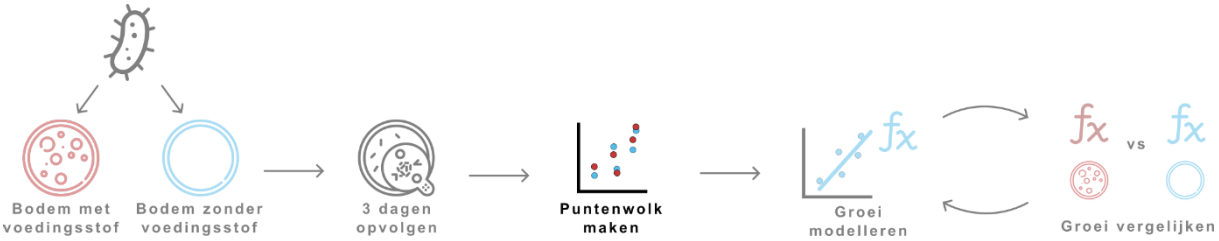
Deelprobleem 3

Visualiseer de opgemeten data.



Deze sessie gaan jullie aan de slag met de foto's die jullie de voorbije dagen hebben gemaakt. Eerst kiezen jullie enkele kolonies uit waarvan je de groei goed kan volgen. Vervolgens gebruiken jullie een online tool om te meten hoe groot elke kolonie is. Dit levert nuttige data op, die jullie met behulp van Python kunnen weergeven als een puntenwolk.

STAP 1: KOLONIES KIEZEN

Op jullie foto's zijn enorm veel kolonies te zien. Sommige van deze kolonies raken elkaar ook aan. Is dit erg?

Kies zowel voor de testcultuur als de controlecultuur telkens twee kolonies uit waarvan je de groei gemakkelijk kan zien op basis van jullie foto's. Waarom kiezen jullie deze kolonies? Als jullie deze vier kolonies hebben gevonden, roep er dan een leraar bij en leg uit op basis waarvan je je keuze hebt gemaakt.

STAP 2: GROEI METEN

De uitgekozen kolonies zien er uit als cirkels. Als we willen weten hoeveel een bepaalde kolonie is gegroeid, kunnen we kijken hoe de oppervlakte van de kolonie verandert. Om deze oppervlakte nauwkeurig te kunnen meten, kan je https://ij.imjoy.io/ gebruiken. Verdeel de foto's goed over jullie groep zodat iedereen kan meewerken.

Hier vind je een stappenplan:

1. Klik op *File* → *Open…* → *Select local file*
2. Kies de foto die je wil analyseren. Zorg dat je goed weet welke chronologisch het eerst komt en om welke kolonie het gaat!
3. Klik op de rechte lijn en teken de lijn zo dat die exact één millimeter lang is. Hoe ga je hiervoor zorgen?
4. Klik op *Analyze* → *Set scale…*
5. Vul bij "*Known distance*" de afstand van het lijnstuk in, en vul bij "*unit of length*" de eenheid van de afstand in. Klik daarna op OK. Nu weet het programma welke lengte in pixels overeenkomt met een exacte lengte!
6. Klik op de ellips, teken een ellips en versleep de punten zodat de gekozen kolonie zo goed mogelijk wordt benaderd door de ellips.
7. Klik op *Analyze* → *Measure*. Hier vind je een heleboel data over de ellips. Waar is de oppervlakte? In welke eenheid staat dit?

8. Voer deze oppervlakte in de Excel file in. **Gebruik een punt (.) in plaats van een komma (,) als je met kommagetallen werkt.** Zorg dat dezelfde kolonie steeds in dezelfde kolom staat. **Vul ook de datum en tijdstip van de foto exact in zoals aangegeven.**
9. Herhaal stappen 6-8 voor de tweede geselecteerde kolonie.
10. Herhaal het hele proces voor alle foto's.

STAP 3: PUNTENWOLK MAKEN

Splits jullie op in groepjes van twee. Per twee gaan jullie samen code schrijven om ofwel de testdata, ofwel de controledata te visualiseren als een puntenwolk.

Stap 3.1: Een csv-bestand exporteren uit Excel

Alle data staat nu in Excel op OneDrive. Wij willen er echter lokaal, op onze eigen laptop, mee werken. Je kan ook niet zomaar met een Excel-bestand werken in Python, dus hier moeten we ook iets op vinden.

Volg dit stappenplan:

1. Open het Excel bestand met de groeidata door erop te klikken in OneDrive.
2. Ga onderaan naar het juiste tabblad. Gebruik het ene tabblad voor de testkolonie en het andere voor de controlekolonie. Haal "*Testkolonie*" en "*Controlekolonie*" niet door elkaar!
3. Klik linksboven op *File → Export →Download as CSV UTF-8*.
4. Sla het bestand op op een logische plek op je laptop. Zorg dat je het later terug kan vinden!

Ga naar het bestand dat je net hebt gedownload en open het. Beantwoord de volgende vragen.

- Hoe ziet het bestand eruit? Wat valt op?

- Wat is er speciaal aan de eerste regel van het bestand? Vergelijk dit met de originele tabel in Excel.

- Dit is een csv-bestand. Waar staat "csv" voor? Waar zijn deze bestanden nuttig voor? Zoek dit op op het internet.

Dit nieuwe csv-bestand kunnen we heel eenvoudig gebruiken binnen Python. Op OneDrive vinden jullie een Python-bestand dat al een groot deel van de code bevat die dit csv-bestand krijgt als invoer en een puntenwolk geeft als uitvoer. Download dit bestand. **Zorg ervoor dat dit in dezelfde folder staat als het csv-bestand.**

Stap 3.2: Een csv-bestand inlezen als dataframe in Python

Open het Python-bestand. De meesten lijnen moeten jullie zo meteen zelf nog aanpassen. Als dit nodig is, staat er steeds *#TODO* geschreven. Bovenaan staan enkele softwarebibliotheken die we gaan gebruiken. Matplotlib helpt ons om grafieken te tekenen. Pandas wordt overal ter wereld gebruikt door datawetenschappers om grote hoeveelheden informatie te analyseren. Hiervoor wordt een nieuw datatype gebruikt: **een dataframe**.

Op lijn 7 wordt een functie gebruikt die zo'n dataframe maakt: *read_csv*. Deze functie krijgt als invoer de naam van een csv-bestand en geeft als uitvoer een dataframe. **Pas zelf de invoer op lijn 7 aan** en voer de code uit. Wat wordt er geprint? Herken je deze structuur?

In een dataframe heb je rijen en kolommen. Al deze kolommen hebben een naam. Net zoals bij een dictionary, kan je een hele kolom van een dataframe tegelijk selecteren door vierkante haakjes te gebruiken en daarbinnen de juiste naam te gebruiken.

Voorbeeld

Stel dat ik een dataframe heb met de naam *data* dat er als volgt uitziet:

	X-waarde	Y-waarde
0	1	2
1	2	4
2	3	8
3	4	16

Als ik *data[Y-waarde]* print, krijg ik het volgende:

	Y-waarde
0	2
1	4
2	8
3	16

Door vierkante haakjes te gebruiken, kan je dus één kolom van een dataframe aanpassen of bekijken. Deze kolom gedraagt zich zoals een lijst, dus je kan die ook indexeren.

Stap 3.3: Het datatype van een tijdstip

In de eerste kolom van *df* staat telkens een datum en een tijdstip. Jammer genoeg ziet Python dit niet effectief als een echte datum, maar als een gewone string. Als we deze kolom willen gebruiken om mee te rekenen, moeten we dit dus aanpassen. Ook hier heeft pandas een functie voor: *to_datetime*. Deze functie neemt als invoer een kolom en geeft een nieuwe kolom terug, die nu wel wordt beschouwd als een datum en een tijdstip. Het datatype wordt dus aangepast. De nieuwe kolom kunnen we ook toevoegen aan de originele dataframe.

Op lijn 10 zie je dat de *to_datetime* functie gebruikt wordt, met nog een extra *format* argument. Wat denk je dat dit betekent?

Vul nu de code op lijn 10 aan door het juiste argument mee te geven. Denk goed aan welke kolom we willen aanpassen!

Nu hebben we een kolom in onze dataframe waarmee we een datum en een tijdstip weergeven. We noemen dit een "*datetime*" datatype. Python interpreteert al deze data nu correct en kan er ook mee rekenen.

Wij zijn echter niet meteen geïnteresseerd in de exacte data en tijdstippen, maar in hoeveel tijd er voorbij is gegaan tussen verschillende metingen. Ook hier zijn er ingebouwde functionaliteiten voor in Python: als je een *datetime* aftrekt van een andere *datetime*, krijg je een *timedelta* object. Dit geeft een verandering in tijd aan, dus de breedte van een tijdsinterval.

Een kolom van een pandas dataframe kan je gemakkelijk volledig aanpassen. Op elk element in deze kolom kan je een bepaalde wiskundige operatie uitvoeren, zoals in het voorbeeld hieronder.

Voorbeeld

Stel dat ik een dataframe heb met de naam *data* dat er als volgt uitziet:

	Y-waarde
0	2
1	4
2	8
3	16

Als ik de lijn code
data[Y-waarde] = data[Y-waarde] + 3
uitvoer, zal er bij elke waarde in de kolom het getal 3 opgeteld worden.

	Y-waarde
0	5
1	7
2	11
3	19

Gebruik nu alles wat je hebt geleerd om een nieuwe kolom te maken, “Voorbije tijd”. Denk goed na over welk tijdsverschil je wil bepalen! **Pas zelf regel 11 in het bestand aan.**

Print deze nieuwe kolom. Wat valt er op?

Als we een grafiek willen maken van data, is het handiger om met één exact getal te werken voor elke as. De functie *dt.total_seconds()* verandert een timedelta object naar een aantal seconden. Bepaal zelf een tijdseenheid op basis waarvan je overzichtelijk kan zien hoeveel tijd er verstreken is. Dit is de grootheid die op één van de assen gaat komen te staan in de uiteindelijke puntenwolk.

Vul regel 12 aan en maak een nieuwe kolom. Denk zelf na hoe je een aantal seconden omvormt in een andere tijdseenheid! Kies ook een gepaste naam voor deze kolom.

Stap 3.4: Data visualiseren als puntenwolk

De data is verwerkt, nu kunnen we het eindelijk visualiseren! De functie *df.plot()* kan hiervoor gebruikt worden. We geven drie argumenten mee. Het eerste, (*kind* = *scatter*), vertelt aan de functie dat we een puntenwolk willen maken. De andere argumenten, met namen *x* en *y*, geven aan wat er op de horizontale en verticale assen moet komen. Vul hier zelf de namen van de kolommen die je wil plotten. **Vul regel 14 aan**.

Regel 15 zorgt ervoor dat deze plot ook effectief getekend wordt. Voer de code uit. Normaal krijg je een mooie puntenwolk! Denk na of deze er uit ziet zoals je ongeveer verwacht. Het zou kunnen dat je nog wat moet debuggen.

De puntenwolk verschijnt in een nieuw venster. Normaal zie je hier onderaan een soort “floppydisk” (zie screenshot hieronder). Als je hierop klikt, kan je een afbeelding opslaan op je computer. Maak één puntenwolk voor elke kolonie die je gevolgd hebt. Elk groepje zou dus **minstens vier puntenwolken** moeten hebben.



In het volgende deelprobleem gaan we volop verder met al deze data en gaan we een wiskundig model zoeken dat het gedrag verklaart.