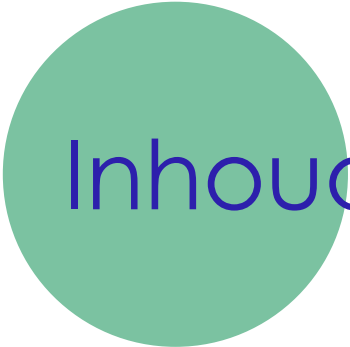


Datapilot NEAR GPS bestemmingsdata

De weg van data naar inzicht



landelijke
data
alliantie



Inhoud

1. Introductie	3
2. Achtergronden NEAR data	5
APP DATA EN MAANDELIJKS ACTIEVE GEBRUIKERS	5
REGISTRATIE VAN GPS DATA EN DE NAUWKEURIGHEID.	5
3. Uitvoering van de datapilot	8
DATA-AANKOOP, PREPARATIE EN ANALYSE	8
VERKENNING DATA OP HEXAGONNIVEAU 9 MET DREMPELWAARDE VAN 10 SIGNALLEN PER UUR	8
VERKENNING NEAR VISTA / PINNACLE DASHBOARD,	9
VERKENNING DATA HEXAGONNIVEAU 6 & 8	9
IMPRESSIES VAN DE MOGELIJKHEDEN MET NEAR'S PIN DATA	15
4.Reflectie op de pilot	16
REFLECTIE OP DE UITKOMSTEN	16
REFLECTIE OP DE BEPERKINGEN VAN DE DATA	16
REFLECTIE OP HET GEBRUIK VAN DE DATA: TIPS & ADVIEZEN	16
REFLECTIE OP DE MOGELIJKE TOEPASSINGEN VAN DE DATA:	16
VERVOLG STAPPEN EN MOGELIJK VERVOLGONDERZOEK	17
Bijlage 1: technische verantwoording	18

1. Introductie

Deze rapportage doet verslag van de projectuitkomsten waarbij data van de firma NEAR is gebruikt. Ten eerste, omdat de unieke mogelijkheid zich aandoet om voor heel Nederland een dataset te verkrijgen en te analyseren. Ten tweede, omdat de dataset zowel data bevat over Nederlanders als over buitenlandse bezoekers (inkomende bezoekers). Ten derde, omdat de data als 'data dump' beschikbaar is, waardoor data analyse in eigen beheer goed mogelijk is. Tot slot, de leverancier garandeert dat de data AVG / GDPR gekeurd is.

Bewegings- en herkomstdata van bezoekers op basis van GPS (Global Positioning Systems) data, bieden relatief nieuwe mogelijkheden om een beter beeld te krijgen van toeristen en recreanten in termen van aantallen, herkomst, bewegingen in tijd en ruimte, druktebeelden/intensiteiten, spreiding in tijd en ruimte.

Dit is echter een veronderstelling. Want, hoewel de data beschikbaar is, is de bruikbaarheid van de data op grote schaal om bezoeker- en toerismestromen in beeld te brengen nog niet integraal onderzocht. Er zijn momenteel enkele aanbieders op de markt die GPS-data aanbieden, waarvan de twee bekendste wellicht zijn: Resono en NEAR Intelligence (nu Azira). Resono biedt data aan van Nederlanders. NEAR biedt data aan van Nederlanders en buitenlanders (inkomende bezoekers). Er zijn reeds diverse pilots gaande met Resono data.

Doel van het project

Het doel van het project is om de 'GPS data' van de firma NEAR Intelligence (nu Azira) te gebruiken om 1.) herkomst van bezoekers te kunnen bepalen en 2.) druktebeelden te creëren ten behoeve van inzichten die relevant zijn voor spreiding in tijd en ruimte. Voorafgaand is de betrouwbaarheid en bruikbaarheid van de data onderzocht.

In de aanloop naar het project heeft het European Tourism Futures Institute (ETFI) van NHL Stenden Hogeschool samengewerkt met amsterdam&partners, Toerisme Veluwe Arnhem Nijmegen (TVAN) en het Nederlands Bureau voor Toerisme & Congressen (NBTC). Al deze partijen zoeken continue naar nieuwe mogelijkheden en betere inzichten ten aanzien van toerisme en recreatie. In hun zoektocht zijn alle partijen op het spoor van GPS data gekomen, wat hen ook heeft samengebracht. In gezamenlijkheid hebben deze partijen gediscussieerd over de potentie van de GPS data van NEAR en hoe tot een uitvoerbaar project te komen, waar vervolgens de projectdoelen uit zijn voortgekomen.

Projectuitvoering

De uitvoering lag bij een projectteam bestaande uit toerisme-experts van het European Tourism Futures Institute (ETFI) van NHL Stenden Hogeschool en medewerkers van DataFryslan, een coöperatie van publieke partijen in Friesland, gericht op data science.

Er is gewerkt in SCRUM sprints, ondersteund door wekelijkse bijeenkomsten. De focus lag op het begrijpen, analyseren en presenteren van de NEAR data. De opgedane kennis en inzichten tijdens het proces, zijn de basis voor de verslaglegging. Dit rapport beschrijft de structuur en inhoud van de NEAR data. Daarnaast wordt ingegaan op de gebruikerservaringen van andere onderzoekers en hun toepassing en gebruik van de data.

NHL Stenden / European Tourism Futures Institute (ETFI)

Dr. Stefan Hartman - projectleider

Jörg Wenzel - projectleider

DataFryslan

Dr. Carlos de Matos Fernandes: projectleider vanaf februari 2024 / data analist

Bas van der Meulen: projectleider 1e fase tot januari 2024

Taeke van der Laan - Data-analist / databank specialist, Gemeente Súdwest Fryslân

Oswald Dijkstra - Data analist / databank specialist, Gemeente Leeuwarden

Sigrid Andela - Data analist / databank specialist, Gemeente Leeuwarden

Hang Zheng - Data analist / databank specialist, Gemeente Leeuwarden

Marcel Dekker,- Data analist / databank specialist / GIS specialist, Elkien

Het project is uitgevoerd in de periode november 2023 tot april 2024. De eerste drie maanden waren nodig om tot een bruikbare dataset te komen. De laatste maand werd benut om de data toegankelijk te maken en de prototype informatie tools te ontwikkelen. De 'bruikbaarheidsslag' van de prototype dashboards kan in een vervolgproject en in samenspraak met gebruikers verbeterd worden.

2. Achtergronden NEAR data

De data die NEAR aanbiedt, bestaat uit GPS bewegingsdata die door verschillende applicaties (apps) op de mobiele telefoons van gebruikers verzameld wordt. Hier geven de gebruikers toestemming voor via de 'end user licence agreement' (EULA) die bij het installeren van een app goedgekeurd moet worden. De data koopt NEAR van de aanbieders van deze apps. In totaal gaat het om duizenden apps.

App data en maandelijks actieve gebruikers

Het is belangrijk om te weten van welke app aanbieders NEAR data gekocht heeft om te kunnen bepalen of deze voor de desbetreffende markt relevant is. Dit is echter maar ten dele bekend (NEAR levert een longlist van apps) en gebruikers worden niet procentueel bekend gemaakt per app. Een aantal voorbeelden van apps waar bewegingsdata voor de Nederlandse markt vandaan komt zijn: "9292 reisplanner OV + e-ticket", "De Morgen: nieuws & duiding", "PZC nieuws", "Dutch English Dictionary & Translator", "De Stentor Nieuws", "NL Train Navigator: Dutch train planner & transit", "AD – Nieuws & Sport", "Dutch Study Bible"

Naast de soorten apps die data verzameld hebben is het van belang om te weten hoeveel gebruikers data hebben bijgedragen. De zogenoemde *user engagement data* betreft data die op maandelijks niveau gegenereerd wordt en neerkomt op een getal van maandelijks actieve gebruikers van alle apps waar NEAR de data van gekocht heeft. Deze 'Monthly Active Users' (MAU) overzicht geeft aan hoe groot de steekproef is voor een bepaald land.

NL Populatie	Oct/22	Nov/22	Dec/22	Jan/23	Feb/23	Mar/23	Apr/23	May/23	Jun/23	Jul/23
01/04/2021	33%	35%	33%	32%	22%	22%	18%	14%	14%	15%
17530000	5795668	6137152	5716241	5558175	3823669	3832065	3099642	2513535	2482265	2630291

Table 1: MAU data NL voor dataset 10/22-7/23

De data is dus een steekproef van een totale populatie die apps gebruikt die GPS positiedata registreert. Hoeveel gebruikers van specifieke apps dat precies zijn kan per maand verschillen en is niet bekend in absolute getallen of verhoudingen. Dat is een nadeel omdat de kennis van absolute bijdrage van een bepaalde app ook naar een bepaald gebruikersprofiel kan verwijzen. Dat de verhouding van gebruikers per app niet door NEAR gecommuniceerd wordt is een nadeel omdat daardoor open blijft of de data representatief is voor het gedrag van de hele populatie.

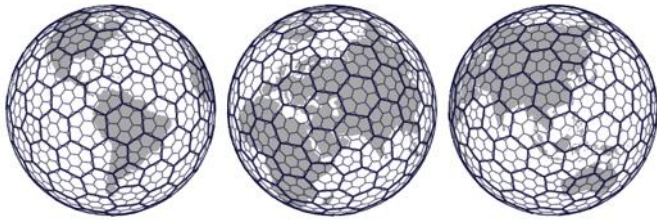
Lessons learned: Het is relevant om te kunnen bepalen welke marktsegmenten in de data vertegenwoordigd zijn door de exacte verdeling van de MAU op app-niveau te achterhalen.

Registratie van GPS data en de nauwkeurigheid.

De gemiddelde horizontale positienauwkeurigheid van GPS data ligt tussen de 7 en 13 meter. Dus verplaatsingen onder de 7 meter worden mogelijk niet geregistreerd. Onzekere datapunten zullen een grote horizontale nauwkeurigheidswaarde hebben en worden verwijderd. Als een apparaat stilstaat, wordt er geen locatie opgenomen. Dit betekent dat er geen onnodige data wordt verzameld als een telefoon bijvoorbeeld 's nachts op een nachtkastje ligt. De GPS-data wordt verzameld met behulp van een interval timing methode, zoals het verzamelen van data van de telefoon elke 10 minuten. De data die wij ontvangen is hierop gebaseerd maar ingedeeld in tijdsvakken van een uur en de Uber H3 ruimtelijke indeling.

Ruimtelijke weergave en resolutie via Uber H3 methode

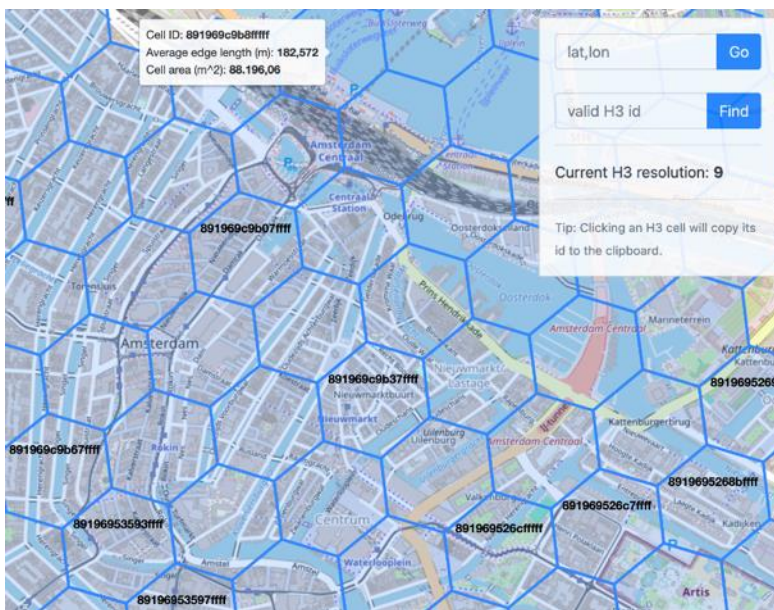
Voor de ruimtelijke weergave maakt NEAR gebruik van de door het bedrijf Uber geïntroduceerde hiërarchische indeling van het oppervlak van de aarde in polygonen met name hexa- en pentagonen¹.



Afbeelding 1: H3 Uber's Hexagonal Hierarchical Spatial Index²

Polygoon niveau 0 bevat ongeveer het hele middellandse zeegebied, niveau 1 is meer gedetailleerd (bevat ongeveer heel Spanje) en niveau 15 zoomt in tot het hoogste detailniveau van een halve vierkantmeter en bevat ongeveer het oppervlak van een parkeerautomaat.

Om binnen een stad, regio of provincie bezoekersstromen te kunnen identificeren is in eerste instantie niveau 9 gekozen voor een eerste test. Niveau 9 polygonen hebben een doorsnee van ongeveer 300 meter.³ Om een zo klein mogelijk schaalniveau te krijgen is er in eerste instantie voor niveau 9 gekozen. Ook om binnen een stad verplaatsingen te kunnen identificeren. Dit niveau leek een goed compromis te zijn om detailverplaatsing te kunnen bepalen voor drukke kernen maar wel nog steeds verplaatsingen te herkennen in buitengebieden.



Afbeelding 2: H3 niveau 9 op het centrum van Amsterdam⁴

Het detailniveau is belangrijk omdat hoe groter de het oppervlak van de polygonen wordt, de meer bezoekers erin geteld worden en de minder nauwkeurig te zien is hoe de bezoekers zich verplaatsten. Tegelijkertijd wordt onduidelijker waar precies de bezoekers vandaan komen. Om zoveel mogelijk belanghebbende met relevante bezoekersinformatie te dienen is de keuze van detailniveau uiterst belangrijk. Een laag polygoonnummer betekent een groter polygoonoppervlak.

Lessons learned: Het doel van het onderzoek moet bepalen welke polygoongrootte het meest geschikt is.

¹ Omdat een spherische weergave niet mogelijk is met de alleen hexagonen, bevat ieder niveau ook 12 pentagonen. Omdat er veel meer hexagonen zijn richten wij de focus alleen op de zeskantige polygonen.

² <https://www.uber.com/en-NL/blog/h3/>

³ <https://h3geo.org/docs/core-library/restable/>

⁴ <https://wolf-h3-viewer.glitch.me/>

De bewegingsdata van NEAR bevat een PIN en een CEL rapport.

De PIN data bevat naast het polygoon nummer (het desbetreffende polygoon op het desbetreffende detailniveau) het tijdvak van de meeting en de hoeveelheid bezoekers die vanuit een andere CEL binnen het tijdvak 'erbij' zijn gekomen.

PIN Report Schema

Polygon ID (Uber H3 Level 8)	Date	Hour	Visitors
------------------------------	------	------	----------

Tabel 2: PIN Report schema

Het CEL rapport bevat naast het bezochte polygoon nummer, de herkomst van de bezoeker. Deze wordt nader gespecificeerd door het nummer van het polygoon waar de bezoeker hoogstwaarschijnlijk vandaan komt. Dat wordt vastgesteld door te kijken waar zich een bezoeker meestal op een avond bevindt, dus de woonplek. Deze wordt nader aangegeven door postcode en de gemeente waarin zich de postcode bevindt en uiteindelijk het tijdstip van bezoek.

CEL Report Schema

Polygon ID (Uber H3 Level 8)	Common Evening (Uber H3 Level 8]	Common Evening Country	Common Evening Admin	Common Evening Micro	Common Evening Postal	Common Evening Municipality	Visit Date	Number of Visitors
------------------------------	----------------------------------	------------------------	----------------------	----------------------	-----------------------	-----------------------------	------------	--------------------

Tabel 3: CEL Report schema

Niveau 8 heeft een doorsnee vergeleken met de afstand tussen het Centraal Station van Amsterdam tot de Dam. Het gehele centrum van Amsterdam binnen de Prinsengracht is op niveau 8 opgedeeld in 7 hexagonalen. Dit lijkt een logische en relevante keuze om ook binnen een stad bewegingen van bezoekers te onderzoeken.

Privacy, AVG (GDPR) en de 'drempelwaarde'

NEAR gebruikt een zogenoemde drempelwaarde (threshold) voor het 'versluieren' van individuen in het dataset zodat er geen individuele bewegingen getraceerd kunnen worden. Dat werkt al volgt: Als er meer dan een x aantal bezoekers uit een bepaald polygoon een ander polygoon tijdens een uur/dag/jaar bezoekt, dan wordt deze bezoeker en zijn herkomst getoond. Met de drempelwaarde (x) wordt dus bepaald wanneer de herkomst van de bezoeker in de CEL data verschijnt. De drempelwaarde blijkt uiterst belangrijk te zijn als het gaat om de hoeveelheid bezoekers die daadwerkelijk in het dataset voorkomen. Als op niveau 9 de drempelwaarde 10 per uur is, dan worden alleen bezoekers getoond die vanuit dezelfde polygoon een ander polygoon bezoeken binnen dat uur. Mensen uit een wijk die naar een winkelcentrum in de buurt gaan worden dan wel getoond, maar iemand uit Zeeland die op visite gaat bij vrienden in Rotterdam wordt dan niet getoond. Als de drempelwaarde 1 is per jaar, dan wordt deze bezoeker wel getoond. De drempelwaarde heeft tegelijkertijd de functie om te voorkomen dat de bewegingen van een individu binnen de dataset getraceerd kan worden.

Tijdens de preparatie van de data bleek de eerste dataset die wij van NEAR hebben ontvangen op niveau 9 een drempelwaarde van 10 per dag (CEL) PIN per uur te hebben. Hierdoor waren lokale bewegingen van grote groepen binnen steden te traceren, maar geen enkele relevante bezoekersinformatie te genereren. Bewegingen die in de dataset waar te nemen waren, betroffen vooral bewegingen tussen direct aan elkaar grenzende buurt polygoonen.

Lessons learned: Advies drempelwaarde 10 is niet geschikt voor fijnmazige polygoonniveaus.

Geschiedenis en ervaring van andere NEAR data gebruikers

Zoals hierboven genoemd, bleek de drempelwaarde een reden dat de dataset voor onze doeleinden ongeschikt was. Daardoor zijn een aantal interviews gevoerd met een drietal gebruikers van NEAR data. Uit deze interviews kwam naar voren dat de data die in 2019 aan een onderzoeker verstrekt zijn een veel hoger detailniveau kende. Bewegingen van individuen konden getraceerd worden. Dit was privacy technisch niet acceptabel en volgens de huidige AVG/GDPR normen niet meer mogelijk. Daarnaast werd duidelijk dat twee andere onderzoeksbureaus die met NEAR data werken alleen de zogenoemde VISTA dashboard data gebruiken. Dat zijn bewegingsdata rond of op vooraf gespecificeerde plekken of gebieden. Behalve experimentele toepassingen is er dus geen gebruik gemaakt van de ruwe PIN en CEL data voor wat betreft het managen van bestemmingen.

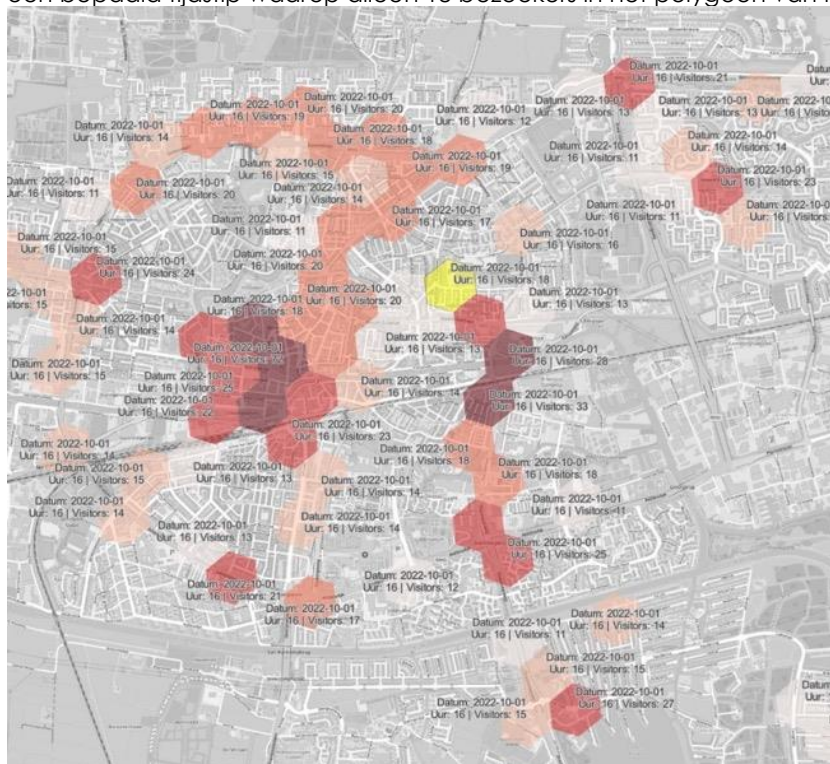
3. Uitvoering van de datapilot

Data-aankoop, preparatie en analyse

Vanwege de ontbrekende ervaring met de aangeboden NEAR data blijken de stappen van aankoop en voorbereiding van de data sterk samen te hangen met de analyse van de ontvangen data. Om het proces te verduidelijken wordt het tijdsplan en de verschillende datasets die ontvangen zijn door NEAR nader toegelicht.

Verkenning data op hexagonniveau 9 met drempelwaarde van 10 signalen per uur

De eerste dataset is medio november 2023 geleverd. De data is door het projectteam onderzocht door simpele en geavanceerde steekproeven uit te voeren. Voorbeelden als een voetbalwedstrijd van SC Cambuur Leeuwarden op een bepaald tijdstip waarop alleen 18 bezoekers in het polygoon van het stadion te vinden waren.



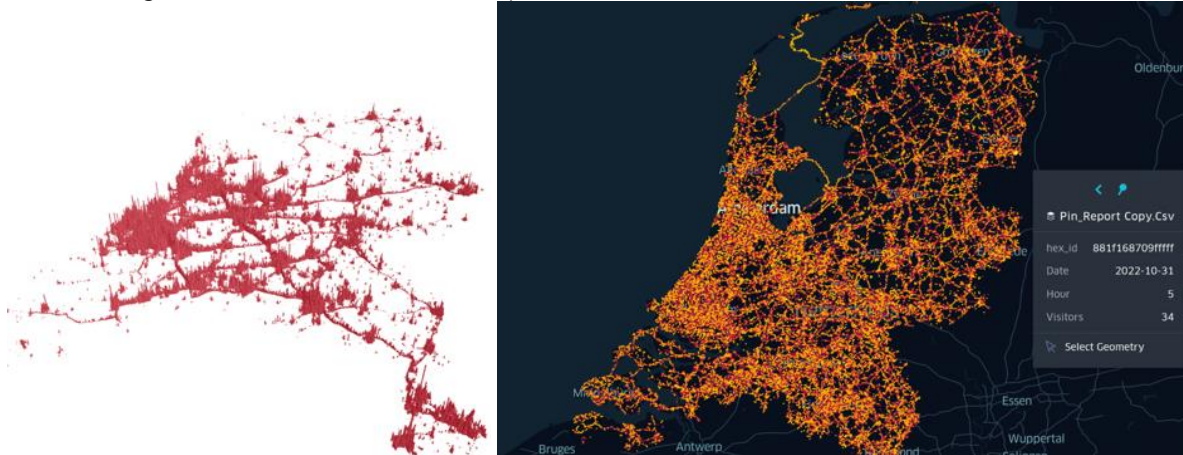
Afbeelding 3: Drukbeeld Leeuwarden tijdens een Voetbalwedstrijd van Cambuur

Het drukste polygoon op die dag rond vier uur 's middags bleek station Utrecht met 218 bezoekers. Beide indicaties waren onjuist. Daarnaast bleek de correlatie tussen polygoonen vooral in aangrenzende polygoonen te vinden. Dus beweging vond plaats rond de woonplaats. Bezoekers van elders waren er amper te vinden en de set bevatte geen buitenlandse bezoekers. Vrij snel kon de conclusie getrokken worden dat de ontvangen data niet geschikt was om representatieve druktebeelden te genereren nog bewegingen van bezoekers te traceren en/of de herkomst binnen Nederland aan te tonen.

Lesson learned: Simpele toeristisch-recreatieve bewegingen konden in deze eerste dataset niet gevonden worden.

Lessons learned: Inzicht hebben in hoe data gegenereerd wordt en hoe variabelen tot stand komen zoals de common evening place, dus hoe de woonplaats wordt bepaald, is noodzakelijk om resultaten te kunnen beoordelen.

Eerste weergaven van de druktebeelden op basis van de eerste data:



Afbeelding 4: Experimentele dataweergave QGIS (links) en Kepler GL (rechts)

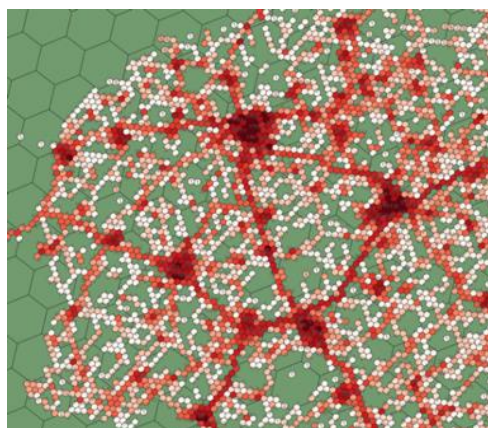
Data van detailniveau 8 en een drempelwaarde van 10 van eind November 2023; er klopt iets niet. Op basis van de eerste ervaring met het niveau 9 dataset en gesprekken met andere gebruikers is naar NEAR teruggekoppeld dat de geleverde data onbruikbaar is en verzocht een voorbeeld van een niveau 8 dataset te sturen. De set bevatte 95% meer data terwijl er minder data verwacht werd en er werd de helft van de dag weergegeven, de resterende uren ontbraken. Met de verzamelde inzichten en gedeelde ervaringen van medegebruikers (Forward Keys, Wonderful Copenhagen en een onderzoek door dr. Roman Egger) is de wens naar NEAR gecommuniceerd om samen met de productie engineers te kijken waar wel interessante data te vinden is. Eigenlijk de vraag: welk niveau en welke drempelwaarde genereert interessante bewegings- en drukte data? Tegelijkertijd werd duidelijk dat de bedrijfsonderdelen van NEAR die in de VS zijn geregistreerd voor een "chapter 11 bankruptcy" staan i.v.m. een gekelderde beurskoers. Dit is een doorstart-procedure die op de korte termijn vermoedelijk niet van veel invloed zou zijn omdat deze procedure maanden/jaren in beslag kan nemen. Op de lange termijn kan dit wel effect hebben op databeschikbaarheid. Ook dit werd onderdeel van het gesprek medio december. De uitkomst was de toezegging vanuit NEAR nog eens naar de geleverde data te kijken en ons voorbeelden van verschillende niveaus te sturen zodat wij een gegronde keuze kunnen maken.

Verkenning NEAR VISTA / Pinnacle Dashboard

Uit gesprekken gehouden in december 2023 met NEAR datagebruikers Forward Keys en Wonderful Copenhagen bleek het Dashboard product van NEAR interessant om te verkennen. Via een proefaccount en een beperkt aantal aanvragen is hierna gekeken. Via een kaart kan een gebied gekozen worden en de output is de drukte en herkomstdata voor een gekozen periode.

Verkenning data hexagonniveau 6 & 8

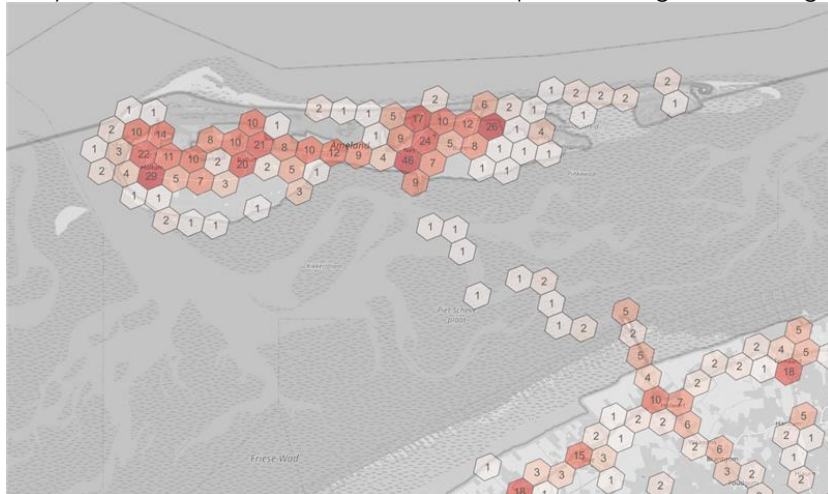
Kort voor de jaarwisseling van 2024 stuurde NEAR twee voorbeeldbestanden van niveau 6 en 8 met een drempelwaarde van 1 per maand. De verwachting was hoog omdat de dataset veel rijker was vanwege de veel lagere drempelwaarde. Polygonen die eerder leeg waren hebben nu bezoekers en station Utrecht Centraal 800+ bezoeker per uur terwijl dat eerder 200 was. CEL rapport omvat heel veel data, ongeveer 3GB per week. Om die eerdere set met het nieuwe voorbeeld te vergelijken is deze geprojecteerd op een kaart van de provincie Friesland.



Afbeelding 5: Vergelijking datasets H6 vs.H8

De data zijn van de eerste week van november 2023. Voor niveau H6 zijn er in totaal 211.471 records, minimum waarde: 1 maximum waarde: 12266. Voor level 8 zijn er in totaal 6.014.434 records. Minimum waarde: 1, maximum waarde: 1.061. In de figuur zie je level 8 ingekleurd en daaronder in groen level 6. Drempelwaarde per uur voor PIN data en CEL waarde per dag. De drempelwaarde 10 per polygoon is ingesteld per maand.

Analyse van de PIN data voor 4 november op Ameland geeft het volgende beeld.



Afbeelding 6: PIN Data Level H6 Ameland op 4 November 14 uur

Bezoekers aan centrum van Leeuwarden (H6) op 4 november 2023 naar herkomstland:

3	Rijlabels	Som van Visitors
4	BEL	1
5	DEU	2
6	ESP	2
7	GBR	1
8	NLD	934
9	SWE	1
10	USA	1
11	Eindtotaal	942

Afbeelding 7: PN Data Level H6 Ameland op 4 November 14 uur

Herkomst bezoekers centrum Leeuwarden op basis van CEL rapport niveau H6 op een zaterdag in November 2023.



Afbeelding 8: Herkomst bezoekers centrum Leeuwarden (CEL/H6) op zaterdag 4 november 2023

Van de 934 bezoekers komen er 553 uit de gemeente Leeuwarden (59,2%). Op basis van deze observatie is het nu ook heel interessant om specifieke evenementen te bekijken (bijvoorbeeld Oerol) of bijvoorbeeld: Welke provincie stad trekt op zaterdag de meeste bezoekers van buiten of hoe verdelen de bezoekers van de binnenstad van Leeuwarden op zaterdag zich over de 52 weken van het jaar? De CEL-data bevat qua herkomst zowel de hexagoon (common

evening), het land, postcode 4 als gemeente. Op basis van de analyse is niveau 8 het meest voor de hand liggend voor de PIN data, dus de platte telling en voor CEL level 6, de herkomst data. Dit omdat level 8 dan te weinig waarden oplevert ook al ligt de drempelwaarde bij 10 per uur. Voor herkomstdata moeten een hoger niveau gekozen worden met grotere polygoonen om een beter beeld te krijgen.

Lesson learned: Voor gedetailleerde druktebeelden (bijvoorbeeld bezoekers op de Dam in Amsterdam) kan niveau 8 niet voldoende zijn. Dus de mate van stedelijkheid kan bepalend zijn voor het niveau dat gekozen wordt. Voor herkomstdata is het juist wel beter grotere polygoonen te kiezen omdat anders de drempel van 10 tellingen per uur niet bereikt wordt.

Data detailniveau 8 drempelwaarde 10 per maand per polygooneind januari 2024

De grootste dataset is verstuurd door NEAR eind januari 2024 (22GB aan gecomprimeerde data). Door de lage drempelwaarde zijn de rapportagebestanden aanzienlijk groter dan bij de eerdere datasets. De uitdaging is nu deze in twee grote tabellen in een online database te zetten die deze snel en betrouwbaar beschikbaar maakt voor analyse. Dat gebeurt door eerst iedere dataset in nieuwe databasetabellen te zetten. Daarna voegen we die tabellen samen naar CEL en PIN. Het ondersteunende ICT bedrijf heeft de databank op het snelste niveau moeten zetten om een soepel verloop daarvan te garanderen. Helemaal vlekkeloos verloopt de importoperatie niet. Het probleem zit met name in de verwerking van de grootte van de tabellen. Op alle fronten werkt de omvang van de data suboptimaal: opslag, tooling, geheugen en verbinding blijken struikelblokken te bevatten. Daarom is uit tijdsoverweging in eerste instantie gekozen voor een lokale database op een stand-alone computer en pas in het vervolg voor een Azure databank.

Lessons learned: Er is nu een stappenplan hoe grote datahoeveelheden in de Azure databank geplaatst kunnen worden.

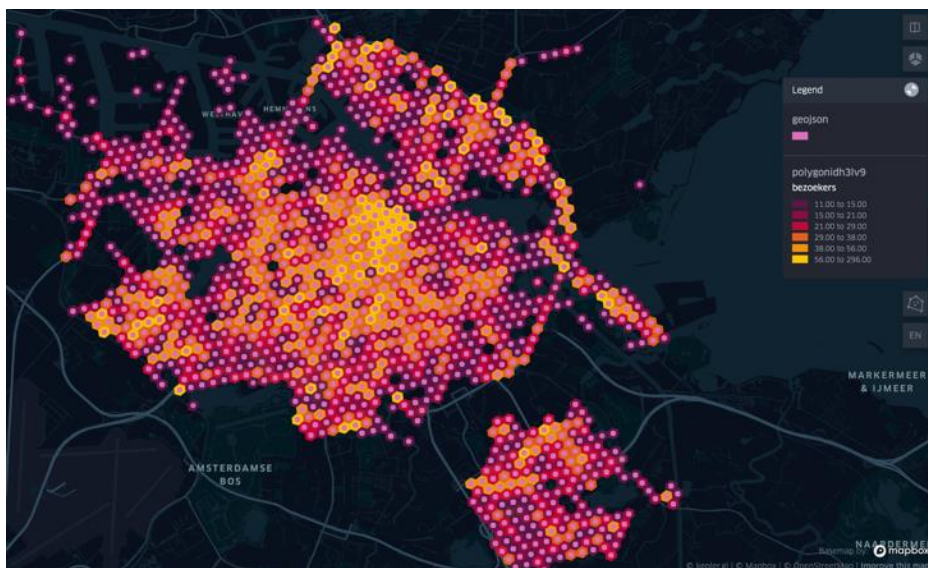
Maar ook nu weer blijkt bij nader inzien dat de PIN data tot Mei 2023 niet alle uren bevat. Waar eerder de bezoekers per uur in de bestanden te vinden waren, stond er nu alleen een 1, 2 of 3. Aan NEAR het verzoek gestuurd om hierop te reageren, waarop vrij snel de omissie ontdekt werd en een nieuw dataset werd gegenereerd dat na ontvangst in onze databank geladen moest worden om de oude data te vervangen. Vanwege technische beperkingen (geheugenruimte) is ervoor gekozen om alleen de CEL data van Friesland, regio Amsterdam en Terschelling in een online databank te zetten van Data Fryslân.

De vervolgstappen voor het onderzoek, nadat de data klaar stond in de database, waren:

- Per hexagon: waar mensen naar toe gaan en vandaan komen (visualisatie per woonplaats/gemeente/provincie)
 - Voor een heel jaar
 - Voor de dagen van de week
 - Per dagdeel de drukte in kaart

Experimentele visualisatie voor CEL en PIN data

- Provincie Fryslân
- Eiland Terschelling in kepler GL / qgis



Abbeelding 9: Drukbeeld bezoekers op een dag in Amsterdam niveau 9 op basis van eerste ontvangen dataset

De nieuwe dataset met een lagere drempelwaarde omvat aanzienlijk meer data wat op alle datahandling fronten een uitdaging is. De mogelijkheden voor analyse zijn interessant maar kunnen vanwege de ontbrekende tijd nog niet volop benut worden. De volgende stap is de informatietools te ontwikkelen.

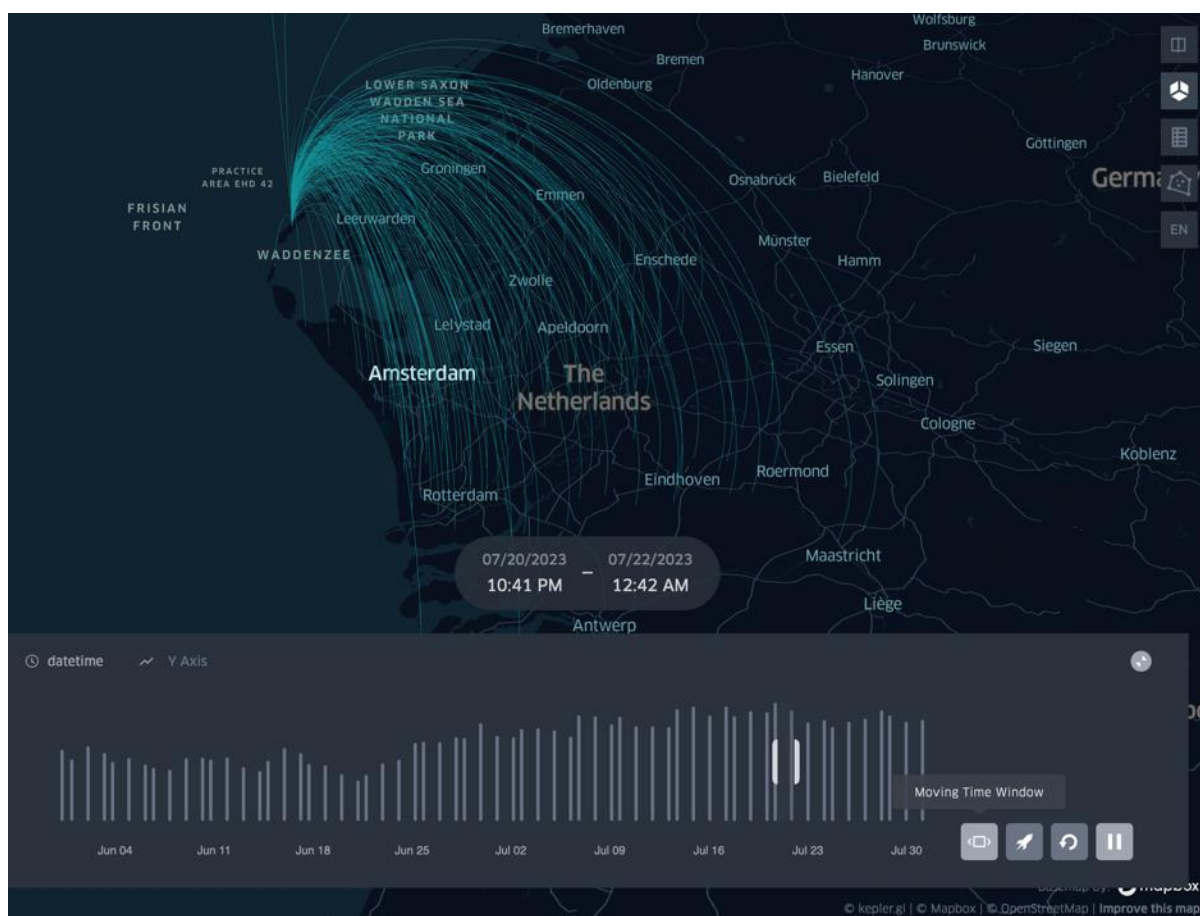
Lesson learned: Niveau 8 levert relevante verplaatsingsdata maar druktebeeld is voor centrale kernen misschien te grofmazig.

Van data naar informatie

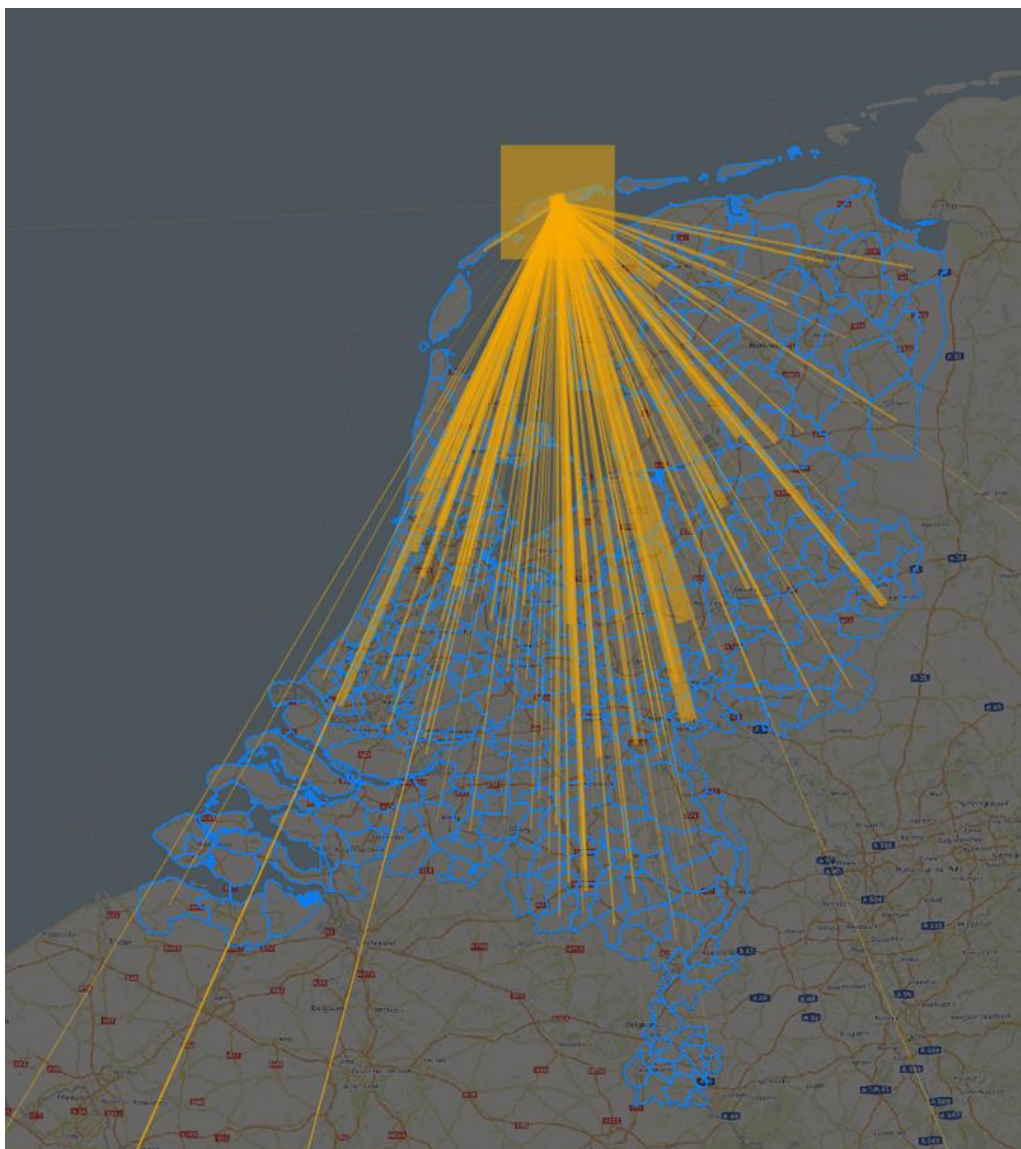
Met alle ervaring in het handhaven van grote datasets en het toegankelijk maken van de data via een snelle SQL database/Azure is de volgende stap gemaakt om informatietools te ontwikkelen om informatie te generen. Hieronder enkele voorbeelden van de mogelijkheden vanuit de PIN en CEL data. PIN geeft het druktebeeld van bezoek weer en CEL data is de herkomst van het bezoek.

De onderstaande impressies dienen alleen ter illustratie van de data en verdere analyse moet gedaan worden om weging en bruikbaarheid de data te bepalen.

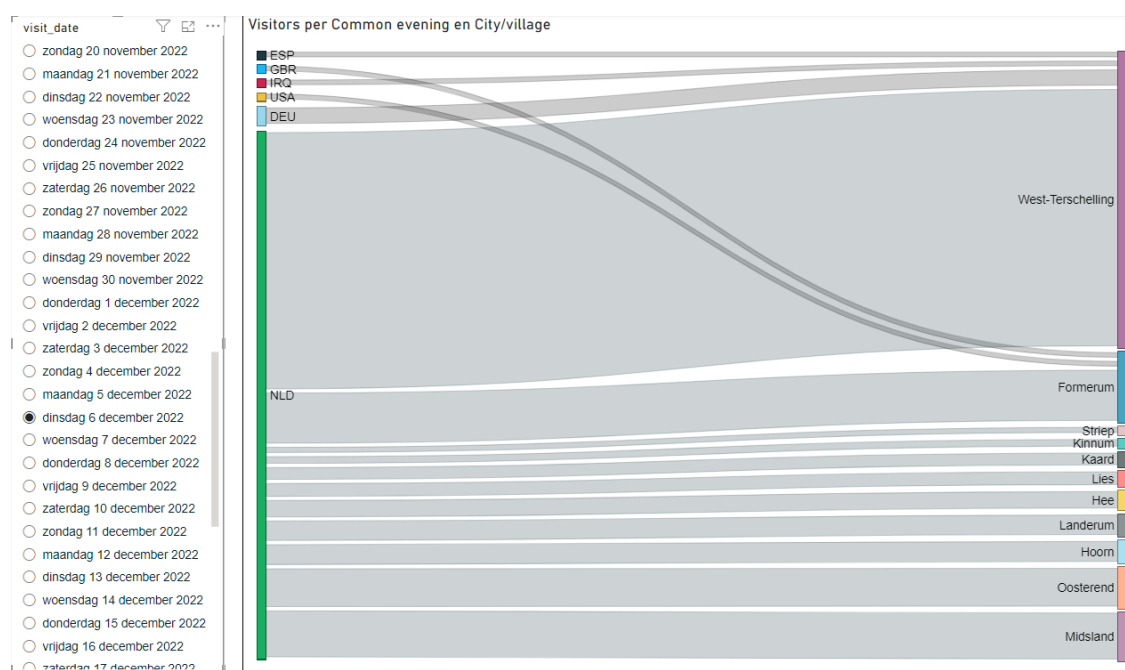
Impressies van de mogelijkheden met NEAR's CEL data:



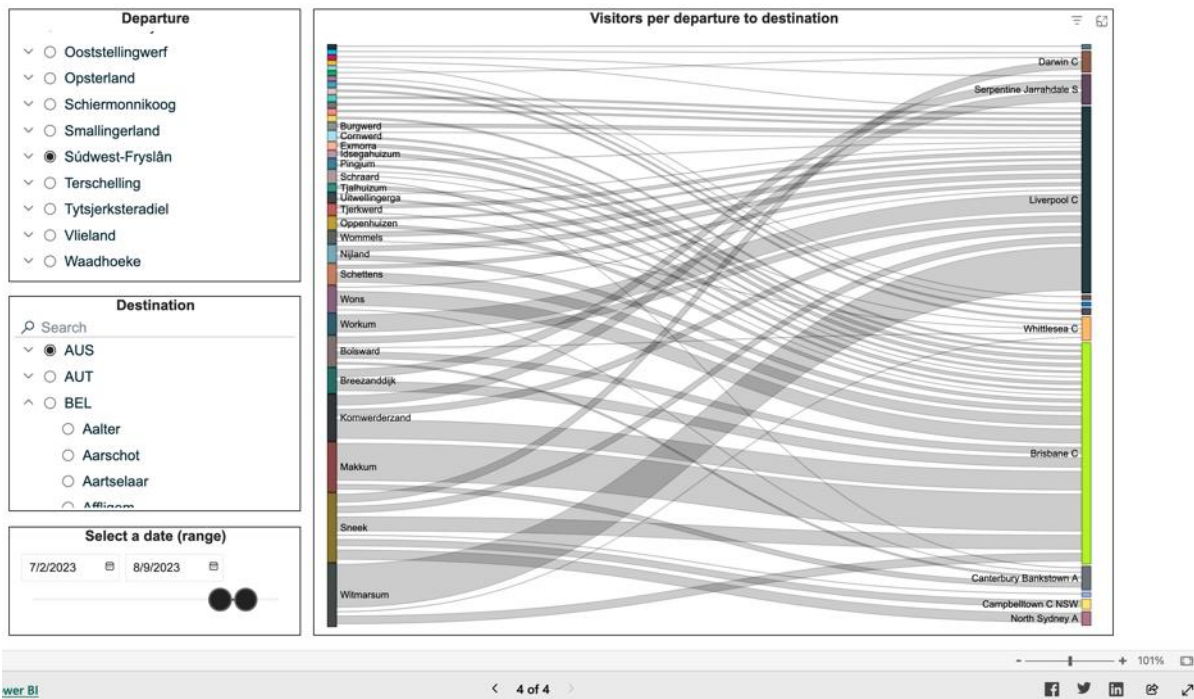
Afbeelding 10: Herkomst bezoekers Terschelling in timeline juni op een kaart geprojecteerd met Kepler GL



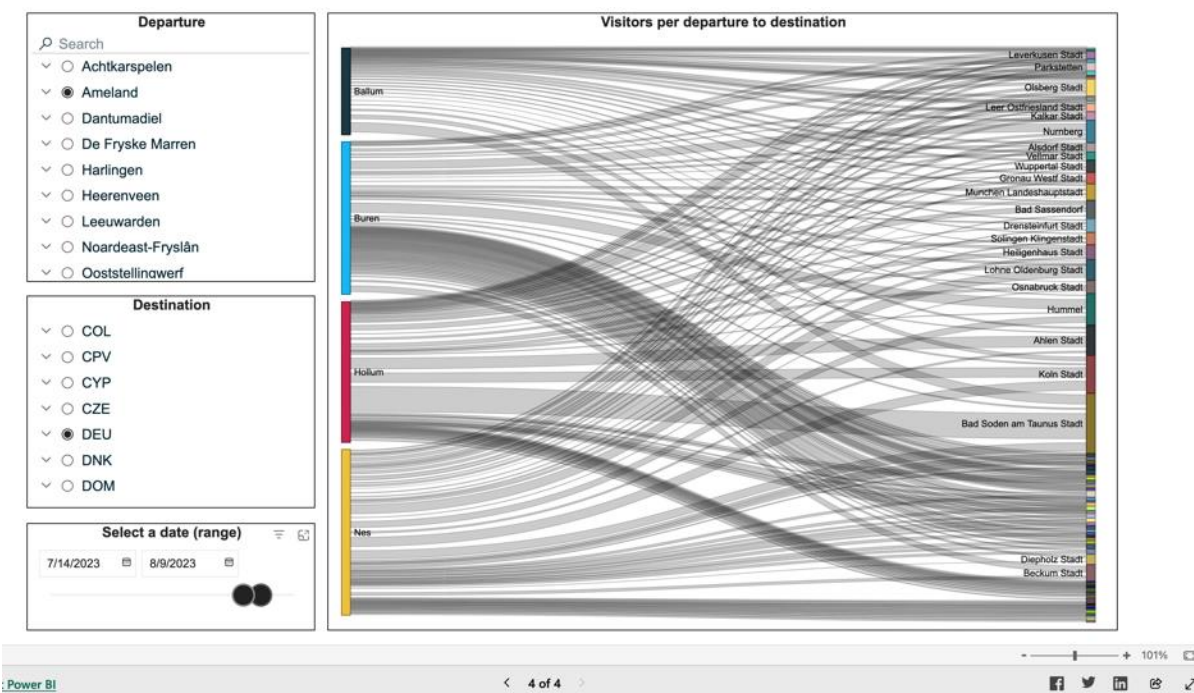
Afbeelding 11: CEL rapport herkomst bezoekers op een juni dag op Terschelling in QGIS



Afbeelding 12: CEL data herkomst bezoekers van Terschelling op een dag in december 2022

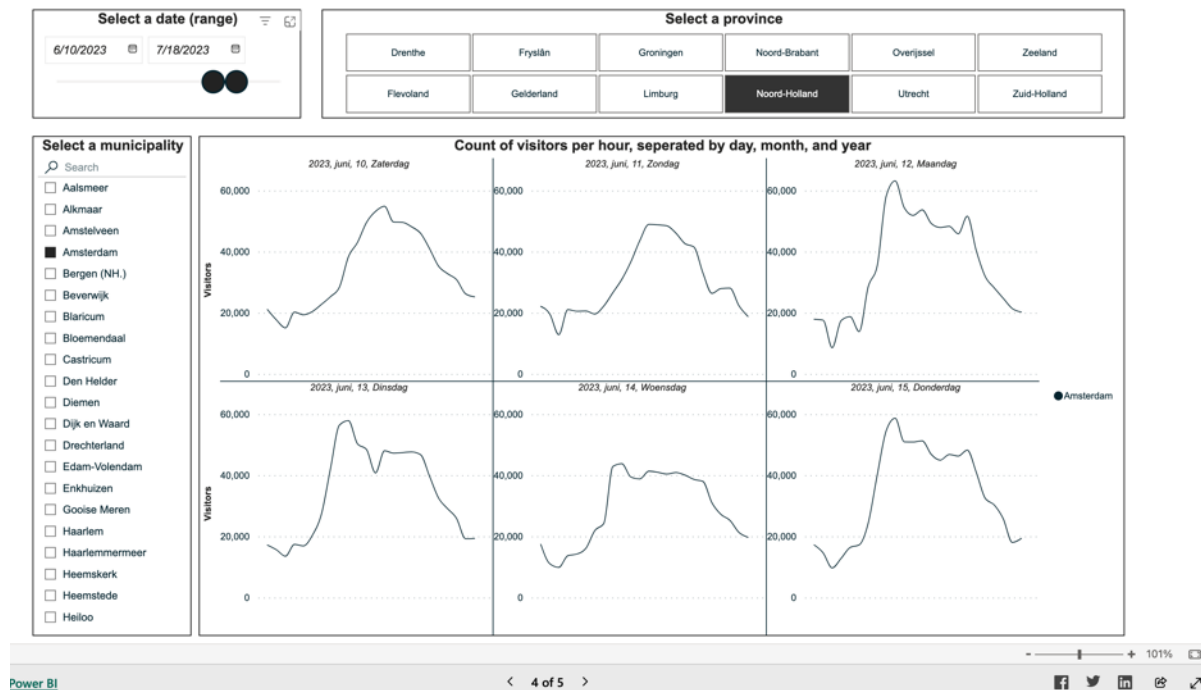


Afbeelding 13: CEL data herkomst Australische bezoekers aan Sudwest Fryslan

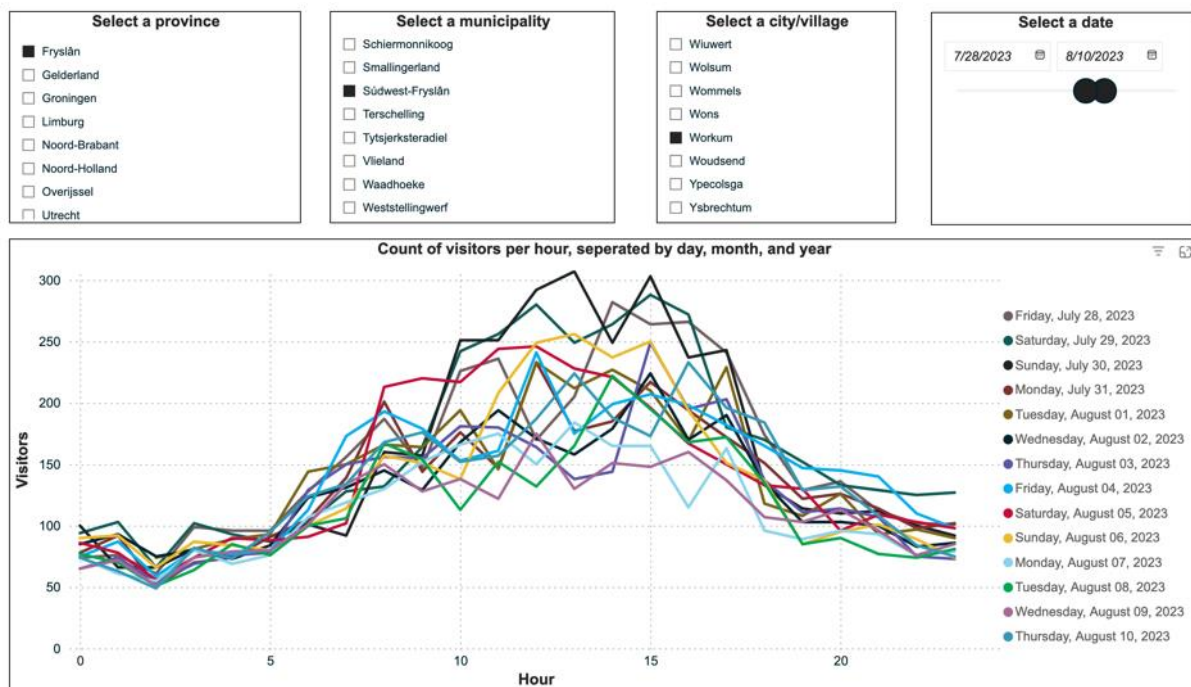


Afbeelding 14: CEL data herkomst Duitse bezoekers op dorpen op Ameland

Impressies van de mogelijkheden met NEAR's PIN data



Afbeelding 15: Bezoekers per gemeente in een periode gespecificeerd per dag in PowerBI



Afbeelding 16: Bezoekers Workum per uur binnen 14 dagen in het hoofdseizoen in PowerBI

4. Reflectie op de pilot

Reflectie op de uitkomsten

Het ontvangen van goede data via NEAR was een cruciaal element in het proces. Hoe logisch dit ook is, bleek de praktijk weerbarstig. Het kostte zo'n drie maanden voordat er een bruikbare en consistente dataset in de eigen databank stond. De tussentijdse analyse van datasets die achteraf niet bruikbaar waren vergde verhoudingsgewijs veel tijd. De experimentele fase is afgerond, maar kan nog veel vervolg krijgen. Samen met partners zou nu gekeken moeten worden naar tools waar toegankelijkheid van de data voorop staat.

De informatietools die zijn ontwikkeld, zijn klaar om aan te bieden aan werkveldpartners en deze laten experimenteren met de tools om te bepalen hoe waardevol de data is. Een aantal opmerkelijke mogelijkheden die nu via het PowerBI Dashboard mogelijk is:

- Druktebeeld, per uur, in iedere gemeente/plek van Nederland (op basis van PIN dataset): directe link naar [PowerBI PIN data](#)
- Herkomst nationaal en internationaal bezoek Friesland (op basis van CEL dataset): directe link naar [PowerBI CEL data](#)

De herkomstdata (CEL data) is beperkt toegankelijk omdat deze alleen voor Friesland in een Azure databank staat en via het Power BI dashboard benaderd kan worden. De druktedata (PIN data) staat voor heel Nederland in de databank en is volledig beschikbaar via het Power BI dashboard.

Omdat het een experiment was hebben de huidige tools nog een hoog data science gehalte en deze zijn niet zozeer geënt op toegankelijkheid. Samen met partners (bijv. amsterdam&partners, NBTC en Visit Veluwe) en nog te definiëren gebruikersgroepen kan er een pilot opgestart worden om deze tools te verbeteren. Belangrijk is hier om de eindgebruikers mee te nemen in het proces om enerzijds de kwaliteit van de te ontwikkelende tools/dashboards te garanderen, maar ook de datageletterdheid van de eindgebruikers te verhogen zoals beleidsmakers, DMOs, brancheverenigingen, business support organisaties (BSOs) en ondernemers.

Reflectie op de beperkingen van de data

De beperkingen van de data spelen ook een rol, met name de herkomst van de data van de verschillende apps. De data geven geen beeld van de daadwerkelijk drukte, alleen een relatieve drukte gerelateerd aan de gebruikers van de verschillende apps. Dus er kan een vergelijking plaatsvinden tussen gebieden. Die regio's zijn te vergelijken op basis van de apps die gebruikt worden. Het gaat om drukte-indicaties van mensen die bepaalde apps gebruiken die een plek hebben bezocht. Dus we kunnen alleen spreken van relatieve drukte o.b.v. appgebruikers die op een bepaalde plek waren. Er kan dus niet gesteld worden dat X bezoekers in Leeuwarden waren, maar wel dat Leeuwarden op een bepaald moment drukker was dan bijvoorbeeld Sneek. Relatieve vergelijkingen lijken mogelijk. Verdere analyse van de betrouwbaarheid is aanbevolen omdat het alleen om een relatief drukte- en herkomstbeeld gaat. Hier zou de dataset vergeleken kunnen worden met echte tellingen van gemeentes (bijv parkeerdata), festivals (verkoopcijfers) of relatieve teldata via het Mobidot Verplaatsingspaneel.

Reflectie op het gebruik van de data: tips & adviezen

- Zorgvuldig specificeren welke data nodig is. Opvragen gegevens over Monthly Active Users (MAUs)
- Zorgvuldige check op geleverde data
- Kennis, kunde en infrastructuur om grote hoeveelheid data te kunnen verwerken
- Diverse experts nodig: server inrichten, data bewerking, data visualisatie/presentatie, data interpretatie

Reflectie op de mogelijke toepassingen van de data:

Pin niveau 8

- Relatieve druktebeelden, vrij nauwkeurig op uur niveau over een jaar, op niveau 9 (ongeveer afstand CS - Dam), over heel Nederland. Ieder plek kan hiervan gebruik maken en druktebeelden creëren.
- Visualisaties van druktes over een seizoen/tijdsperiode om drukteverdeling op een dorp/stad/regio/land te visualiseren redelijk eenvoudig via Kepler GL

- Vergelijkende studies tussen plekken o.b.v. evenementen/gebeurtenissen. Meer bezoekers op plek A dan B i.v.m. een reden.
- Vergelijken met andere datasets om beeld te krijgen hoe realistisch de data is.

Cel niveau 8

- Herkomst bezoek nationaal/internationaal vrij nauwkeurig op dagniveau over een jaar, op niveau 9 (ongeveer afstand CS-Dam), over heel Nederland. Ieder plek kan hiervan gebruik maken en herkomst visualiseren.
- Omgedraaid kan de beweging van een specifieke herkomstland/herkomstpostcode gevisualiseerd worden. Waar waren bezoekers uit Duitsland/regio/postcode op een bepaald moment. Dat kan voor een gebied gevisualiseerd worden in een tijdspad. Het dynamische druktebeeld van een bepaalde herkomst kan hypothetisch ook iets zeggen over de beweging van deze groep mits deze beweegt.
-
- Selecteren op specifieke relevante herkomstcriteria (Nationaliteit en EU-postcodeniveau, non EU op land) om te zien hoe specifieke bezoekers zich door stad/regio/land bewegen.
- Individuele bewegingsbeelden zijn in principe niet mogelijk
- Groepsbewegingen zijn in potentie misschien wel te achterhalen als de beweging op dagniveau plaatsvindt (schoolklas op een bruine vloot schip die langs de IJsselmeer/wadden havens gaat. Riviercruises. Het filteren op groepsbewegingen zou in principe moeten kunnen.
-
- Koppeling van herkomstdata aan leefstijlvinder en daarmee rijkere bezoeker data/bezoeker profiel
-
- Er kan potentieel veel meer met de herkomstdata omdat de verblijfstijd van individuen/groepen iets kan zeggen over dagrecreatie/werk of langer weekend weg/bezoek vrienden/familie of vakantie

Vervolg stappen en mogelijk vervolgonderzoek

Vanuit de pilot hebben we geleerd dat het gebruik van NEAR data complex kan zijn. Bij het gebruik van de data is toetsing van de data erg belangrijk. In de pilot zijn enkele analyses uitgevoerd, maar de dataset biedt nog veel meer analysemogelijkheden. De druktedata is volledig beschikbaar via de Azure Databank en toegankelijk via PowerBI. In verband met de omvang is de herkomstdata momenteel alleen gedeeltelijk toegankelijk. Omdat de CEL herkomstdata zeer omvangrijk is, kan deze beter opgeknipt worden in kleine sets (datamarks). Verder zal een analyse van betrouwbaarheid, relevantie en het eventueel vergelijken of eiken met andere datasets (NL Verplaatsingspanel) via Python moeten plaatsvinden.

Het onderwijs en DMO's kunnen de data gebruiken om in gezamenlijkheid de bruikbaarheid van de data verder te toetsen en use-cases te kunnen genereren.

Bijlage 1: technische verantwoording

Eerste verkenning van Hexagon Level 8 NEAR data

Eerste analyse gps data: celreport

```
import pandas as pd
import geopandas as gpd
from sqlalchemy import create_engine
import matplotlib.pyplot as plt
import contextily as cx
import rasterio
from rasterio.plot import show as rioshow
import json, configparser
# connectie gegevens uitlezen uit ini file
config = configparser.ConfigParser()
config.read('DB_connectie.ini')
db = 'NEAR'
server = config.get(db, 'server')
port = config.get(db, 'port')
uid = config.get(db, 'user')
pwd = config.get(db, 'pwd')
engine = create_engine(f"postgresql://{uid}:{pwd}@{server}:{port}/{db}")
con = engine.connect()
# hoe vaak is het aantal bezoekers 0 IN CELREPORT
con.rollback()
sql = "SELECT * FROM data.trf_hex_uni_celreport WHERE bezoekers is NULL or bezoekers = 0 or com_ev_polygonidh3lv9 is null"
df = pd.read_sql(sql, con)
print(len(df))
0
# hoe vaak is het aantal bezoekers 0v in PINREPORT
sql = "SELECT * FROM data.trf_hex_uni_pinreport WHERE bezoekers= 0 or bezoekers is NULL"
df = pd.read_sql(sql, con)
print(len(df))
0
sql = "SELECT * FROM data.v_celreport_lwd"
df = pd.read_sql(sql, con)
df.head()
```

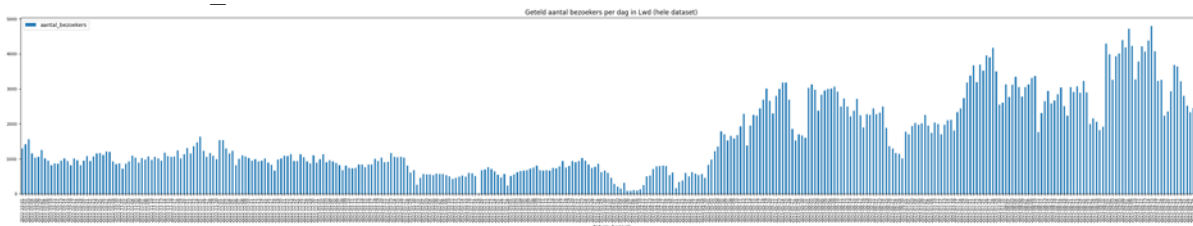
	id	polygonidh3lv9	com_ev_polygonidh3lv9	com_ev_land	com_ev_adm	com_ev_mic	com_ev_pc4	com_ev_gem	datum bezoeker	bezoekers
0	9827151	89196d196c7ffff	89196d196c7ffff	NLD	None	00804502	8935.0	Leeuwarden	2023-09-05	25
1	9827153	89196d196b3ffff	89196d19433ffff	NLD	None	00805003	8939.0	Leeuwarden	2023-09-05	21
2	9827155	89196d19437ffff	89196d19437ffff	NLD	None	00805003	8939.0	Leeuwarden	2023-09-05	65
3	9827156	89196d196cbffff	89196d196cbffff	NLD	None	00804503	8935.0	Leeuwarden	2023-09-05	20

	id	polygonidh3lv9	com_ev_polygonidh3lv9	com_ev_land	com_ev_adm	com_ev_mic	com_ev_pc4	com_ev_gem	datum_bezoek	bezoekers
4	9827159	89196d1955bffff	89196d19427ffff	NLD	None	00805003	8939.0	Leeuwarden	2023-09-05	12

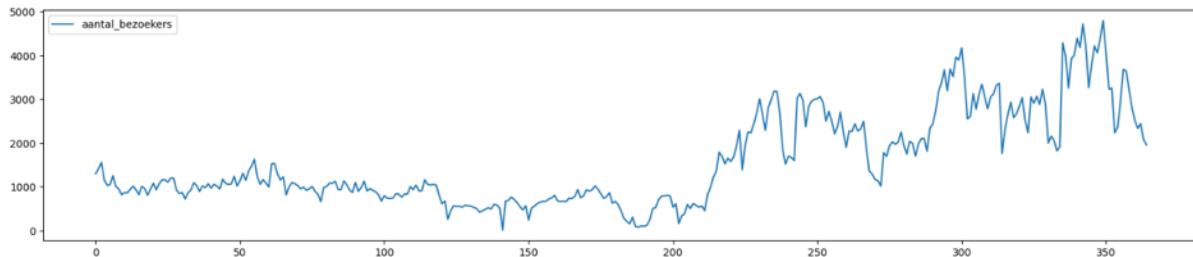
```
# geteld per dag in Lwd, hele dataset
```

```
sql = """
SELECT datum_bezoek, d.month_name as maand, sum(bezoekers) aantal_bezoekers
FROM data.v_celreport_lwd c
    inner join data.dim_datum d ON c.datum_bezoek = d.date_actual
GROUP BY datum_bezoek, d.month_name
ORDER BY 1 asc
"""
```

```
df = pd.read_sql(sql, con)
df.plot.bar(x='datum_bezoek',figsize=[40,6], title="Geteld aantal bezoekers per dag in Lwd (hele dataset)",fontsize=8)
<Axes: title={'center': 'Geteld aantal bezoekers per dag in Lwd (hele dataset)'},
xlabel='datum_bezoek'>
```



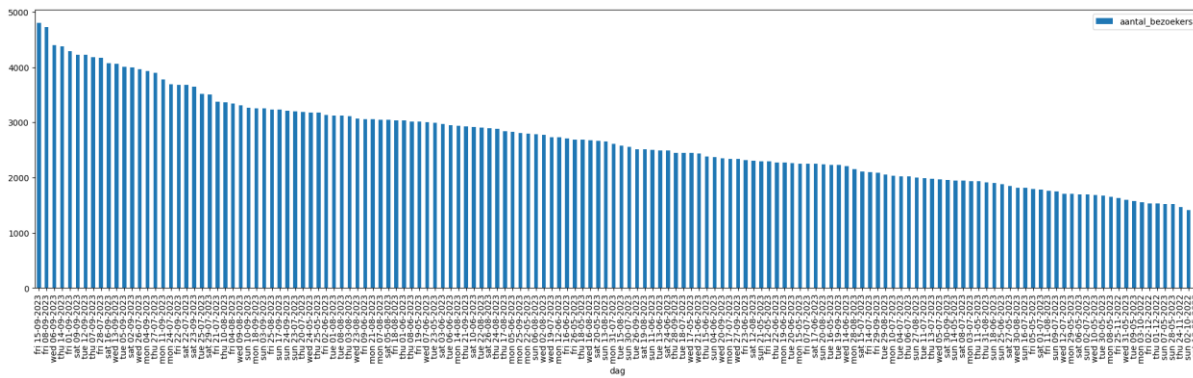
```
df.plot(figsize=[20,4])
plt.show()
```



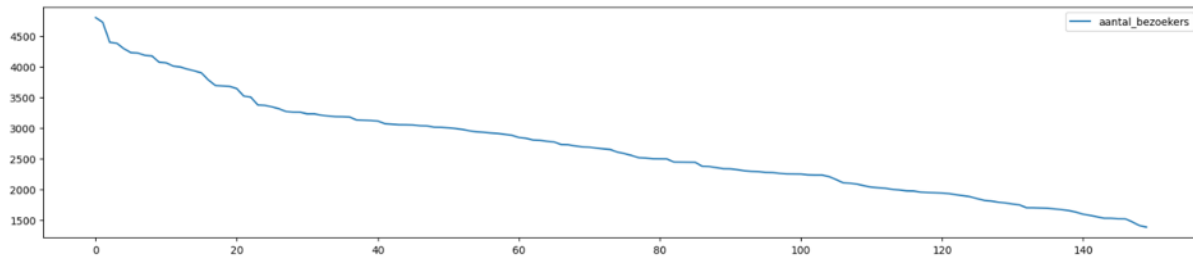
```
# drukste 150 dagen in Leeuwarden
```

```
con.rollback()
sql = """
select to_char(datum_bezoek, 'dy DD-MM-IYYY') dag, * from (
    SELECT datum_bezoek, substr(lower(d.month_name),1,3) as maand,
lower(SUBSTR(d.day_name,1,2)) weekdag,
    sum(bezoekers) aantal_bezoekers
FROM data.v_celreport_lwd c
    inner join data.dim_datum d ON c.datum_bezoek = d.date_actual
GROUP BY datum_bezoek, d.month_name, d.day_name
ORDER BY aantal_bezoekers desc
limit 150)
"""
```

```
df = pd.read_sql(sql, con)
df.plot.bar(x='dag',figsize=[25,6])
<Axes: xlabel='dag'>
```



```
df.plot(figsize=[20,4])
plt.show()
```



```
# gem. drukte per weekday, verdeeld per kwartaal
```

```
con.rollback()
```

```
sql = """
```

```
    SELECT d.day_of_week dow, d.day_name as weekday, d.quarter_actual as kwartaal,
    AVG(bezoekers)::numeric(10,2) aantal_bezoekers
```

```
    FROM data.v_celreport_lwd c
```

```
        INNER JOIN data.dim_datum d ON c.datum_bezoek = d.date_actual
```

```
    GROUP BY 1,2,3
```

```
    ORDER BY 1,2,3
```

```
    """
```

```
df = pd.read_sql(sql, con)
```

```
# pivot maken om kwartaal naar kolommen te brengen
```

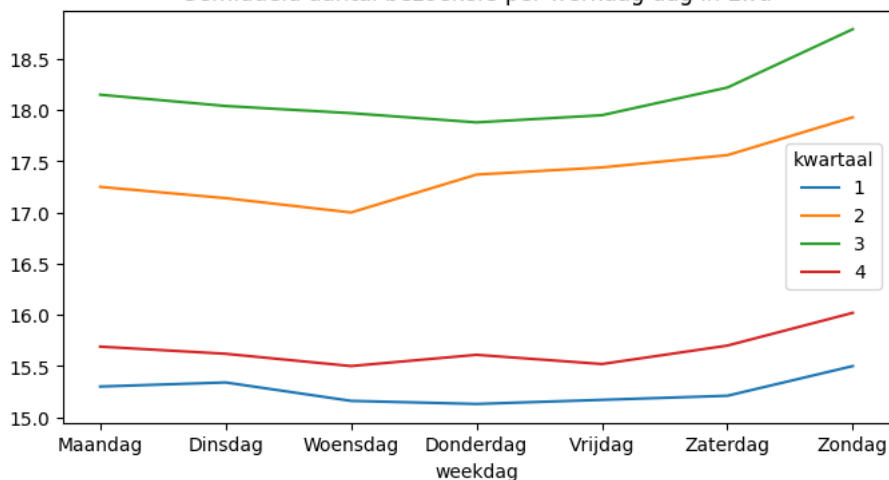
```
piv = pd.pivot_table(df,
columns='kwartaal',index=['dow','weekday'],values='aantal_bezoekers')
```

```
piv['weekday'] = piv.index.get_level_values(1)
```

```
piv.plot(x='weekday',figsize=[8,4], title="Gemiddeld aantal bezoekers per werkdag dag in Lwd")
```

```
<Axes: title={'center': 'Gemiddeld aantal bezoekers per werkdag dag in Lwd'},
xlabel='weekday'>
```

Gemiddeld aantal bezoekers per werkdag dag in Lwd

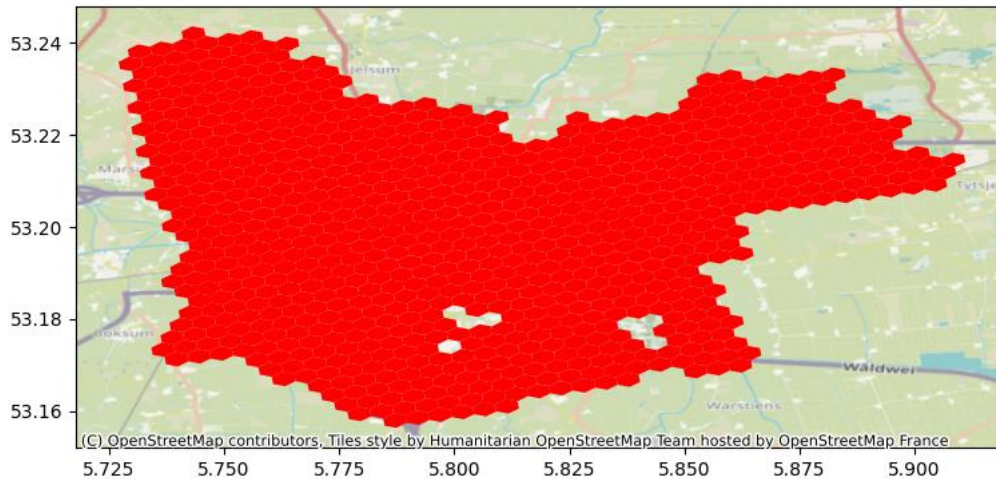


```
# laden en displayen alle hexagons uit gpkg file
```

```
db = gpd.read_file('./leeuwarden_hex_h3.gpkg')
```

```
ax = db.plot(color="red", figsize=(9, 9))
```

```
cx.add_basemap(ax, crs=db.crs.to_string())
```



Op 15-09-2023 zijn in Leeuwarden de meeste bezoekers.

Wat zijn de top5 hexagons en waar in Leeuwarden zijn die??

```
# welke hexagons worden er op vr 15-09-2023 het meest bezocht?
con.rollback()
sql = """
    SELECT polygonidh3lv9, geometry, sum(bezoekers) as aantal_bezoekers
    FROM data.v_celreport_lwd c
    INNER JOIN data.h3lv19_leeuwarden ON c.polygonidh3lv9 = h3lv19_leeuwarden.h3hex
    WHERE datum_bezoek = '15-09-2023'
    GROUP BY polygonidh3lv9, geometry
    ORDER BY aantal_bezoekers DESC
    LIMIT 5
    """

#df = pd.read_sql(sql, con)
gdf = gpd.read_postgis(sql, con, geom_col='geometry')
gdf
```

	polygonidh3lv9	geometry	aantal_bezoekers
0	89196d19447ffff	POLYGON ((5.82956 53.18356, 5.82986 53.18198, 5.83245 53.18...	138.0
1	89196d196b7ffff	POLYGON ((5.82271 53.18027, 5.82302 53.17869, 5.82560 53.17...	120.0
2	89196d19433ffff	POLYGON ((5.84508 53.18069, 5.84539 53.17911, 5.84798 53.17...	119.0
3	89196dccc3bffff	POLYGON ((5.83382 53.21315, 5.83412 53.21158, 5.83671 53.21...	97.0
4	89196d196b3ffff	POLYGON ((5.81982 53.18232, 5.82013 53.18075, 5.82271 53.18...	90.0

```
gdf['geometry'].centroid
C:\Users\odijks\AppData\Local\Temp\ipykernel_27468\3996546279.py:1: UserWarning:
Geometry is in a geographic CRS. Results from 'centroid' are likely incorrect. Use
'GeoSeries.to_crs()' to re-project geometries to a projected CRS before this operation.
```

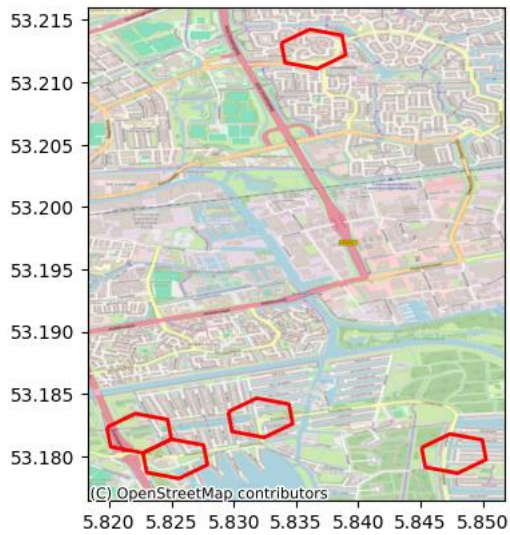
```
gdf['geometry'].centroid
0    POINT (5.83215 53.18308)
1    POINT (5.82530 53.17979)
2    POINT (5.84767 53.18021)
3    POINT (5.83641 53.21267)
4    POINT (5.82241 53.18184)
dtype: geometry
cx.providers #dit laat zien welke map-providers er zijn (onderliggend kaartmateriaal)
xyzservices.Bunch
39 items
21
```


- OpenStreetMap xyzservices.Bunch
- MapTilesAPI xyzservices.Bunch
- OpenSeaMap xyzservices.TileProvider
- OPNVKarte xyzservices.TileProvider
- OpenTopoMap xyzservices.TileProvider
- OpenRailwayMap xyzservices.TileProvider
- OpenFireMap xyzservices.TileProvider
- SafeCast xyzservices.TileProvider
- Stadia xyzservices.Bunch
- Thunderforest xyzservices.Bunch
- CycloSM xyzservices.TileProvider
- Jawg xyzservices.Bunch
- MapBox xyzservices.TileProvider
- MapTiler xyzservices.Bunch
- TomTom xyzservices.Bunch
- Esri xyzservices.Bunch
- OpenWeatherMap xyzservices.Bunch
- HERE xyzservices.Bunch
- HEREv3 xyzservices.Bunch
- FreeMapSK xyzservices.TileProvider
- MtbMap xyzservices.TileProvider
- CartoDB xyzservices.Bunch
- HikeBike xyzservices.Bunch
- BasemapAT xyzservices.Bunch
- nlmaps xyzservices.Bunch
- NASAGIBS xyzservices.Bunch
- NLS xyzservices.TileProvider
- JusticeMap xyzservices.Bunch
- GeoportailFrance xyzservices.Bunch
- OneMapSG xyzservices.Bunch
- USGS xyzservices.Bunch
- WaymarkedTrails xyzservices.Bunch
- OpenAIP xyzservices.TileProvider
- OpenSnowMap xyzservices.Bunch
- AzureMaps xyzservices.Bunch
- SwissFederalGeoportal xyzservices.Bunch
- Gaode xyzservices.Bunch
- Strava xyzservices.Bunch
- OrdnanceSurvey xyzservices.Bunch

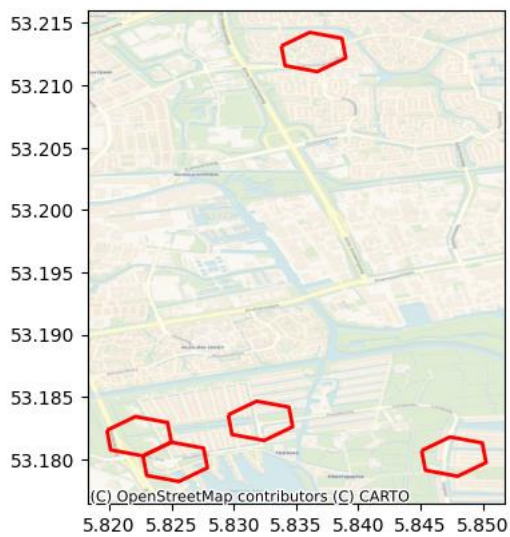
4 vd 5 hexagons zijn in de wijk Zuiderburen

```
# gebruik de plot functie van geopandas (uiteindelijk is dat matplotlib) en het package
contextily (cx)
# om de hexagons op een kaart te plotten

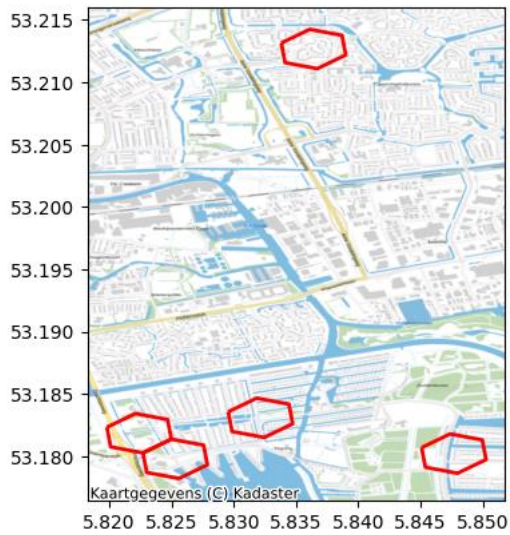
ax = gdf.plot(facecolor="none",
              edgecolor="red",
              linewidth=2
            )
cx.add_basemap(ax,
              crs=gdf.crs.to_string(),
              source=cx.providers.OpenStreetMap.DE
            )
```



```
# zelfde met provider CartoDB
ax = gdf.plot(facecolor="none",
              edgecolor="red",
              linewidth=2
            )
cx.add_basemap(ax,
               crs=gdf.crs.to_string(),
               source=cx.providers.CartoDB.Voyager
            )
```



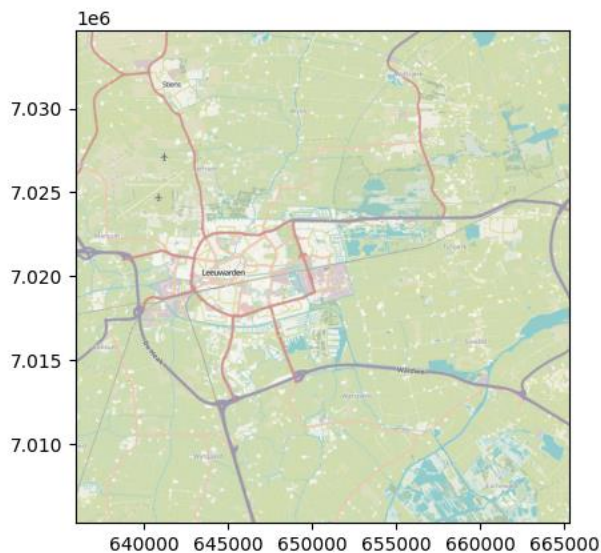
```
# zelfde met provider NLMAPS
ax = gdf.plot(facecolor="none",
              edgecolor="red",
              linewidth=2
            )
cx.add_basemap(ax,
               crs=gdf.crs.to_string(),
               source=cx.providers.nlmaps.standaard
            )
```



```
# zelfde met provider nlmaps.luchtfoto
ax = gdf.plot(facecolor="none",
              edgecolor="red",
              linewidth=2
            )
cx.add_basemap(ax,
               crs=gdf.crs.to_string(),
               source=cx.providers.nlmaps.luchtfoto
            )
```



```
# plot kaartje van Leeuwarden
leeuwarden = cx.Place("Leeuwarden", path="./leeuwarden.tif")
with rasterio.open("leeuwarden.tif") as r:
    rioshow(r)
```



Wat is de herkomst van de bezoeken aan Leeuwarden?

```
# voeg analyse toe op herkomst
con.rollback()
sql = """
    SELECT polygonidh3lv9, com_ev_polygonidh3lv9, geometry, sum(bezoekers) as
aantal_bezoekers
    FROM data.v_celreport_lwd c
    INNER JOIN data.h3lv9_leeuwarden ON c.polygonidh3lv9 = h3lv9_leeuwarden.h3hex
    WHERE datum_bezoek = '15-09-2023'
    GROUP BY polygonidh3lv9, com_ev_polygonidh3lv9, geometry
    ORDER BY aantal_bezoekers DESC
    --LIMIT 5
    """
```

```
gdf = gpd.read_postgis(sql, con, geom_col='geometry')
gdf
```

	polygonidh3lv9	com_ev_polygonidh3lv9	geometry	aantal_bezoekers
0	89196d19437ffff	89196d19437ffff	POLYGON ((5.84798 53.17863, 5.84828 53.17706, ...	64.0
1	89196d19433ffff	89196d19433ffff	POLYGON ((5.84508 53.18069, 5.84539 53.17911, ...	62.0
2	89196dcc81bffff	89196dcc81bffff	POLYGON ((5.79591 53.21559, 5.79621 53.21402, ...	58.0
3	89196dccc0fffff	89196dccc0fffff	POLYGON ((5.83092 53.21521, 5.83123 53.21363, ...	57.0
4	89196dcca0bffff	89196dcca0bffff	POLYGON ((5.75984 53.20857, 5.76014 53.20700, ...	55.0
...	...	...	...	...
241	89196d19417ffff	89196d19433ffff	POLYGON ((5.84417 53.18542, 5.84448 53.18384, ...	11.0
242	89196d19667ffff	89196d19667ffff	POLYGON ((5.79928 53.17244, 5.79959 53.17087, ...	11.0
243	89196d19607ffff	89196d19607ffff	POLYGON ((5.80810 53.17841, 5.80841 53.17684, ...	11.0
244	89196d194abffff	89196d19437ffff	POLYGON ((5.84706 53.18336, 5.84737 53.18179, ...	11.0
245	89196d1940bffff	89196d194afffff	POLYGON ((5.83443 53.18418, 5.83473 53.18260, ...	11.0

246 rows × 4 columns

```
# Van de lijst herkomst aggregeren op kolom "com_ev_polygonidh3lv9", en kolom
#aantal_bezoekers" totaliseren.
herkomst =
gdf.groupby('com_ev_polygonidh3lv9').agg(totaal=pd.NamedAgg(column="aantal_bezoekers",
aggfunc="sum"))
herkomst = herkomst.reset_index()
from folium import Map, Marker, GeoJson
from geojson.feature import *

res = 9 # geef het lvl van de h3 hex aan, in ons geval 9
#list_hex_res = []
list_hex_res_geom = []

# coördinaten (centroide) vd hexagons toevoegen & verhoudingsgetal totaalaantal
toevoegen
max_bezoekers = herkomst['totaal'].max()
import h3
list_centroids = []
list_norm = []
for index, row in herkomst.iterrows():
    hex_id = row['com_ev_polygonidh3lv9']
    h3_centroid = h3.h3_to_geo(hex_id)
    list_centroids.append(h3_centroid)
    # haal de geometry vd hexagon op en convert naar geojson
    h_geom = {"type": "Polygon",
              "coordinates": [h3.h3_to_geo_boundary(h = hex_id, geo_json = True)]
    }
    list_hex_res_geom.append(h_geom)
    list_norm.append(row['totaal'] / max_bezoekers)
herkomst['hex_id'] = herkomst['com_ev_polygonidh3lv9']
herkomst['centroid'] = list_centroids
herkomst['geometry'] = list_hex_res_geom
herkomst['norm'] = list_norm
herkomst
```

	com_ev_polygonidh3lv9	totaal	hex_id	centroid	geometry	norm
0	89196d190c3ffff	53.0	89196d190c3ffff	(53.172389947793356, 5.824236339514233)	{'type': 'Polygon', 'coordinates': [[(5.821649...	0.173203
1	89196d1928bffff	43.0	89196d1928bffff	(53.16661905506138, 5.797918153601903)	{'type': 'Polygon', 'coordinates': [[(5.795331...	0.140523
2	89196d192d7ffff	24.0	89196d192d7ffff	(53.168672699907916, 5.795026325150859)	{'type': 'Polygon', 'coordinates': [[(5.792439...	0.078431
3	89196d19407ffff	13.0	89196d19407ffff	(53.1822648591218, 5.8447801953263365)	{'type': 'Polygon', 'coordinates': [[(5.842192...	0.042484
4	89196d1940bffff	72.0	89196d1940bffff	(53.183700843229616, 5.837017780118912)	{'type': 'Polygon', 'coordinates': [[(5.834430...	0.235294
...	...	...	...	...	...	...
134	89196dccda7ffff	17.0	89196dccda7ffff	(53.214530256482355, 5.851025591843181)	{'type': 'Polygon', 'coordinates': [[(5.848436...	0.055556

	com_ev_polygonidh3lv9	totaal	hex_id	centroid	geometry	norm
135	89196dccdabffff	80.0	89196dccdabffff	(53.215965183766244, 5.843259112279669)	{'type': 'Polygon', 'coordinates': [[(5.840670...	0.261438
136	89196dccdaafffff	86.0	89196dccdaafffff	(53.21391199051986, 5.846152391523911)	{'type': 'Polygon', 'coordinates': [[(5.843563...	0.281046
137	89196dcccdfffff	14.0	89196dcccdfffff	(53.22150478437362, 5.829705348742134)	{'type': 'Polygon', 'coordinates': [[(5.827116...	0.045752
138	89196dcd9d7ffff	12.0	89196dcd9d7ffff	(53.21175950471309, 5.73426716877916)	{'type': 'Polygon', 'coordinates': [[(5.731679...	0.039216

139 rows × 6 columns

zie <https://towardsdatascience.com/uber-h3-for-data-analysis-with-python-1e54acdcc908>

Opzetten van de map

```
import geocoder
a = geocoder.arcgis('Leeuwarden')
a.geojson
{'type': 'FeatureCollection',
 'features': [{'type': 'Feature',
 'properties': {'address': 'Leeuwarden',
 'bbox': [5.74022255, 53.136324803, 5.87022255, 53.266324803],
 'confidence': 3,
 'lat': 53.201324803000034,
 'lng': 5.805222550000053,
 'ok': True,
 'quality': 'Locality',
 'raw': {'name': 'Leeuwarden',
 'extent': {'xmin': 5.74022255,
 'ymin': 53.136324803,
 'xmax': 5.87022255,
 'ymax': 53.266324803},
 'feature': {'geometry': {'x': 5.805222550000053, 'y': 53.201324803000034},
 'attributes': {'Score': 100, 'Addr_Type': 'Locality'}}},
 'score': 100,
 'status': 'OK'},
 'bbox': [5.74022255, 53.136324803, 5.87022255, 53.266324803],
 'geometry': {'type': 'Point',
 'coordinates': [5.805222550000053, 53.201324803000034]}]}]}
centr_lat = a.geojson['features'][0]['properties']['lat']
centr_lon = a.geojson['features'][0]['properties']['lng']
pd.set_option('display.max_colwidth', 63)

map_herkomst = Map(location = [centr_lat, centr_lon],
                    zoom_start = 11,
                    tiles = "cartodbpositron"
                    )
# zet kolom geometry om in
list_features = []

for i, row in herkomst.iterrows():
    feature = Feature(geometry = row["geometry"],
                      id = row["hex_id"],
```

```

        properties = {"resolution": int(res)})
    list_features.append(feature)

feat_collection = FeatureCollection(list_features)
geojson_result = json.dumps(feat_collection)

GeoJson(
    geojson_result,
    style_function = lambda feature: {
        'fillColor': None,
        'color': ("green"
                ),
        'weight': 2,
        'fillOpacity': 0.05
    },
    name = "Herkomst"
).add_to(map_herkomst)

map_herkomst.save('maps/1_resolutions.html')
map_herkomst
Make this Notebook Trusted to load map: File -> Trust Notebook

# test, geocodering vinden van hexagon
lat,lng = h3.h3_to_geo('891fa5db657ffff')
geocoder.osm(f"{lat},{lng}").geojson
{'type': 'FeatureCollection',
 'features': [{'type': 'Feature',
  'properties': {'accuracy': 9.9999999995449e-06,
  'address': '21, Heike, Heikant Oost, Veldhoven, Noord-Brabant, Nederland, 5508 PA,
Nederland',
  'bbox': [5.3966914, 51.4249049, 5.3967914, 51.4250049],
  'confidence': 10,
  'country': 'Nederland',
  'country_code': 'nl',
  ' housenumber': '21',
  'importance': 9.9999999995449e-06,
  'lat': 51.4249549,
  'lng': 5.3967414,
  'municipality': 'Veldhoven',
  'ok': True,
  'osm_id': 2833409989,
  'osm_type': 'node',
  'place_id': 127269825,
  'place_rank': 30,
  'postal': '5508 PA',
  'quality': 'house',
  'quarter': 'Heikant Oost',
  'raw': {'place_id': 127269825,
  'licence': 'Data © OpenStreetMap contributors, ODbL 1.0.
http://osm.org/copyright',
  'osm_type': 'node',
  'osm_id': 2833409989,
  'lat': '51.4249549',
  'lon': '5.3967414',
  'category': 'place',
  'type': 'house',
  'place_rank': 30,
  'importance': 9.9999999995449e-06,
  'addresstype': 'place',
  'name': '',
  'display_name': '21, Heike, Heikant Oost, Veldhoven, Noord-Brabant, Nederland,
5508 PA, Nederland',
  'address': {'house_number': '21',
  'road': 'Heike',

```

```

    'quarter': 'Heikant Oost',
    'town': 'Veldhoven',
    'municipality': 'Veldhoven',
    'state': 'Noord-Brabant',
    'ISO3166-2-lvl4': 'NL-NB',
    'country': 'Nederland',
    'postcode': '5508 PA',
    'country_code': 'nl'},
    'boundingbox': ['51.4249049', '51.4250049', '5.3966914', '5.3967914']},
    'region': 'Noord-Brabant',
    'state': 'Noord-Brabant',
    'status': 'OK',
    'street': 'Heike',
    'town': 'Veldhoven',
    'type': 'house'},
    'bbox': [5.3966914, 51.4249049, 5.3967914, 51.4250049],
    'geometry': {'type': 'Point', 'coordinates': [5.3967414, 51.4249549]}}}]

```

Analyse herkomst - bestemming

op datum 15-09-2023 en bestemming Leeuwarden

```

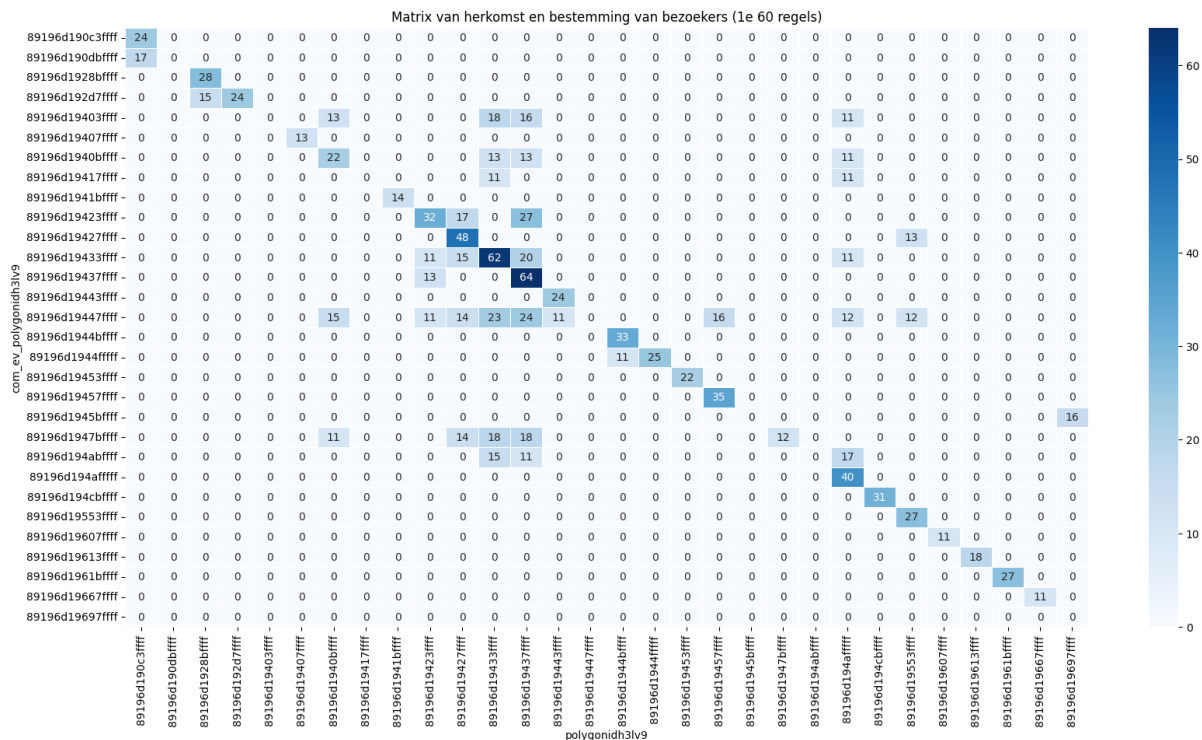
# Analyse herkomst - bestemming
con.rollback()
sql = """
SELECT polygonidh3lv9, com_ev_polygonidh3lv9, sum(bezoekers) as aantal_bezoekers
FROM data.v_celreport_lwd c
WHERE datum_bezoek = '15-09-2023'
GROUP BY polygonidh3lv9, com_ev_polygonidh3lv9
--ORDER BY aantal_bezoekers DESC
ORDER BY polygonidh3lv9 ASC
limit 60
"""

df = pd.read_sql(sql, con)
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# matrix maken
unieke_hexagons = pd.unique(df[['polygonidh3lv9',
'com_ev_polygonidh3lv9']].values.ravel('K'))
matrix = pd.DataFrame(np.zeros((len(unieke_hexagons), len(unieke_hexagons))),
columns=unieke_hexagons, index=unieke_hexagons)
# De matrix vullen met bezoekersaantallen
for index, row in df.iterrows():
    matrix.at[row['polygonidh3lv9'], row['com_ev_polygonidh3lv9']] =
row['aantal_bezoekers']

# Plot de matrix met een heatmap
plt.figure(figsize=(20, 10))
sns.heatmap(matrix, annot=True, fmt='.0f', cmap='Blues', linewidths=.5)
plt.title('Matrix van herkomst en bestemming van bezoekers (1e 60 regels)')
plt.xlabel('polygonidh3lv9')
plt.ylabel('com_ev_polygonidh3lv9')
plt.show()

```



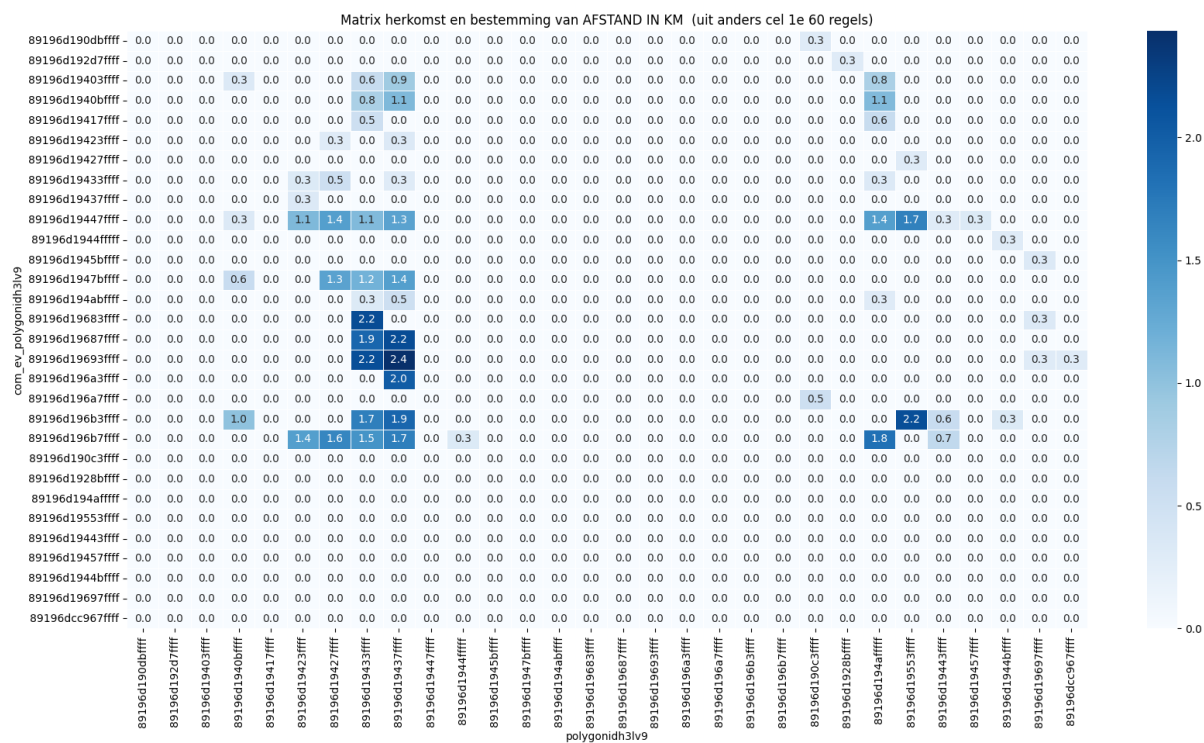
Conclusie. hexagons zijn vooral met zichzelf gecorreleerd!

Vergelijk de eerste 60 regels

```
# Analyse herkomst - bestemming (niet dezelfde CEL)
con.rollback()
sql = """
    SELECT polygonidh3lv9, com_ev_polygonidh3lv9, sum(bezoekers) as aantal_bezoekers
    FROM data.v_celreport_lwd c
    WHERE datum_bezoek = '15-09-2023'
    AND polygonidh3lv9 != com_ev_polygonidh3lv9
    GROUP BY polygonidh3lv9, com_ev_polygonidh3lv9
    ORDER BY polygonidh3lv9 ASC
    limit 60
    """

df = pd.read_sql(sql, con)
# matrix maken
unieke_hexagons = pd.unique(df[['polygonidh3lv9',
'com_ev_polygonidh3lv9']].values.ravel('K'))
matrix = pd.DataFrame(np.zeros((len(unieke_hexagons), len(unieke_hexagons))),
columns=unieke_hexagons, index=unieke_hexagons)
# De lege matrix vullen met bezoekersaantallen
for index, row in df.iterrows():
    matrix.at[row['polygonidh3lv9'], row['com_ev_polygonidh3lv9']] =
row['aantal_bezoekers']

# Plot de matrix met een heatmap
plt.figure(figsize=(20, 10))
sns.heatmap(matrix, annot=True, fmt='.0f', cmap='Blues', linewidths=.5)
plt.title('Matrix herkomst en bestemming van bezoekers UIT EEN ANDERE CEL (1e 60
regels)')
plt.xlabel('polygonidh3lv9')
plt.ylabel('com_ev_polygonidh3lv9')
plt.show()
```

zie ook: <https://python.plainenglish.io/exploring-openstreetmap-data-using-geopandas-d62b55fc40a4>
voeg analyse toe met thuis wedstrijd dagen Cambuur

con.close()

Handelingen gereedmaken data binnen PowerBI

De PIN en CEL data zijn geüpload in de DataFryslân Azure database. De data zijn aangeroepen via de ingebouwde SQL Server-database optie in de desktop versie van PowerBI:

SQL Server-database

Server ⓘ

Database (optioneel)

Gegevensverbindingsmodus ⓘ
☒ Importeren
☐ DirectQuery

▶ Geavanceerde opties

OK

Annuleren

Vervolgens zijn de PIN en CEL data ingeladen in PowerBI. Aan de achterkant ziet de ruwe data er als volgt uit:

CEL-data, na verwijdering gegevens over H3-levels en dergelijke

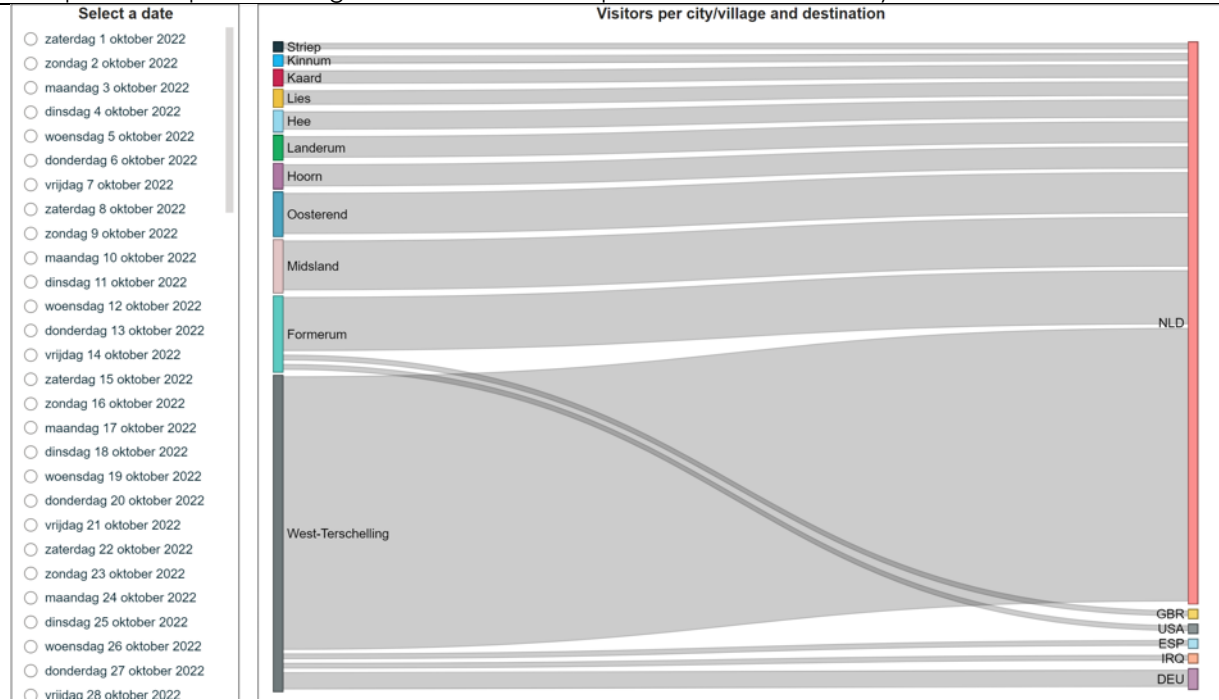
	A[C] woonplaats	A[C] gem_naam	A[C] prov_naam	A[C] common_evening_country	A[C] common_evening_municipality	visit_date	i23 visitors
1	Drachtstercompagnie	Smallingerland	Fryslân	NLD	Marum	29-10-2022	1
2	Wijnaldum	Harlingen	Fryslân	NLD	Haarlem	6-10-2022	1
3	Franeke	Waadhoeke	Fryslân	NLD	Ooststellingwerf	8-10-2022	1
4	Franeke	Waadhoeke	Fryslân	NLD	Harlingen	11-10-2022	1
5	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	30-10-2022	1
6	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	19-10-2022	1
7	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	29-10-2022	1
8	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	20-10-2022	2
9	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	12-10-2022	1
10	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	11-10-2022	1
11	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	2-10-2022	3
12	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	4-10-2022	1
13	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	25-10-2022	1
14	Franeke	Waadhoeke	Fryslân	NLD	Heerenveen	5-10-2022	1
15	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	11-10-2022	1
16	Franeke	Waadhoeke	Fryslân	NLD	Waadhoeke	9-10-2022	1

PIN-data, na verwijdering van kolommen die niet relevant zijn voor analyse

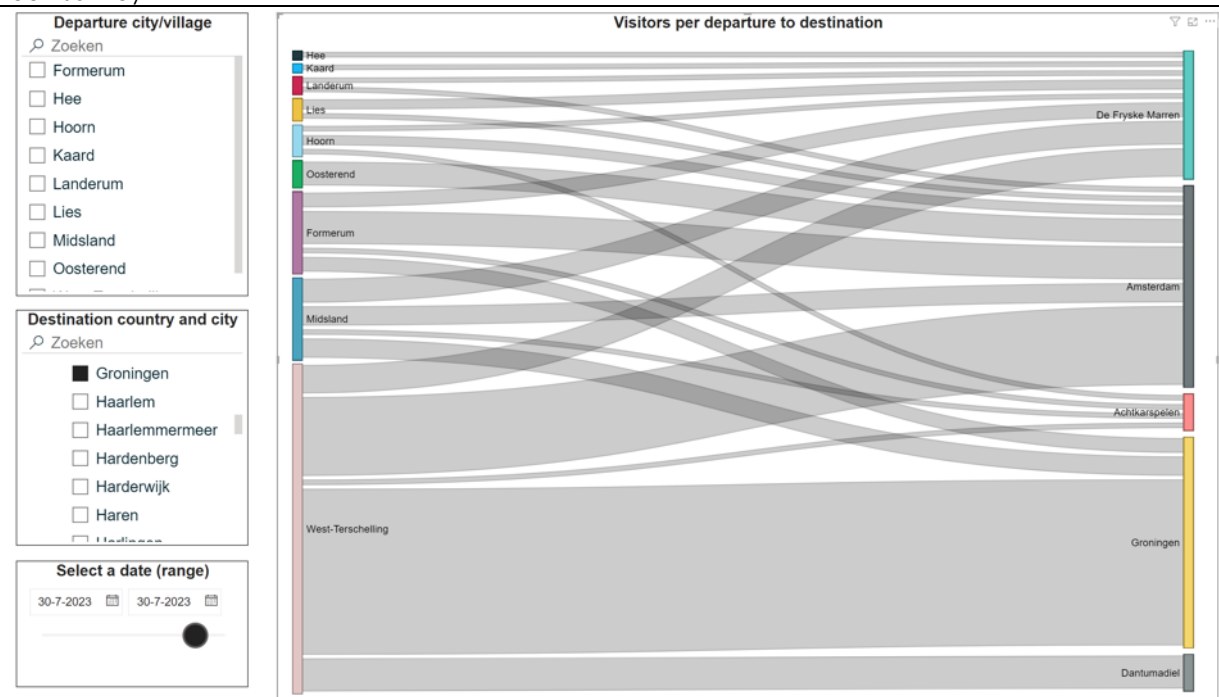
	A[C] Polygon_ID_Uber_H3_Level_8	Date	i22 Hour	i23 Visitors	A[C] DagNaam	A[C] MaandNaam	A[C] woonplaats	A[C] gem_naam	A[C] prov_naam
1	88194d8001ffff	1-10-2022		0	2 Zaterdag	oktober	Waterlandkerkje	Sluis	Zeeland
2	88194d8001ffff	1-10-2022		1	3 Zaterdag	oktober	Waterlandkerkje	Sluis	Zeeland
3	88194d8001ffff	1-10-2022		2	1 Zaterdag	oktober	Waterlandkerkje	Sluis	Zeeland

De front-end van PowerBI is gebruikt om de data geaggregeerd te visualiseren in lijndiagrammen, histogrammen, sankey-diagrammen, met behulp van diverse slicers. De acties in PowerBI betreffen handmatige handelingen gericht op het visualiseren van geografische data op een niet-geografische manier (alhoewel PowerBI wel mogelijkheden biedt om data geo-spatiaal te visualiseren). Vanwege de grootte van de bestanden werkte geo-spatiale visualisatie niet optimaal en vooral traag. Hieronder volgen de PowerBI tabbladen van de CEL en PIN-data in een statisch format. De dashboards zijn toegankelijk via [PIN](#) en [CEL](#).

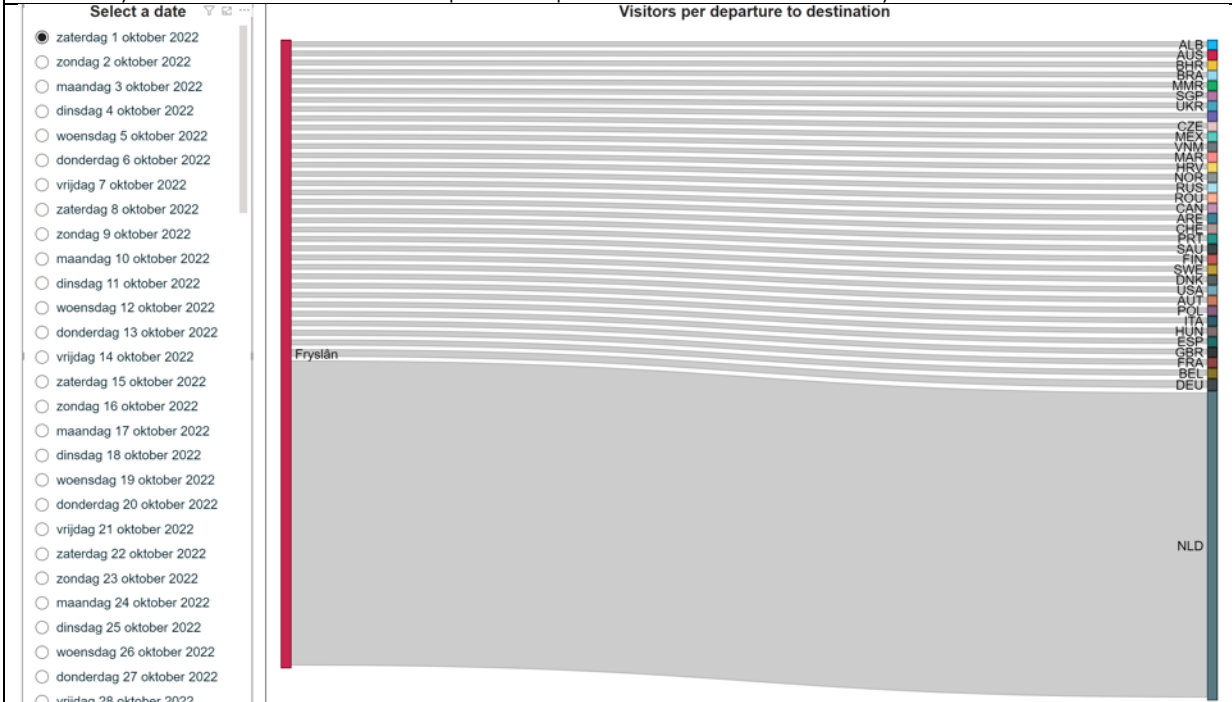
Van plaatsen op Terschelling naar een ander land per datum in een sankey



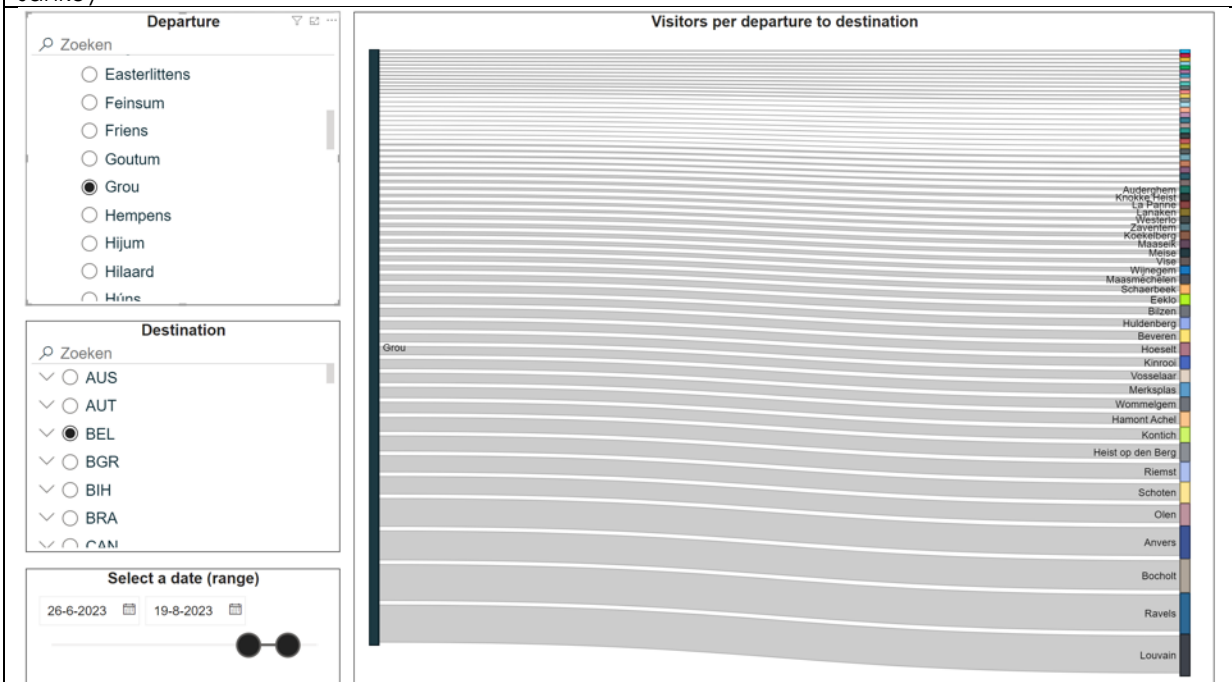
Van plaatsen op Terschelling naar andere steden in Nederland (of elders) op een bepaalde datum in een sankey



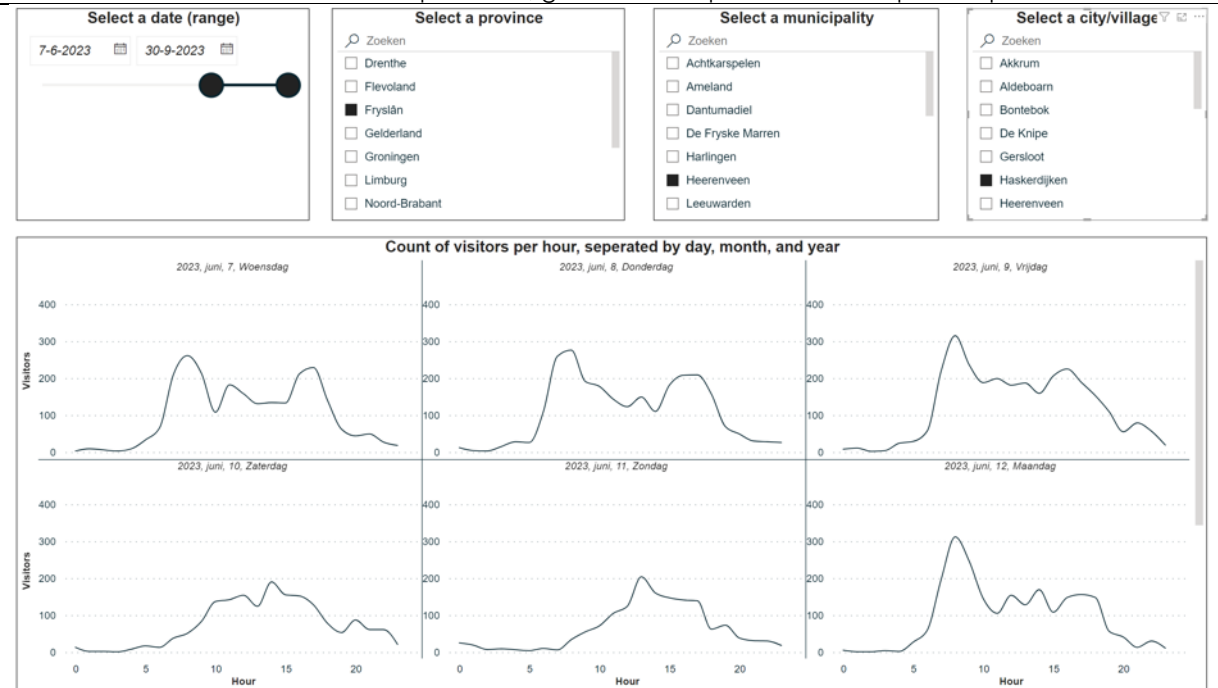
Vanuit Fryslân naar een ander land op een bepaalde datum in een sankey



Van een plek in Fryslân naar een andere plek in Nederland of elders op een bepaalde datum in een sankey



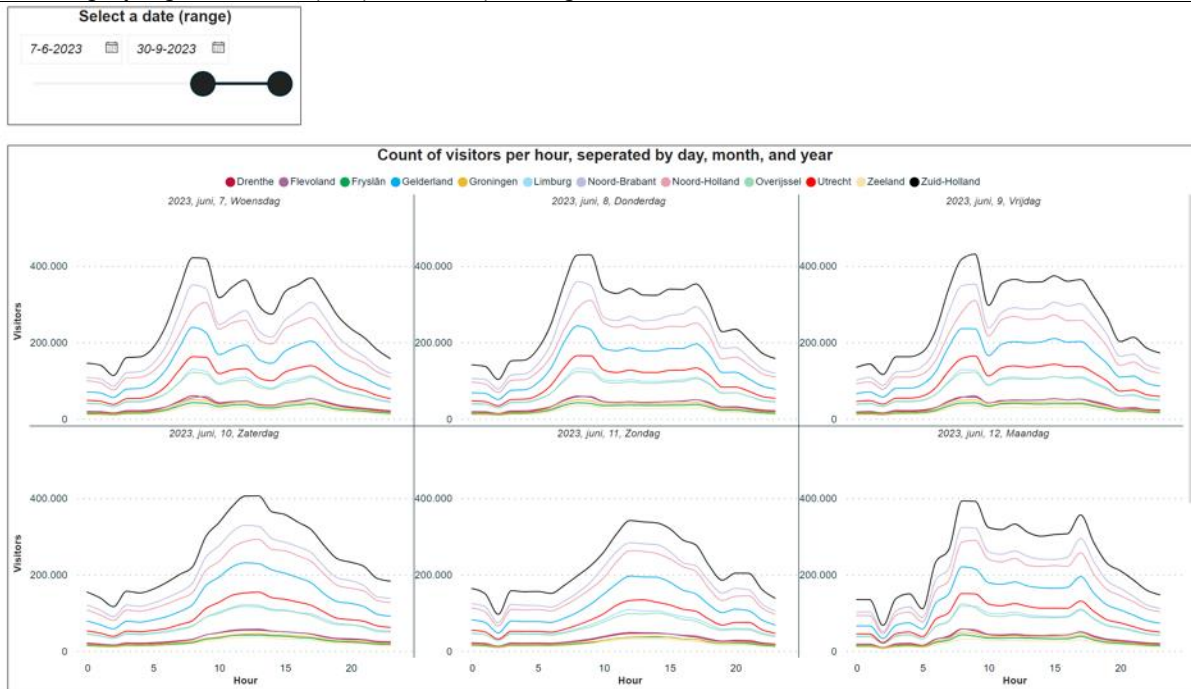
De distributie van bezoekers in een provincie, gemeente en plaats in een bepaalde periode



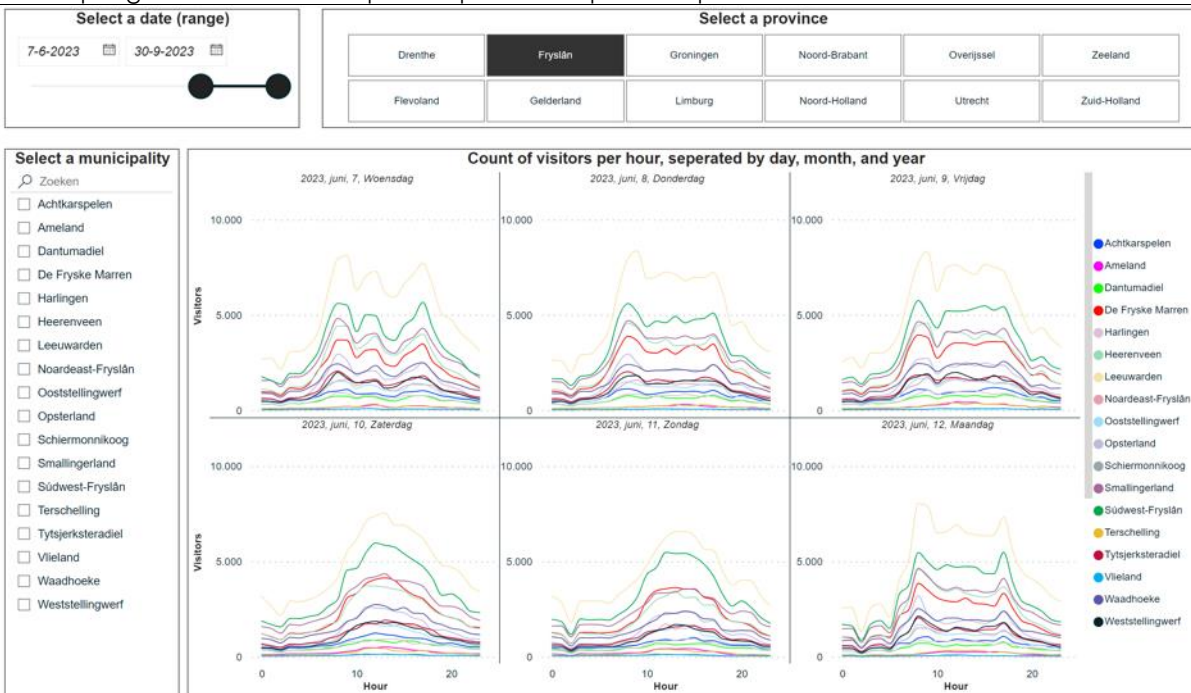
Ranglijsten met druktebeelden per provincie, gemeente en plaats op een bepaalde datum



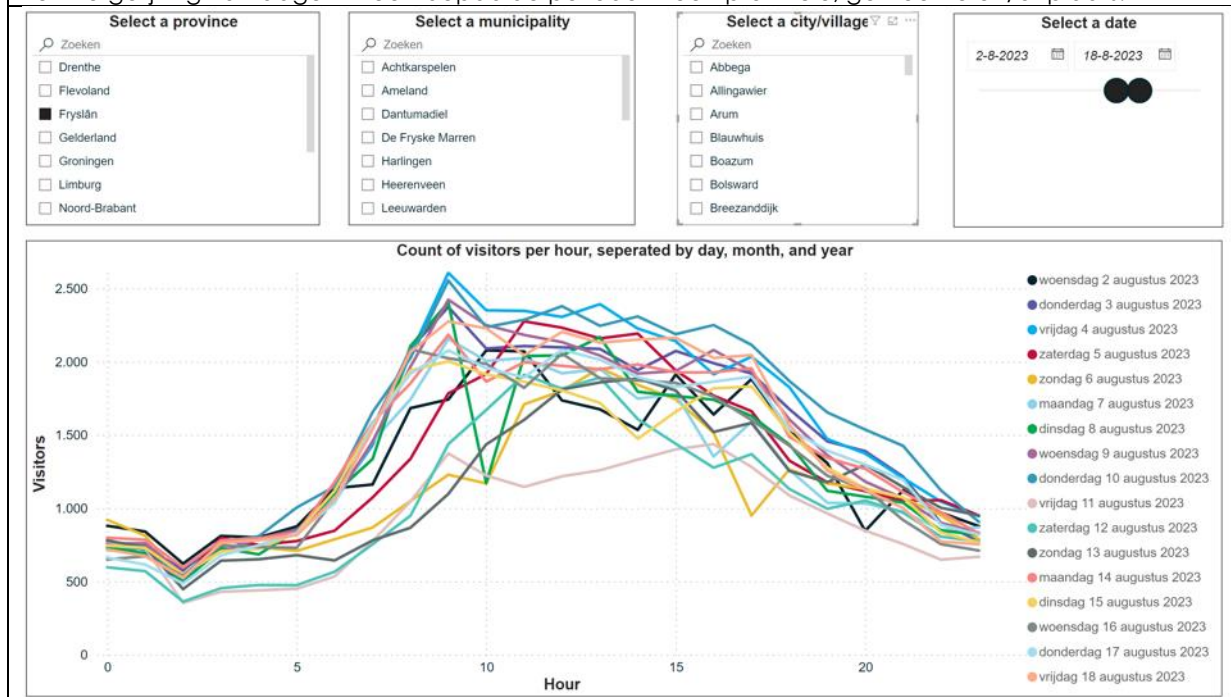
Een vergelijking van drukte per provincie, per dag



Drukke per gemeente in een bepaalde provincie op een bepaalde datum



Een vergelijking van dagen in een bepaalde periode in een provincie, gemeente en/of plaats.



Datapreparatie GPS toerisme in Postgres

CEL data verwerking in lokale postgres database.

De keuze voor lokale database is ontstaan door een bottleneck, doorlooptijd en procestijd. De CEL data bestanden zijn van een dusdanige grote dat de verwerking naar een azure database in de cloud via python jupyter notebooks te veel tijd in beslag nam. Hieronder een aantal cijfers van deze runs. Daarnaast liepen wij tegen proces problemen aan, die ook niet hebben bijgedragen in dit traject. Voor in de toekomst hebben we nu wel de mogelijkheden gevonden om de dataverwerking uit te kunnen voeren, zodat hierop informatieproducten gemaakt kunnen worden. Kanttekening: PIN data is wel gevuld in de Azure database. Deze werkzaamheden zijn opgepakt/uitgevoerd door Oswald Dijkstra

Cijfers dataverwerking via Jupyter notebook

Verwerking in azure db met algemene setup:

cijfers import met pyodbc, pandas en sqlalchemy

- batch van 120000, | 3/1084 [07:41<53:34:46, 178.43s/it] ---> geladen 240000
 - $178 \times 1084 / 60 = 3.215 / 60 = 53$ uur

Verwerking in azure db na opvoeren database:

P4 Azure MSSQL DB

- | 37/1182 [36:22<19:55:45, 62.66s/it] ---> geladen 3850000
- | 39/1182 [38:52<21:50:19, 68.78s/it] ---> geladen 4180000
 - $65 \times 1182 / 60 / 60 = 21$ uur

Na het opvoeren van de azure database een aanzienlijke versnelling van 75 uur naar 21 uur verwerking.

Keuze lokale database

De keuze voor de dataverwerking in een lokale database is genomen op basis van de volgende redenen:

1. Tijdsbestek, verwerking bestanden naar azure database
 - a. Hierbij hadden wij geen rechten op de azure database om bulk import uit te voeren.
2. Bottleneck van tijd algemeen op de verwerking via de python jupyter notebook.

Database opbouw en doel

Doel database

- Datasets voor onderzoek voorbereiden (datapreparatie en transformatie)
 - Extract van specifieke datum/locatie voor onderzoek
- Combineren met aanvullende data:
 - Bv GEO locaties op basis van de h3 bibliotheek
 - Koppelen h3 gebieden aan provincie, gemeenten en woonplaatsen
 - Koppelen iso3 landen aan de werledlanden

Verwerking van de data:

- Aanmaken geodatabase in postgres
- Aanmaken staging & transform schema's
- CSV bestanden inlezen in staging schema, bulk import functionaliteit van postgres hiervoor gebruikt.

Databasestructuur

- Stg schema
 - 33 tabellen en 2 views
 - PIN en CEL 12 maanden
 - H3 geodata NL
 - NL gemeenten grenzen
 - NL Provincies grenzen
 - NL Woonplaatsen grenzen
 - World landsgrenzen
 - v_uni_CEL
 - v_h3_pgw
- Trf Schema
 - 2 tabellen en 2 views
 - t_sneek_uni_CEL
 - t_terschelling_uni_CEL

- v_terschelling_uni_CEL_geom
- v_terschelling_uni_CEL_kepler
- Transformatie en samenvoeging
 1. Union van de PIN en CEL data
 2. Uitsnede op basis van woonplaats en gemeente en tijd.
 - Voorbeeld: Gemeente Terschelling en woonplaats Sneek

Dataextract voor onderzoek en visualisatie

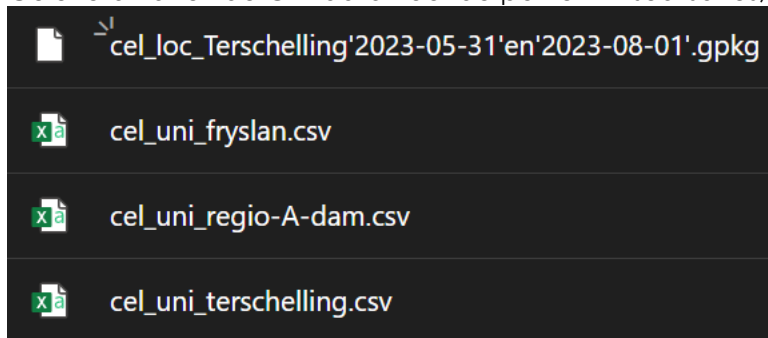
Afgesproken regio's en tijdsbestek.

CEL data (herkomst en bestemming), is alleen op dag niveau, selecties van :

- Provincie Fryslân
- Eiland Terschelling
- Stadregio Amsterdam

(mogelijk de top 10 aantal visitors voor deze punten)

Geleverd vanuit de CEL data voor de powerBI visualisaties, Qgis en KeplerGL map's.



Datavisualisatie in Qgis en KeplerGL

Op basis van de datapreparatie zijn de geometrische locaties geaggregeerd naar gemeenteniveau en/of wereldland niveau, idem het aantal visitors gesommeerd per dag en per aggregatie niveau.

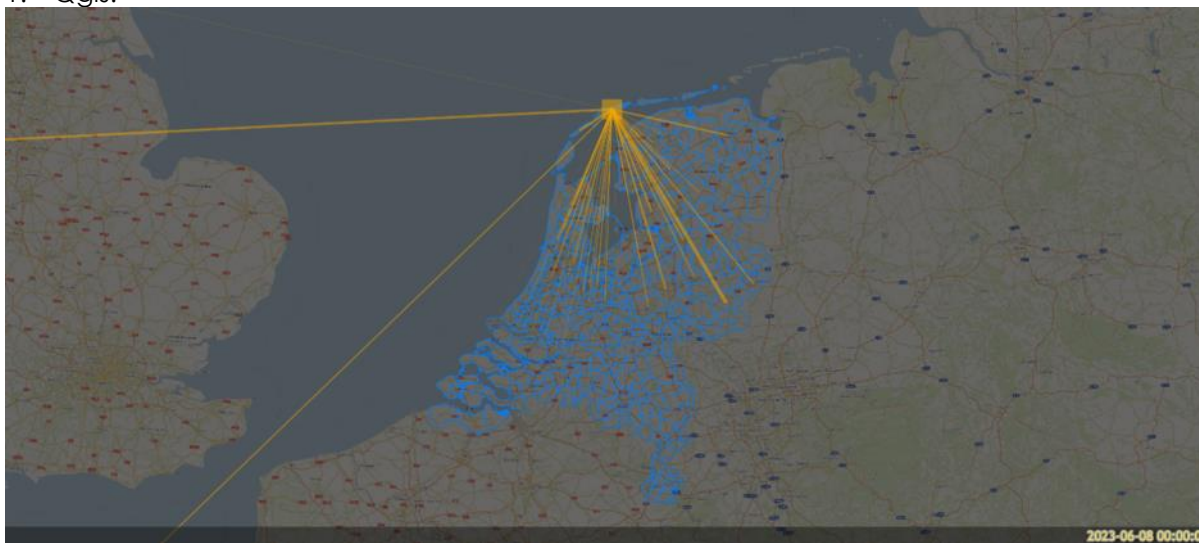
Preparatie voor visualisatie

Voor de visualisatie in Qgis is een datadoorsnede(extarct) gemaakt in de vorm van een geopackage bestand. Deze kan ingelezen worden in Qgis. In Qgis wordt de opmaak gemaakt.

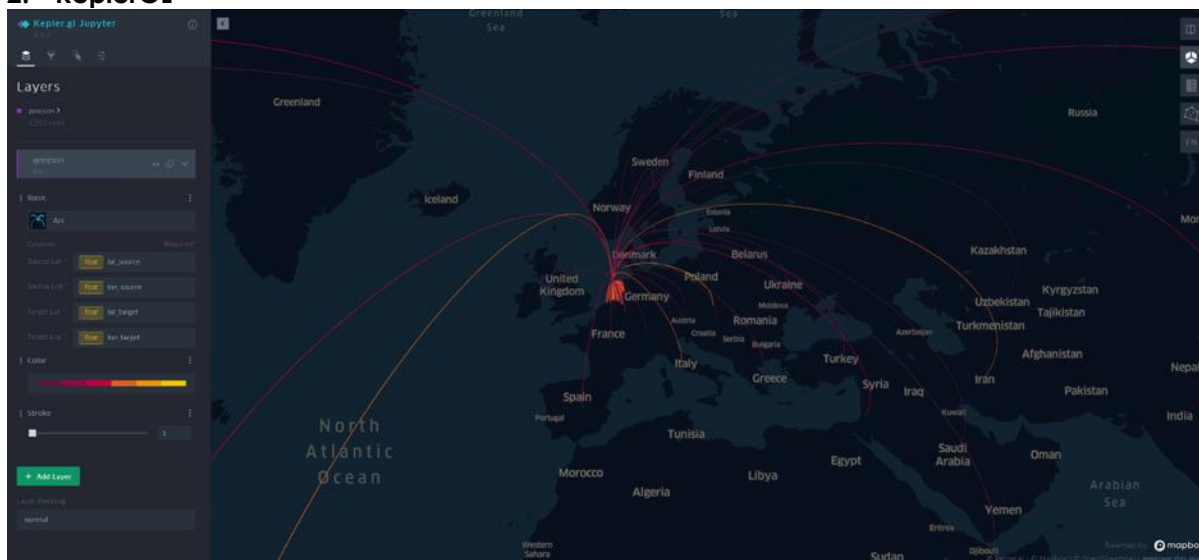
Voor de visualisatie in KeplerGL wordt er gebruik gemaakt van een jupyter notebook, waarin de keplergl bibliotheek gebruikt wordt. Met behulp van deze bibliotheek worden de map's aangemaakt en geëxporteerd naar een html bestand, welke in een internetbrowser geopend kunnen worden.

Visualisaties

1. Qgis:



2. KeplerGL



Bijlagen

1. Jupyter lab

- filter_nearh3_gps-toerisme.ipynb
- kepler_map_jupyter.ipynb
- read_load_csv-postgres.ipynb

2. Pgadmin SQL

- PGSQL_create_stg-tables.sql
- PGSQL_create_trf-tables.sql
- PGSQL_create_datamart_terschelling.sql