read_csv("c:/anaconda3/databestanden/all.csv")
3 sns.set(style="whitegrid")
4 sns.lmplot(x="Moon_altitude",y="MSAS",col="Lightpollution",line_kws={"color":"black"},data=df6)

```

Code figuur 22

Yupyter Notebook

```

1 # LMplots van Moon_fraction en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid Lightpollution
4 df14 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Moon_fraction',y='MSAS',col='Lightpollution',line_kws={"color":"black"}, 
7             data=df14)

```

Code figuur 23

Yupyter Notebook

```

1 # LMplots van Moon_altitude en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid Lightpollution
4 df15 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Moon_altitude',y='MSAS',col='Lightpollution',
7             line_kws={"color":"black"},data=df15)

```

Code figuur 24 bovenste grafiek

Yupyter Notebook

```

1 # LMplots van Moon_ fraction en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid Moon_ altitude en Lightpolution
4 df28 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution_moon_groups.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Moon_fraction_perc',y='MSAS',hue='Moon_altitude',col='Lightpolution',
7             markers='.',data=df28)

```

Code figuur 24 onderste grafiek

Yupyter Notebook

```

1 # LMplots van Moon_ altitude en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid Moon_ fraction en Lightpolution
4 df29 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution_moon_groups.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Moon_altitude_deg',y='MSAS',hue='Moon_fraction',col='Lightpolution',
7             markers='.',data=df29)

```

Code figuur 25

Yupyter Notebook

```

1 # Heatmap voor Borkum-Ostland
2 df18 = pd.read_csv("c:/anaconda3/databestanden/Borkum-Ostland.csv")
3 df18 = df18.loc[:, ~df18.columns.isin(['Location','Year','Day','Moon','Cloud_time',
4                                         'Lightpolution','Ratio'])]
5 df_corr = df18.corr('pearson')
6 mask = np.triu(np.ones_like(df_corr, dtype=bool))
7 sns.set(style="whitegrid")
8 sns.heatmap(df_corr,mask=mask,annot=True,fmt=".2f", linewidth=0.1, linecolor='white',
9             cmap='coolwarm')
10 plt.title("Correlation matrix - Borkum-Ostland", y=1)

```

Code figuur 26

Yupyter Notebook

```

1 # Heatmap voor Noordpolderzijl
2 df19 = pd.read_csv("c:/anaconda3/databestanden/Noordpolderzijl.csv")
3 df19 = df19.loc[:, ~df19.columns.isin(['Location','Year','Day','Moon','Cloud_time',
4                                         'Lightpolution','Ratio'])]
5 df_corr = df19.corr('pearson')
6 mask = np.triu(np.ones_like(df_corr, dtype=bool))
7 sns.set(style="whitegrid")
8 sns.heatmap(df_corr,mask=mask,annot=True,fmt=".2f", linewidth=0.1, linecolor='white',
9             cmap='coolwarm')
10 plt.title("Correlation matrix - Noordpolderzijl", y=1)

```

Code figuur 27

Yupyter Notebook

```

1 # Heatmap voor Delft
2 df20 = pd.read_csv("c:/anaconda3/databestanden/Delft.csv")
3 df20 = df20.loc[:, ~df20.columns.isin(['Location','Year','Day','Moon','Cloud_time',
4                                         'Lightpolution','Ratio'])]
5 df_corr = df20.corr('pearson')
6 mask = np.triu(np.ones_like(df_corr, dtype=bool))
7 sns.set(style="whitegrid")
8 sns.heatmap(df_corr,mask=mask,annot=True,fmt=".2f", linewidth=0.1, linecolor='white',
9             cmap='coolwarm')
10 plt.title("Correlation matrix - Delft", y=1)

```

Code figuur 28

Yupyter Notebook

```

1 # Heatmap voor Rijswijk
2 df21 = pd.read_csv("c:/anaconda3/databestanden/Rijswijk.csv")
3 df21 = df21.loc[:, ~df21.columns.isin(['Location','Year','Day','Moon','Cloud_time',
4                                         'Lightpolution','Ratio'])]
5 df_corr = df21.corr('pearson')
6 mask = np.triu(np.ones_like(df_corr, dtype=bool))
7 sns.set(style="whitegrid")
8 sns.heatmap(df_corr,mask=mask,annot=True,fmt=".2f", linewidth=0.1, linecolor='white',
9             cmap='coolwarm')
10 plt.title("Correlation matrix - Rijswijk", y=1)

```

Code figuur 29

Yupyter Notebook

```

1 # Boxplots van MSAS per maand voor 2 minst en 2 meest lichtvervuilde
2 # stations met onderscheid Lightpolution
3 df16=pd.read_csv("c:/anaconda3/databestanden/least_most_polution_moon_no.csv")
4 sns.boxplot(data=df16,x='Month',y='MSAS',hue='Lightpolution',fliersize=0.5)

```

Code figuur 30

Yupyter Notebook

```

1 # Boxplots van Clouds_ total per maand voor 2 minst en 2 meest lichtvervuilde
2 # stations met onderscheid Lightpolution
3 df17=pd.read_csv("c:/anaconda3/databestanden/least_most_polution_moon_no.csv")
4 sns.boxplot(data=df17,x='Month',y='Clouds_total',hue='Lightpolution',fliersize=0.5)

```

Code figuur 31

Yupyter Notebook

```

1 # LMplots van Clouds_ total en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid aanwezigheid maan en Lightpolution
4 df10 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Clouds_total',y='MSAS',hue='Moon',col='Lightpolution',
7             line_kws={"color":"black"},data=df10)

```

Code figuur 32

Yupyter Notebook

```

1 # LMplots van Clouds_low en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid aanwezigheid maan en Lightpolution
4 df11 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Clouds_low',y='MSAS',hue='Moon',col='Lightpolution',
7             line_kws={"color": "black"},data=df11)

```

Code figuur 33 bovenste grafiek

Yupyter Notebook

```

1 # LMplots van Clouds_total en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid moon_altitude kwartieren en Lightpolution
4 df26 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution_moon_groups.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Clouds_total',y='MSAS',hue='Moon_altitude',col='Lightpolution',
7             markers='.',data=df24)

```

Code figuur 33 onderste grafiek

Yupyter Notebook

```

1 # LMplots van Clouds_total en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid maan_fraction kwartieren en Lightpolution
4 df24 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution_moon_groups.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Clouds_total',y='MSAS',hue='Moon_fraction',col='Lightpolution',
7             markers='.',data=df24)

```

Code figuur 34

Yupyter Notebook

```

1 # Boxplots van Moon_altitude en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid Moon_altitude kwartieren en Lightpolution
4 df27=pd.read_csv("c:/anaconda3/databestanden/least_most_polution_moon_groups.csv")
5 sns.boxplot(data=df26,x='Moon_altitude',y='MSAS',hue='Lightpolution',fliersize=0.5)

```

Code figuur 35

Yupyter Notebook

```

1 # Boxplots van Moon_fraction en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid moon_fraction kwartieren en Lightpolution
4 df25=pd.read_csv("c:/anaconda3/databestanden/least_most_polution_moon_groups.csv")
5 sns.boxplot(data=df25,x='Moon_fraction',y='MSAS',hue='Lightpolution',fliersize=0.5)

```

Code figuur 36

Yupyter Notebook

```
1 # LMplots van Cloud_mid en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid aanwezigheid maan en Lightpolution
4 df12 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Clouds_mid',y='MSAS',hue='Moon',col='Lightpolution',
7             line_kws={"color":"black"},data=df12)
```

Code figuur 37

Yupyter Notebook

```
1 # LMplots van Clouds_high en MSAS voor 2 minst en 2 meest lichtvervuilde
2 # stations (resp. Borkum-Ostland, Noordpolderzijl, Rijswijk, Delft)
3 # met onderscheid aanwezigheid maan en Lightpolution
4 df13 = pd.read_csv("c:/anaconda3/databestanden/least_most_polution.csv")
5 sns.set(style="whitegrid")
6 sns.lmplot(x='Clouds_high',y='MSAS',hue='Moon',col='Lightpolution',
7             line_kws={"color":"black"},data=df13)
```

## Code regressie model 1

Yupyter Notebook

```

1 # Enkelvoudige linear regression obv x = Clouds_total en y = MSAS en alleen
2 # Delft en Rijswijk. Geeft een bijna even hoge R2 zonder dat je Ratio nodig hebt.
3 df9b = pd.read_csv("c:/anaconda3/databestanden/most_polution.csv")
4 X = df9b.drop(['Location','Year','Month','Day','Moon','Cloud_time','MSAS',
5                 'Clouds_low','Clouds_mid','Clouds_high','Lightpolution','Temperature',
6                 'Moon_fraction','Moon_altitude','Ratio'],axis=1)
7 y = df9b['MSAS']
8 model = LinearRegression()
9 model.fit(X,y)
10 print(f"coefficient of determination R2: {model.score(X,y)}")
11 print(f"intercept: {model.intercept_}")
12 print(f"coefficients: {model.coef_}")

```

## Code regressie model 2

Yupyter Notebook

```

1 # Enkelvoudige linear regression obv y = MSAS en x = clouds_total en voor
2 # gemiddeld lichtvervuilde stations (excl. Delft,Rijswijk)
3 df9d = pd.read_csv("c:/anaconda3/databestanden/medium_polution.csv")
4 X = df9d.drop(['Location','MSAS','Year','Month','Day','Moon','Cloud_time',
5                 'Clouds_low','Clouds_mid','Clouds_high','Lightpolution','Temperature',
6                 'Moon_fraction','Moon_altitude','Ratio'],axis=1)
7 y = df9d['MSAS']
8 model = LinearRegression()
9 model.fit(X,y)
10 print(f"coefficient of determination R2: {model.score(X,y)}")
11 print(f"intercept: {model.intercept_}")
12 print(f"coefficients: {model.coef_}")

```

## Code uiteindelijk regressie model

Yupyter Notebook

```

1 # Enkelvoudige linear regression obv x = MSAS en y = clouds_total en alleen
2 # Delft en Rijswijk. Geeft een bijna even hoge R2 zonder dat je Ratio nodig hebt.
3 df9b = pd.read_csv("c:/anaconda3/databestanden/most_polution.csv")
4 X = df9b.drop(['Location','Year','Month','Day','Moon','Cloud_time','Clouds_total',
5                 'Clouds_low','Clouds_mid','Clouds_high','Lightpolution','Temperature',
6                 'Moon_fraction','Moon_altitude','Ratio'],axis=1)
7 y = df9b['Clouds_total']
8 model = LinearRegression()
9 model.fit(X,y)
10 print(f"coefficient of determination R2: {model.score(X,y)}")
11 print(f"intercept: {model.intercept_}")
12 print(f"coefficients: {model.coef_}")

```

## F Programmatuur foto's

Met Python kan de helderheid van een foto op meerdere manieren berekend worden. Hierbij maken we gebruik van de library Pillow, een foto bewerkings tool. Manier 1 en 2 zijn gebaseerd op een greyscale, de ander zijn gebaseerd op de HSV van Finley. Alle manieren zijn bedacht door Thierry Lathuille, als toepassing van het werk van Finley.

Installing Pillow

Terminal

```
Pip install Pillow
```

1. Convert image to greyscale return average pixel brightness.

Python 3.11

```
1 def brightness( im_file ):
2     im = Image.open(im_file).convert('L')
3     stat = ImageStat.Stat(im)
4     return stat.mean[0]
```

2. Convert image to greyscale return RMS pixel brightness.

Python 3.11

```
1 def brightness( im_file ):
2     im = Image.open(im_file).convert('L')
3     stat = ImageStat.Stat(im)
4     return stat.rms[0]
```

3. Average pixels then transform to "perceived brightness".

Python 3.11

```
1 def brightness( im_file ):
2     im = Image.open(im_file)
3     stat = ImageStat.Stat(im)
4     r,g,b = stat.mean
5     return math.sqrt(0.241*(r**2) + 0.691*(g**2) + 0.068*(b**2))
```

4. RMS of pixels then transform to "perceived brightness".

Python 3.11

```
1 def brightness( im_file ):
2     im = Image.open(im_file)
3     stat = ImageStat.Stat(im)
4     r,g,b = stat.rms
5     return math.sqrt(0.241*(r**2) + 0.691*(g**2) + 0.068*(b**2))
```

5. Calculate "perceived brightness" of pixels then return average.

Python 3.11

```
1 def brightness( im_file ):
2     im = Image.open(im_file)
3     stat = ImageStat.Stat(im)
4     gs = (math.sqrt(0.241*(r**2) + 0.691*(g**2) + 0.068*(b**2))
5           for r,g,b in im.getdata())
6     return sum(gs)/stat.count[0]
```

## F1 Verwerking foto's

Voor het plotten van de data, of het vergelijken ervan, moet je ook de tijd weten van de genomen foto. In plaats van deze handmatig in te vullen, wordt deze door de camera automatisch aan elke foto meegegeven via zogenaamde Exif data.

Exif staat voor Exchangeable image file format, en zijn eigenlijk gewoon gegevens die de camera aan de foto meegeeft. Deze kan je meestal alleen met bepaalde software zien. Voorbeelden van Exif data zijn: tijd, locatie en camera-instellingen.

In Python kunnen wij de Exif data van een foto naar een variabele schrijven. De Exif code voor onze bestanden is '36867'. Met de Python datetime library kunnen we deze datum en tijd formatteren.

Datum foto via Exif

Python 3.11

```

1 def get_date_taken(path):
2     exif = Image.open(path)._getexif()
3     if not exif:
4         raise Exception('Image {} does not have EXIF data.'.format(path))
5     date_taken = exif[36867]
6
7     date_object = datetime.strptime(date_taken, '%Y:%m:%d %H:%M:%S')
8
9     # Return a tuple containing the hour and minute parts
10    return date_object.strftime('%H:%M') # Format as hours:minutes

```

Met de Python `glob.glob()` functie kunnen wij per foto in een bepaalde bestandsmap de datum-/tijd en de lichtsterkte schrijven naar een list. Deze lists kunnen vervolgens geplot worden in een grafiek, maar ook later met elkaar worden vergeleken.

Automatisering

Python 3.11

```

1 def analyse_image():
2     time_list = []
3     brightness_list = []
4
5     for i in glob.glob('C:\\\\Users\\\\Tim\\\\Desktop\\\\Code\\\\Python\\\\Weg Melkweg\\\\2h test\\\\*.JPG'):
6         brightness_list.append(brightness(i))
7         time_list.append(get_date_taken(i))
8
9     return brightness_list ,time_list

```

## Literatuur

- [1] De Miguel, A. S., Bennie, J., Rosenfeld, E., Dzurjak, S. A., Gaston, K. J. (2021). First estimation of global trends in nocturnal power emissions reveals acceleration of light pollution. *Remote Sensing*, 13(16), 3311. <https://doi.org/10.3390/rs13163311>
- [2] Over 20 jaar mogelijk geen sterren meer te zien door toename lichtvervuiling. (2023, May 31). RTL Nieuws. <https://www.rtlnieuws.nl/nieuws/nederland/artikel/5387717/lichtvervuiling-nederland-sterren-sterrenhemel>
- [3] Natuur, O. (n.d.). Lichtvervuiling veroorzaakt een domino-effect in onze natuur. Onze Natuur. <https://www.onzenatuur.be/artikel/lichtvervuiling-veroorzaakt-een-domino-effect-in-onze-natuur>
- [4] Lichtvervuiling maakt Merels minder vruchtbaar. (n.d.). Natuurpunt. <https://www.natuurpunkt.be/nieuws/lichtvervuiling-maakt-merels-minder-vruchtbaar-20140131>
- [5] Wedzinga, M. (2023, May 11). De donkere kant van licht: hoe lichtvervuiling mens en natuur beïnvloedt. takeaware.nl. <https://takeaware.nl/blogs/nieuws/de-donkere-kant-van-licht-hoe-lichtvervuiling-mens-en-natuur-beinvloedt>
- [6] De Sagher, A. (2021, December 29). Doof het licht voor meer dier- en sterplezier. EOS Wetenschap. <https://www.eoswetenschap.eu/natuur-milieu/doof-het-licht-voor-meer-dier-en-sterplezier>
- [7] Sky Quality Meter-LU. (n.d.). <http://unihedron.com/projects/sqm-lu/>
- [8] Wikipedia contributors. (2024, February 9). Surface brightness. Wikipedia. [https://en.wikipedia.org/wiki/Surface\\_brightness](https://en.wikipedia.org/wiki/Surface_brightness)
- [9] Sky Quality Meter - FAQ. (n.d.). <http://unihedron.com/projects/darksky/faqsqmlu.php>
- [10] mireianievas/PySQM: Python client for the Sky Quality Meter. (n.d.). GitHub. <https://github.com/mireianievas/PySQM>
- [11] Theo Jurriens. (2020, March 17). Rijksuniversiteit Groningen. [https://www.rug.nl/scien\\_celinx/overons/scienclinxteam/contactscienclinx/theo-jurriens](https://www.rug.nl/scien_celinx/overons/scienclinxteam/contactscienclinx/theo-jurriens)
- [12] Cavazzani, S., Ortolani, S., Bertolo, A., Binotto, R., Fiorentin, P., Carraro, G., Saviane, I., Zitelli. (2020). Sky Quality Meter and satellite correlation for night cloud-cover analysis at astronomical sites. *Monthly Notices of the Royal Astronomical Society*, 493(2), 2463–2471. <https://doi.org/10.1093/mnras/staa416>

- [13] Massetti, L., Materassi, A., Sabatini, F. (2023). NSKY-CD: a system for cloud detection based on night sky brightness and sky temperature. *Remote Sensing*, 15(12), 3063. <https://doi.org/10.3390/rs15123063>
- [14] Wikipedia-bijdragers. (2022, August 24). Bimodale verdeling. Wikipedia. [https://nl.wikipedia.org/wiki/Bimodale\\_verdeling](https://nl.wikipedia.org/wiki/Bimodale_verdeling)
- [15] Lolkema, D. E., Haaima, M., Outer, P. N. D., Spoelstra, H. (2011). Effects of meteorological and atmospheric parameters on night sky brightness — RIVM. (n.d.). <https://www.rivm.nl/publicaties/effects-of-meteorological-and-atmospheric-parameters-on-night-sky-brightness>
- [16] Lorenz. D. Light Pollution Atlas 2006, 2016, 2020 and 2022. (n.d.). <https://djlorenz.github.io/astronomy/lp2020/overlay/dark.html>
- [17] Light Pollution Atlas 2020. (n.d.). <https://djlorenz.github.io/astronomy/lp2020/>
- [18] Langer en korter worden van de dagen, hoe werkt het? (n.d.). InfoNu. <https://wetenschap.infonu.nl/natuurverschijnselen/173771-langer-en-korter-worden-van-de-dagen-hoe-werkt-het.html>
- [19] Wanneer is het nacht? (n.d.). RIVM. <https://www.rivm.nl/wanneer-is-nacht>
- [20] open-meteo/open-meteo: Free Weather Forecast API for non-commercial use. (n.d.). GitHub. <https://github.com/open-meteo/open-meteo>
- [21] Skyfield — documentation. (n.d.). <https://rhodesmill.org/skyfield/>
- [22] Minor. D. dminor/almanac: Python module for calculations of sun and moon rise and set. (n.d.). GitHub. <https://github.com/dminor/almanac>
- [23] pandas - Python Data Analysis Library. (n.d.). <https://pandas.pydata.org/>
- [24] Wikipedia contributors. (2024, February 15). Pearson correlation coefficient. Wikipedia. [https://en.wikipedia.org/wiki/Pearson\\_correlation\\_coefficient](https://en.wikipedia.org/wiki/Pearson_correlation_coefficient)
- [25] Admin. (2022, August 29). Correlation - Correlation Coefficient, Types, Formulas amp; Example. BYJUS. <https://byjus.com/math/pearson-correlation/>
- [26] Team, D. S. (2020, December 15). Wat is een correlatiematrix? DATA SCIENCE. <https://datascience.eu/nl/wiskunde-statistiek/wat-is-een-correlatiematrix/>
- [27] Merkus, J. (2023, February 10). Boxplot maken en aflezen — Stappenplan Voorbeelden. Scribbr. <https://www.scribbr.nl/statistiek/boxplot/>

- [28] Sky Quality Meter - FAQ. (n.d.). <http://www.unihedron.com/projects/darksky/faqsqmludl.php>
- [29] Wikipedia contributors. (2023, December 20). Analysis of covariance. Wikipedia. [https://en.wikipedia.org/wiki/Analysis\\_of\\_covariance](https://en.wikipedia.org/wiki/Analysis_of_covariance)
- [30] Historical Weather API — Open-Meteo.com. (n.d.). <https://open-meteo.com/en/docs/historical-weather-api>
- [31] Van Heijst, L. (2023, March 9). Regressieanalyse uitvoeren en interpreteren. Scribbr. <https://www.scribbr.nl/statistiek/regressieanalyse/>
- [32] Turney, S. (2023, March 8). Coefficient of determination (R2) — Betekenis Voorbeelden. Scribbr. <https://www.scribbr.nl/statistiek/determinatiecoefficient/>
- [33] Wikipedia-bijdragers. (2023, May 15). Regressieanalyse. Wikipedia. <https://nl.wikipedia.org/wiki/Regressieanalyse>
- [34] Dobovizki, N. (2008, April 27). Calculating the perceived brightness of a color. <https://www.nbdtech.com/Blog/archive/2008/04/27/Calculating-the-Perceived-Brightness-of-a-Color.aspx>
- [35] Finley, D. (2006). HSP Color Model - alternative to HSV (HSB) and HSL. (n.d.). <https://alienryderflex.com/hsp.html>
- [36] Wikipedia contributors. (2024, January 10). HSL and HSV. Wikipedia. [https://en.wikipedia.org/wiki/HSL\\_and\\_HSV](https://en.wikipedia.org/wiki/HSL_and_HSV)
- [37] Raspberry Pi Documentation - camera. (n.d.). <https://www.raspberrypi.com/documentation/accessories/camera.html>
- [38] Maanopkomst en maanondergang - Maankalender. (n.d.). <https://www.zonsopgangzonsondergang.nl/maanstanden-maanopkomst/>
- [39] ADAGUC viewer. (n.d.). [https://msgcpp.knmi.nl/adaguc-viewer/index.html?srs=EPSG:4326&bbox=-95,-80,95,80&service=https%3A%2F%2Fmsgcpp-adaguc.knmi.nl%2Fadaguc-server%3Fdataset%3Dmsgrt&request=WMS&layer=atmosphere\\_optical\\_thickness\\_due\\_to\\_cloud](https://msgcpp.knmi.nl/adaguc-viewer/index.html?srs=EPSG:4326&bbox=-95,-80,95,80&service=https%3A%2F%2Fmsgcpp-adaguc.knmi.nl%2Fadaguc-server%3Fdataset%3Dmsgrt&request=WMS&layer=atmosphere_optical_thickness_due_to_cloud)

