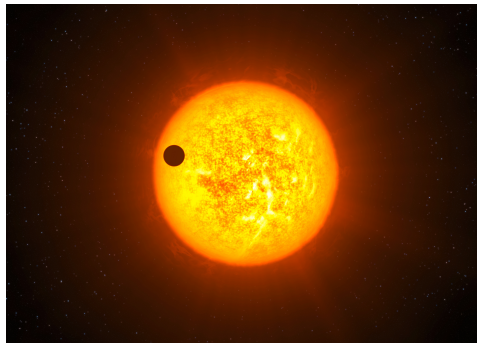


Exoplaneter

Opgaverne i dette lille projekt skal træne dig i at skrive funktioner i din kode. Til det formål skal vi arbejde med et stort dataset, der omhandler exoplaneter.



Exoplaneter

En exoplanet er en planet, der er i kredsløb om en stjerne udenfor vores solsystem. I 1992 blev opdagelsen af de første exoplaneter bekræftet, og siden opsendelsen af rumteleskopet Kepler i 2009, er antallet af kendte exoplaneter eksploderet. I dag mener astronomer at have bekræftet opdagelsen af over 3000 exoplaneter.

Vi skal arbejde med et stort dataset, der indeholder en lang række oplysninger om 3426 exoplaneter. Datasættet er så stort at man ikke rigtig får noget ud af at kigge på det manuelt, så i stedet skal vi bruge programmering til at undersøge og visualisere noget omkring alle disse exoplaneter.

Datasættet

Datasættet er en såkaldt *kommasepareret* fil, med navet `oec.csv`, og det stammer fra *Open Exoplanet Catalogue*¹. Den slags data kan for eksempel analyseres i et regneark, og så ser det ud som på figur 1. Den første række er nogle ord, der beskriver indholdet af hver kolonne, og fra den anden række begynder oplysningerne om exoplaneterne, én exoplanet pr række.

¹<http://www.openexoplanetcatalogue.com/>

	B	C	D	E	F	G	H	I	J
1	TypeFlag	PlanetaryMassJpt	RadiusJpt	PeriodDays	SemiMajorAxisAU	Eccentricity	PeriastronDeg	LongitudeDeg	AscendingNodeDeg
2	0	0.0014	0.054	0.1768913	0.0048				
3	0		0.114	4.194.525	0.039				
4	0		0.071	6.356.006	0.052				
5	0	0.25	0.84	1.922.418	0.143	0.0626			
6	0	0.17	0.82	3.903.106	0.229	0.0684			
7	0	0.022	0.147	1.592.851	0.0271				
8	0	0.0321		52.354	0.053	0.06	350		
9	0		0.192	1.842.794.624					
10	0	0.6	1.24	337.265	0.0449				
11	0	5.21		501.75	1.33	0.15	277.59		
12	0	0.017		3500	2.1				
13	0		0.172	335.183.158					
14	0		0.256	575.322.197					
15	0	1.24	1.67	2.705.782	0.0447				
16	0	0.0195	0.2525	158.040.417	0.01488	0			
17	0		0.2661	18.870.227					
18	0		0.23056	4.690.232					
19	0	1.82		437.05		11.735	0.52		
20	0	2.17		6700	6.95	0.24			
21	0	0.08		14.476	0.11	0.115			
22	0		0.147	0.67564949					
23	0	0.5074		30.965.886	0.04213	0.0093		360	
24	0	1.818		3324		4.417	0.28	258.5	
25	0		0.178	879.617.863					
26	0		0.136	799.019.627					
27	0		0.0682	6.803	0.0719				
28	0		0.0596	8.703	0.0847				
29	0		0.099	11.922	0.1045				
30	0	0.05074	0.219624	16.092					
31	0	0.0750657	0.076540	755.120					

Figur 1: Et udsnit af datasættet

Øvelse 1. Åbn filen `oec.csv` i et regnearksprogram, og prøv at få en idé om hvad det er for noget data vi skal arbejde med.

Der er mange tomme felter i datasættet, hvilket simpelthen skyldes at forskerne endnu ikke har kunnet fastlægge de oplysninger.

For at kunne analysere vores data, skal vi kende betydningen af kolonnerne. Jeg vil ikke gennemgå dem alle, men herunder følger en forklaring til nogle af de mest interessante.

PlanetIdentifier er exoplanetens navn. De er typisk navngivet efter hvordan de er blevet opdaget.

PlanetaryMassJpt, RadiusJpt angiver data om exoplanetens størrelse og masse. Begge angives i enheder i forhold til Jupiter.

PeriodDays, SemiMajorAxisAU angiver oplysninger om exoplanetens bane, det vil sige dens omløbstid om stjernen og størrelsen på den elliptiske bane.

RightAscension, Declination, DistanceFromSunParsec angiver exoplanetens position i forhold til vores solsystem. De to første er vinkler, den sidste er en afstand.

Datasættet i python

Jeg har forberedt et lille program, der hjælper dig med at få gang i datasættet. Det er en af den slags opgaver, der er meget nemmere, hvis man bruger et forudprogrammeret bibliotek. Vi bruger biblioteket `csv`, der er indbygget i Python².

```
1 #Her importeres biblioteket
2 import csv
3
4 #En global variabel til at holde alle data
5 data = None
6
7 def setup():
8     global data
9     #Filen åbnes
10    infile = open('oec.csv', mode='r')
11    #Data læses af biblioteket
12    reader = csv.DictReader(infile)
13
14    print(reader.fieldnames)
15    #Vi laver vores data om til en liste af dict-
16    #variable
17    data = list(reader)
```

Vores variabel `data` indeholder nu alle data fra filen! Vi kan nu undersøge datasættet ved at bruge nogle af de ting vi allerede har lært om `dict`-variable.

Du kan sikkert forestille dig, at koden hurtigt bliver lidt rodet, hvis man skal behandle alle de her data, ved at lave `for`-løkker hver gang man skal undersøge en kolonne eller række. For at holde koden overskuelig, skal den selvfølgelig inddeles i nogle funktioner. Det næste eksempel er en funktion, der undersøger alle exoplaneterne, og udregner deres gennemsnitlige masse:

```
1 #Denne funktion udregner den gennemsnitlige masse for
2 #alle exoplaneterne
3 def getAverageMass():
4     global data
5     totalMass = 0
6     numberOfPlanets = 0
7
8     #for-løkken gennemløber hele datasættet
9     for row in data:
```

²Læs mere på: <https://docs.python.org/3/library/csv.html>

```

9      #Den kolonne der hedder "PlanetaryMassJpt"
      indeholde massen, målt i enheden "
      jupitermasser".
10     #Hvis massen ikke findes i datasættet får m
      værdien '' (En tom streng).
11     #Derfor er vi nødt til at undersøge om m har
      en gyldig værdi,
12     #før vi tæller planeten med:
13     if row['PlanetaryMassJpt'] != '':
14         m = float(row['PlanetaryMassJpt'])
15         numberOfPlanets += 1
16         totalMass += m
17
18
19     #Nu kan gennemsnittet udregnes:
20     return totalMass/numberOfPlanets

```

Øvelse 2. Hent den samlede kode, og start programmet op. Hvis alt virker, og du forstår nogenlunde hvordan funktionen `getAverageMass` virker, så er du klar til at gå videre!

Husk at placere filen `oec.csv` i samme mappe som programmet.

Øvelse 3. Nu skal du selv igang med at kode. Brug funktionen `getAverageMass` som udgangspunkt, og lav en funktion der udregner og returnerer exoplaneternes gennemsnitlige afstand til vores stjerne, Solen.

`getAverageDistanceFromSun()`

De relevante data står i kolonnen 'DistFromSunParsec'. En parsec er en absurd afstandsenhed, der kun er relevant når vi taler om størrelser udenfor vores solsystem.

$$1 \text{ pc} \approx 3.0856776 \cdot 10^{16} \text{ m}$$

Det tal er ca. 14 millioner gange større end det største tal der kan stå i en `int`-variabel. Så er det ret praktisk at regne i parsec i stedet for...

Øvelse 4. Noget af det der er interessant med exoplaneter er, at undersøge om de er sammensat på samme måde som de planeter vi kender fra vores

eget solsystem. Ved hjælp af kolonnerne `PlanetaryMassJpt` og `RadiusJpt` kan vi udregne exoplaneternes densitet. Skriv først funktionen

`getPlanetDensity(i)`

der returnerer densiteten for den planet man beder om med parameteren `i`. (Læg mærke til at der er mange exoplaneter, hvor man ikke kender begge de nødvendige oplysninger. I de tilfælde kan din funktion returnere resultatet `-1`.)

Derefter skal den funktion bruges af en anden funktion

`getAverageDensity()`

der finder alle de planeter hvor vi kender densiteten, og returnerer gennemsnittet.

Øvelse 5. Lad os se på nogle af de ekstreme planeter. Prøv at lave nogle af følgende funktioner

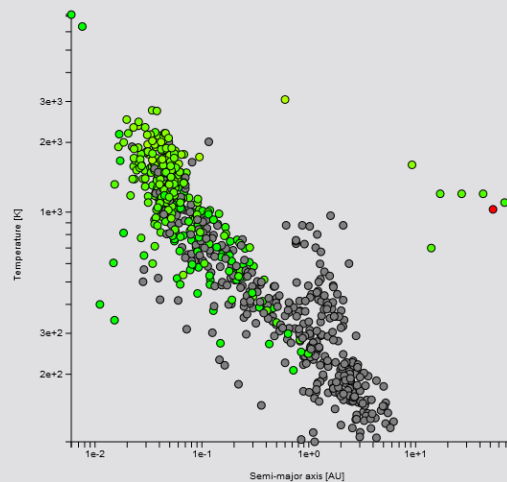
`getHeaviestPlanetMass()` der returnerer massen på den tungeste exoplanet vi kender.

`getLargestPlanetRadius()` der returnerer radius på den største exoplanet vi kender. Er det mon den samme som den tungeste?

`getMostDistantPlanet()` der returnerer afstanden fra solen til den fjerneste exoplanet vi kender.

Hvis du vil undersøge disse ekstreme planeter nærmere, kan du få dine funktioner til at udskrive noget information undervejs, for eksempel det index i datasættet, hvor resultatet blev fundet.

Øvelse 6. Når man skal sige noget generelt om exoplaneterne, kan man for eksempel kigge på såkaldte *korrelationer* mellem nogle af alle de data vi har. Det vil egentlig bare sige, at man vil undersøge om der er en sammenhæng mellem to ting. Det gøres lettest ved at tegne en hel masse punkter i et koordinatsystem, og lede efter grafiske sammenhænge. Se her:



På grafen kan man se, at der nok er en sammenhæng mellem planeternes afstand til deres stjerne, og temperatur. Ikke så overraskende.

Genskab grafen i dit program.

Lav evt. selv flere undersøgelser af sammenhængen mellem to udvalgte kolonner i datasættet.

Øvelse 7. Når vi sidder her på jorden og undersøger disse fjerne planeter, går vi ud fra at fysikkens love er de samme alle steder i universet. En af de fysiske love der er relevante for planeter, er Kepler's tredje lov, der siger at

$$p = k \cdot a^{3/2}$$

hvor p er planetens omløbstid og a er planetens middelfastand til stjernen. De to oplysninger hedder `PeriodDays` og `SemiMajorAxisAU` i datasættet.

Gælder Keplers tredje lov for exoplaneterne?

For at undersøge dette spørgsmål kan du

- Tegne et punkt for hver exoplanet i et (p, a) koordinatsystem.
- Undersøge størrelsen af k for exoplaneterne. *Pas på! k skal ikke have samme værdi for alle kombinationer af stjerner og planeter.*
- Huske at potensfunktioner bliver til rette linjer, hvis man tager logaritmen til både den afhængige og uafhængige variabel.

Denne opgave er meget åben, og temmelig svær at finde et rimeligt svar på. Spørg endelig om hjælp, hvis du går i stå!