Cheatsheet eksempler

Indholdsfortegnelse

> **Interaktive plots**
>
> Hvis interaktive plots ønskes i VScode skal følgende Python kommando køres som det første i ens notebook. Dette kræver biblioteket `ipympl` (hentes ved `pip install ipympl` i terminal).
>
> ```
> %matplotlib widget
> ```

`gym_cas` indlæses

```
from gym_cas import *
```

B1. Tal- og bogstavregning

```
expand( (x+3) ** 2)
```

$x^2 + 6x + 9$

```
factor( x**2 + 6*x + 9 )
```

$(x + 3)^2$

B2. Ligninger og uligheder

```
solve(x**2+3.*x-2)
```

```
[-3.56155281280883, 0.561552812808830]
```

```
solve([a+b-2,a-b-4])
```

```
{a: 3, b: -1}
```

```
nsolve(2*x**2-x,0.5)
```

0.5

B3. Geometri og trigonometri

```
Sin(90)
```

1.0

```
aSin(1)
```

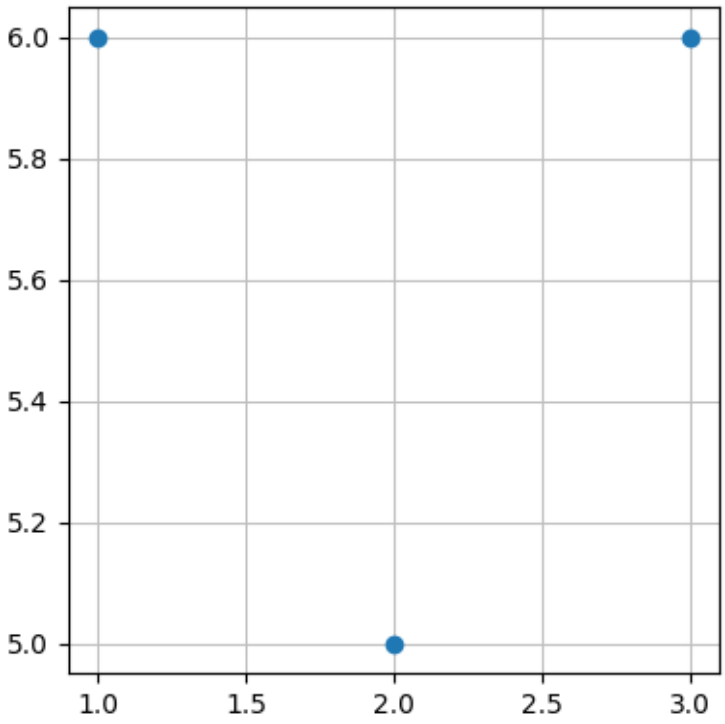
90.0

B4. Analytisk Plangeometri

Figurbeskrivelser kan laves med `#| fig-cap: "Beskrivelse"` i kodeblokke.

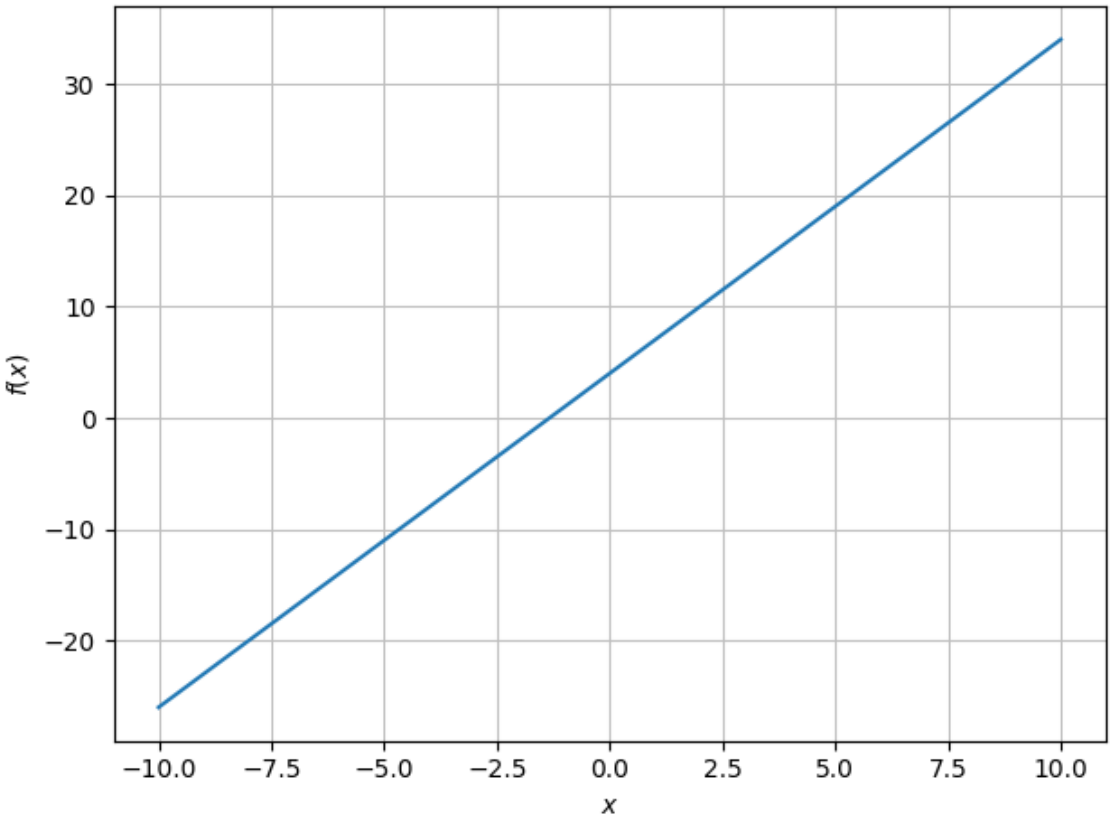
Bemærk at størrelsen også er angivet i første figur (enhed er tommer).

```
plot_list([1,2,3],[6,5,6],is_point=True,size=[4,4])
```

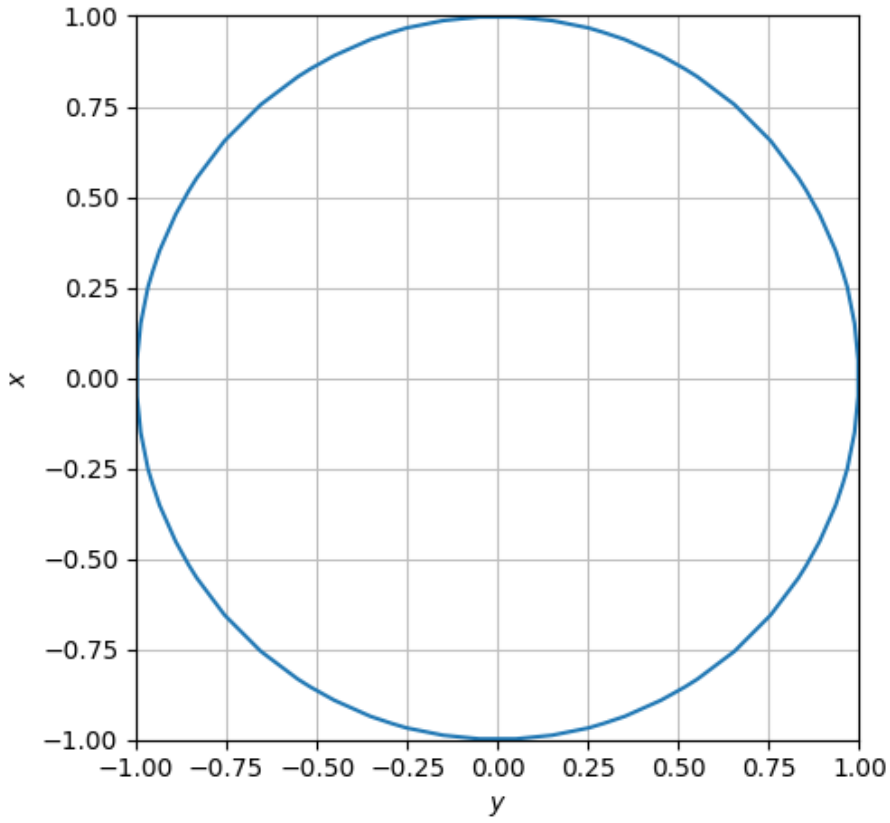


Figur 1: Punkter med plot_list.

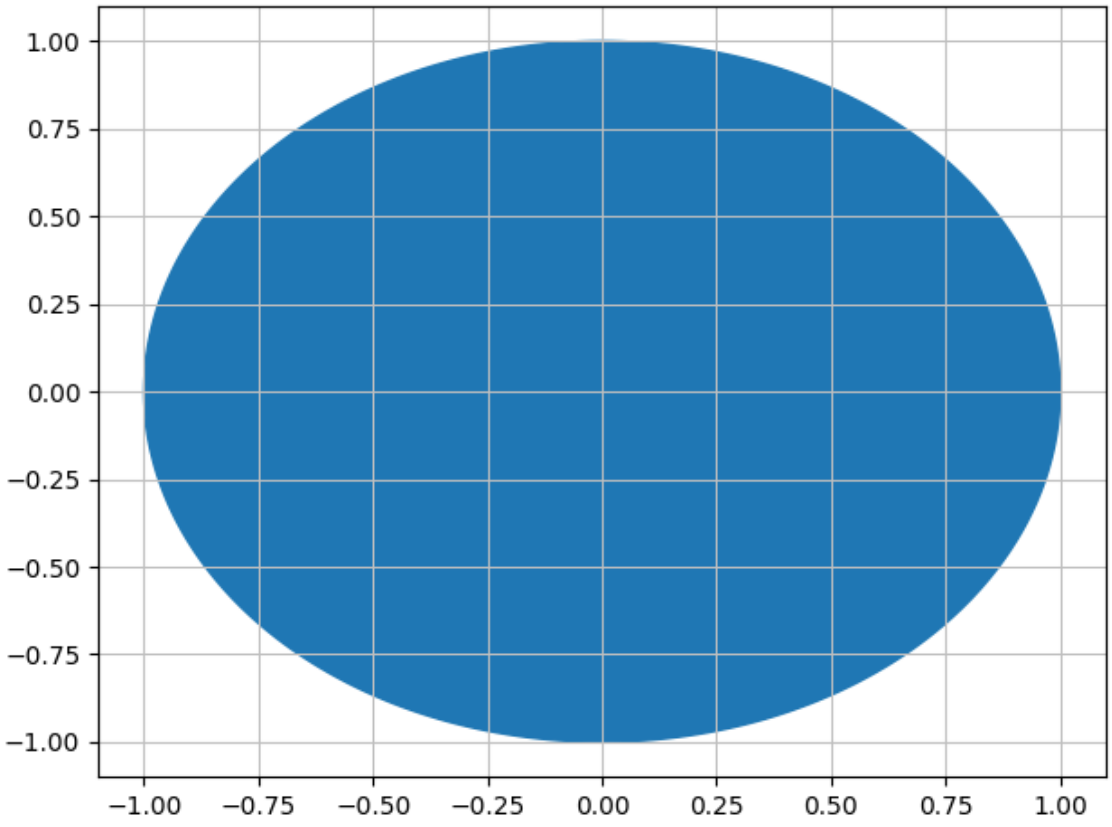
```
plot(3*x+4)
```



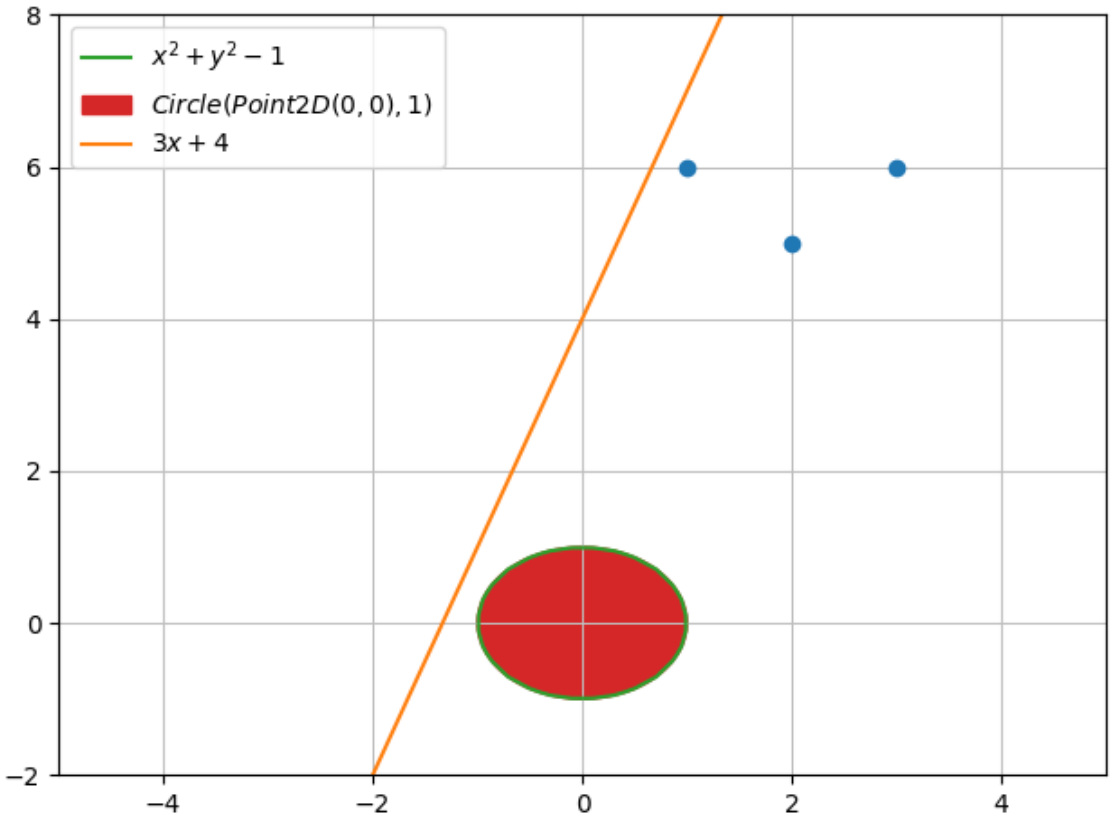
```
plot_implicit(x**2 + y**2 - 1,xlim=(-1,1),ylim=(-1,1), n=200, aspect="equal")
```



```
from sympy import Circle, Point
plot_geometry(Circle(Point(0, 0), 1))
```



```
p1 = plot_list([1,2,3],[6,5,6],xlim=(-5,5),ylim=(-2,8),is_point=True,show=False)
p2 = plot(3*x+4,show=False)
p3 = plot_implicit(x**2 + y**2 - 1,xlim=(-1,1),ylim=(-1,1),show=False)
p4 = plot_geometry(Circle(Point(0, 0), 1),show=False)
p = p1 + p2 + p3 + p4
p.show()
```



B5. Vektorer

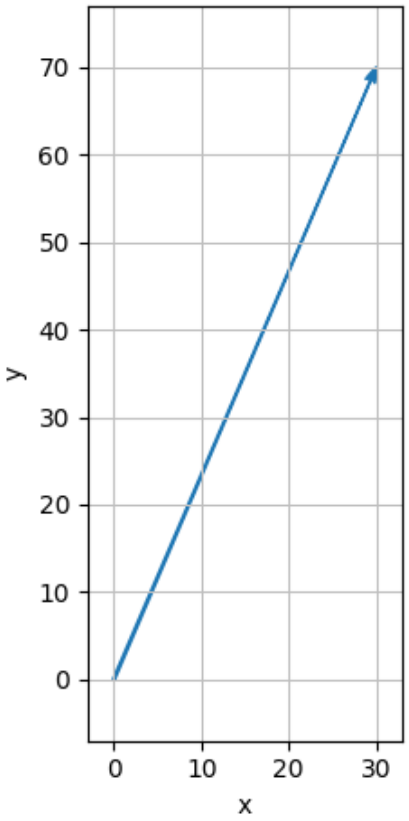
```
a = vector(30,70)
a
```

$$\begin{bmatrix} 30 \\ 70 \end{bmatrix}$$

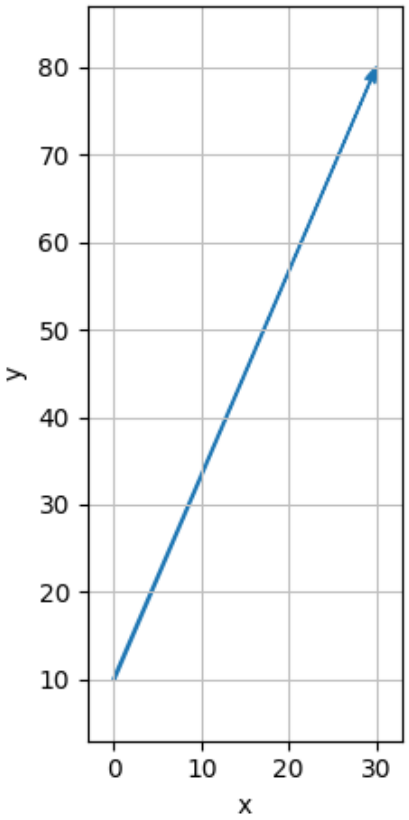
```
a.dot(vector(6,0))
```

$$180$$

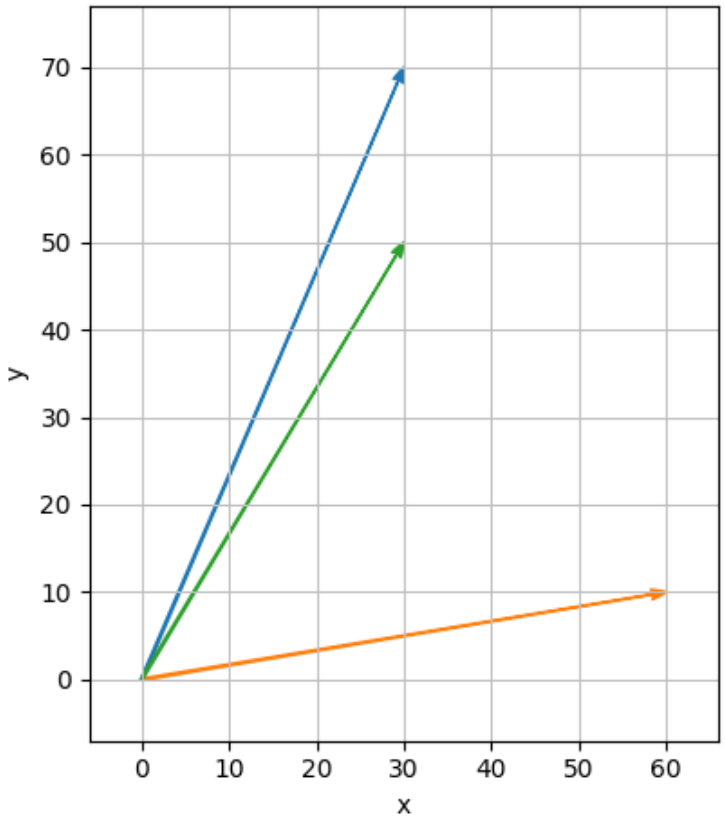
```
plot_vector(a)
```



```
plot_vector((0,10),a)
```



```
plot_vector([a,(60,10),(30,50)])
```



B6. Deskriptiv Statistik

Ugrupperet

```
max([1,1,1,3,7,8])
```

```
8
```

```
min([1,1,1,3,7,8])
```

```
1
```

```
mean([1,1,1,3,7,8])
```

```
3.5
```

```
median([1,1,1,3,7,8])
```

```
2.0
```

```
kvartiler([1,1,1,3,7,8])
```

```
[1, 1.0, 2.0, 7.0, 8]
```

```
percentile([1,1,1,3,7,8],[20,50,80])
```

```
array([1, 1, 7])
```

```
frekvenstabel([1,1,1,3,7,8])
```

Observation	Hyppighed	Frekvens %	Kumuleret frekvens %
1	3	50.0	50.0
3	1	16.667	66.667
7	1	16.667	83.333
8	1	16.667	100.0

```
<gym_cas.stats.FrequencyTable at 0x2106203f620>
```

```
var([1,1,1,3,7,8],ddof=1)
```

10.3

```
var([1,1,1,3,7,8])
```

8.583333333333334

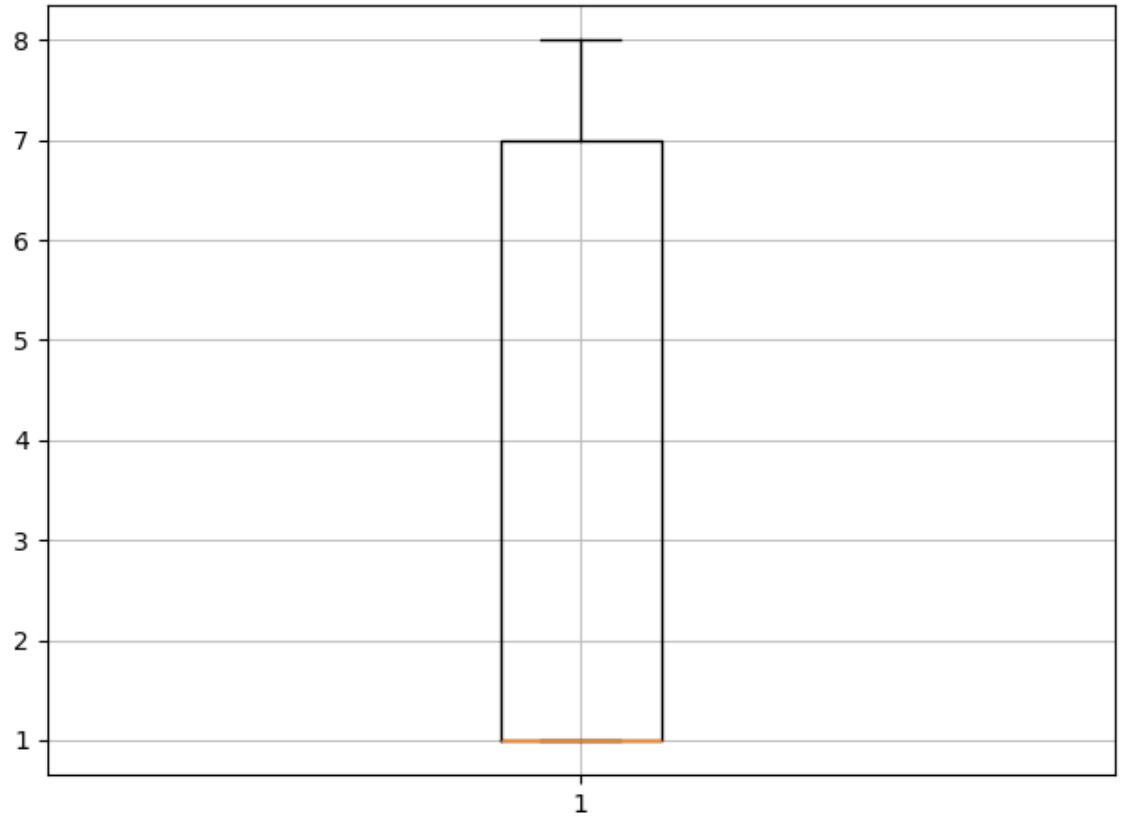
```
std([1,1,1,3,7,8],ddof=1)
```

3.2093613071762426

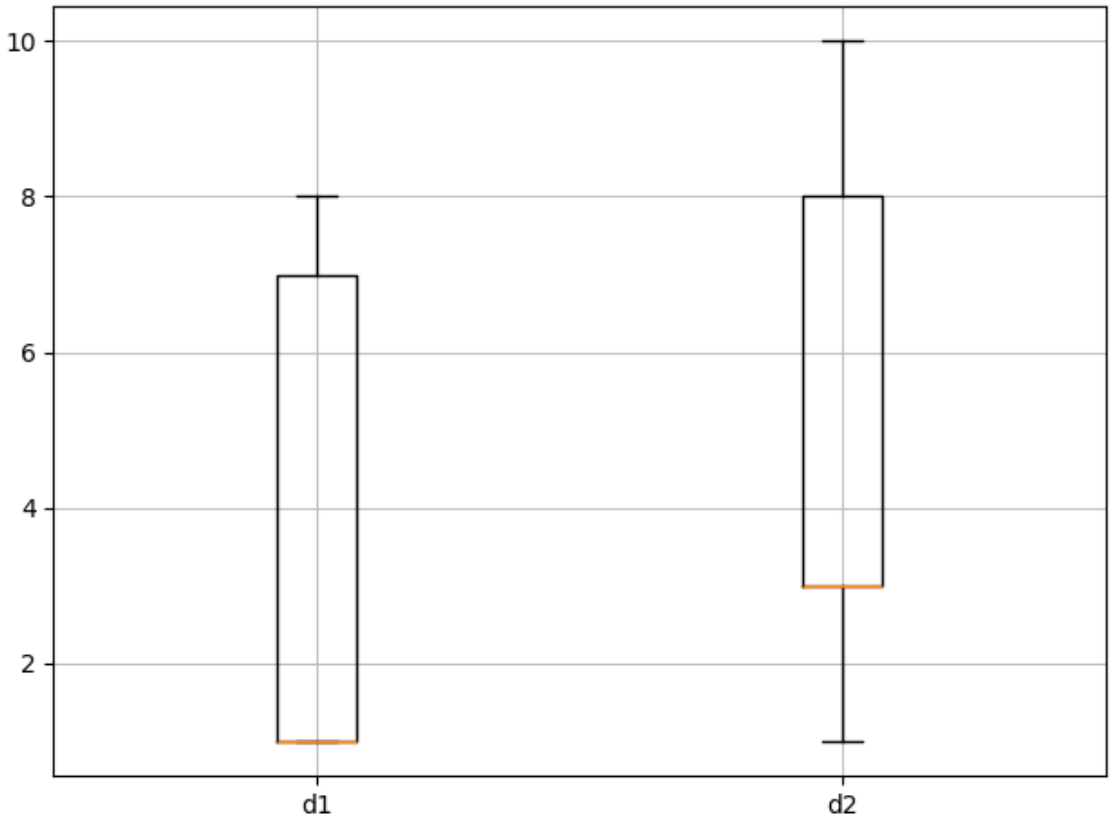
```
std([1,1,1,3,7,8])
```

2.9297326385411577

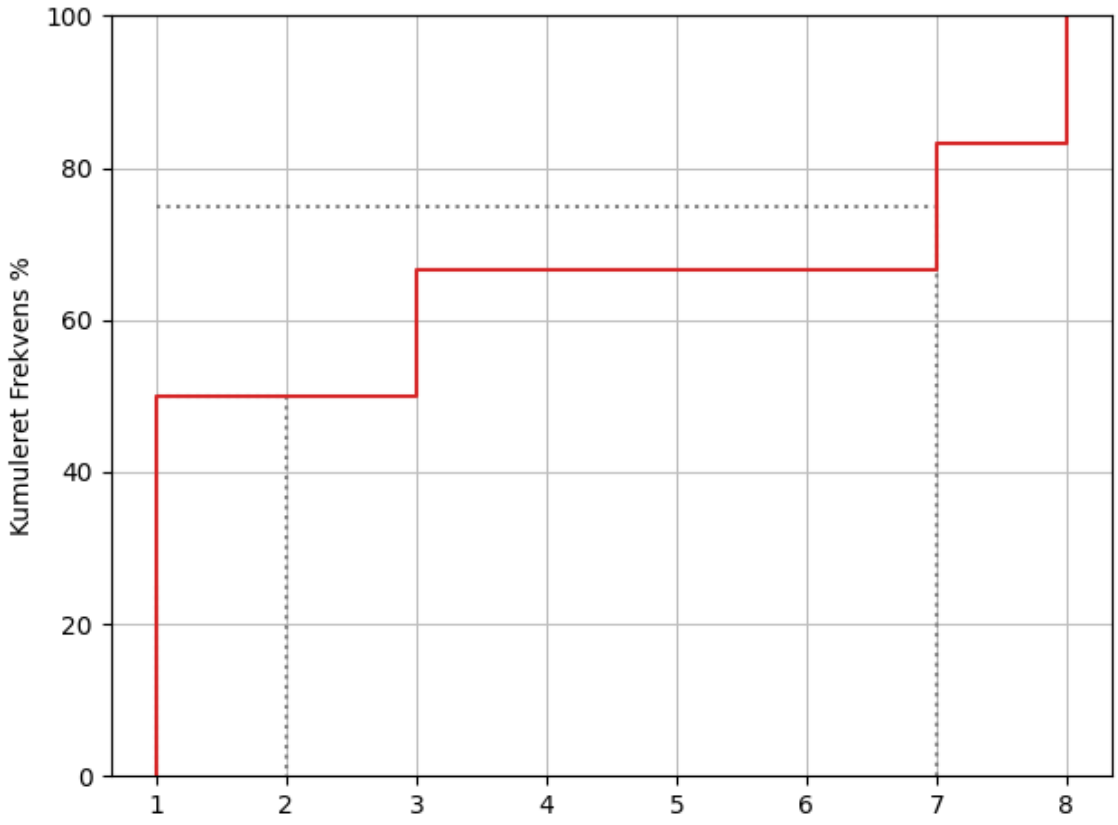
```
boxplot([1,1,1,3,7,8])
```



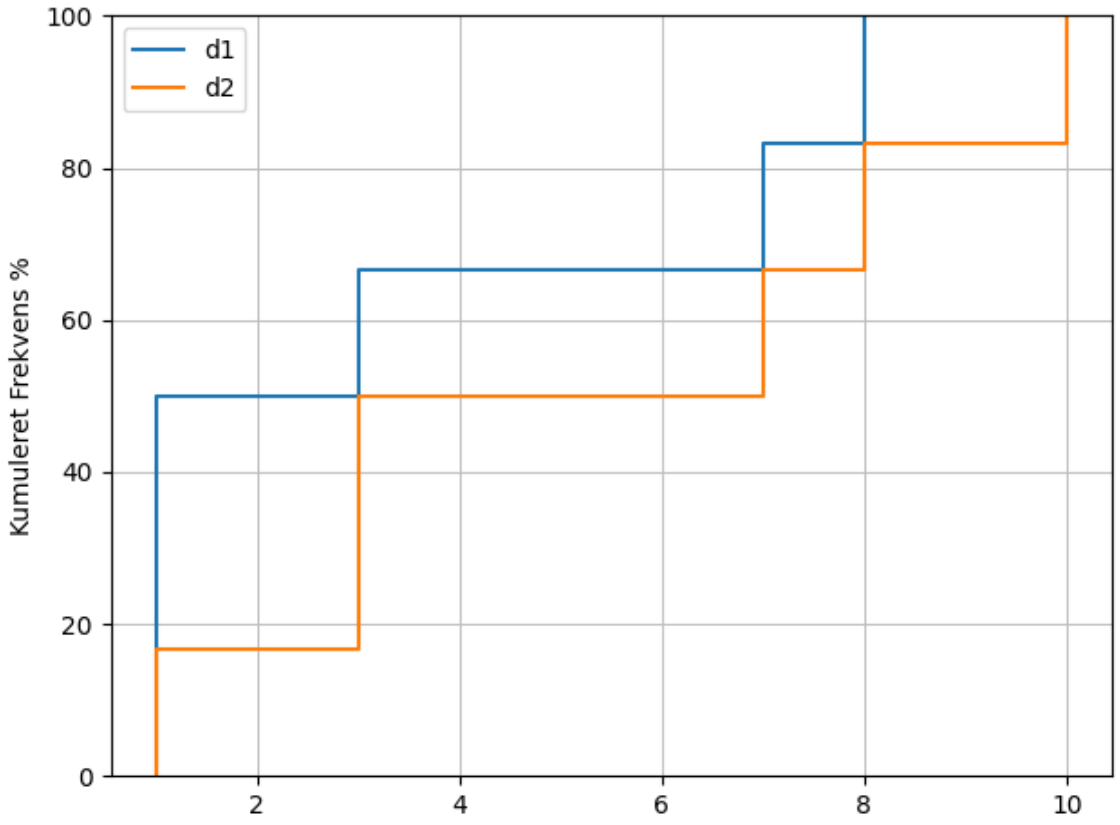
```
boxplot([[1,1,1,3,7,8],[1,3,3,7,8,10]],label=["d1","d2"])
```



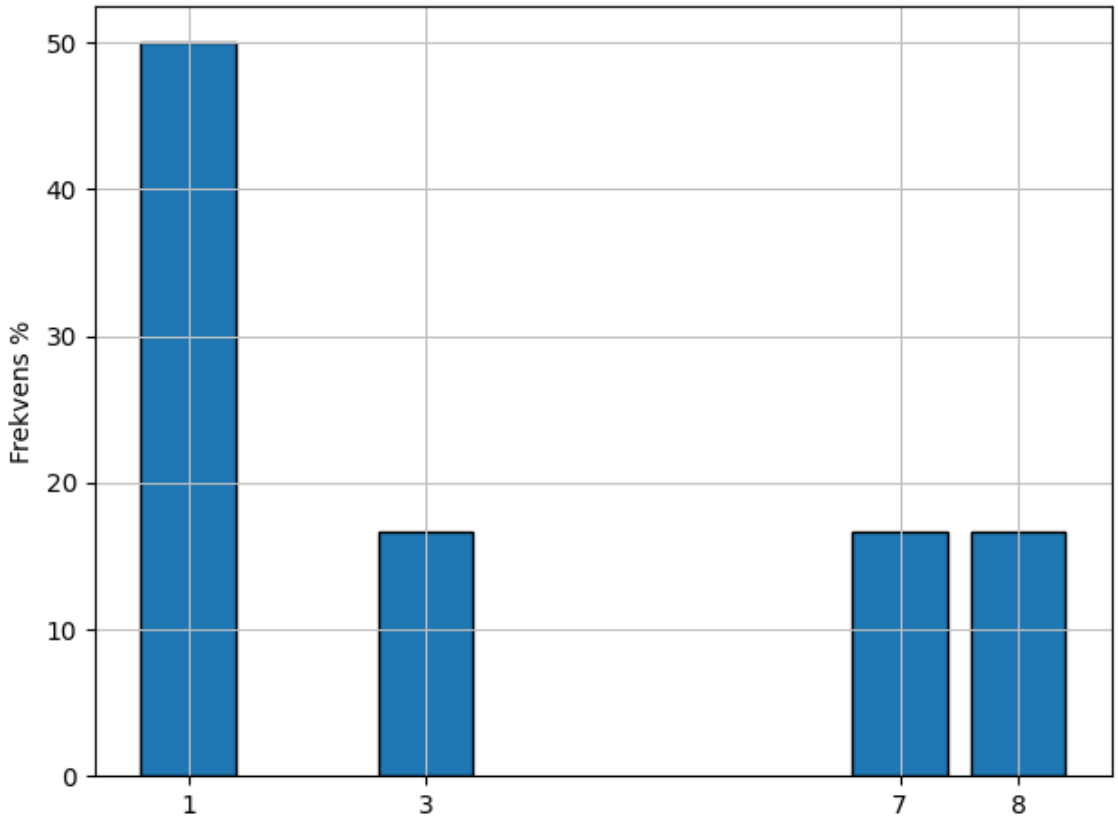
```
plot_sum([1,1,1,3,7,8])
```



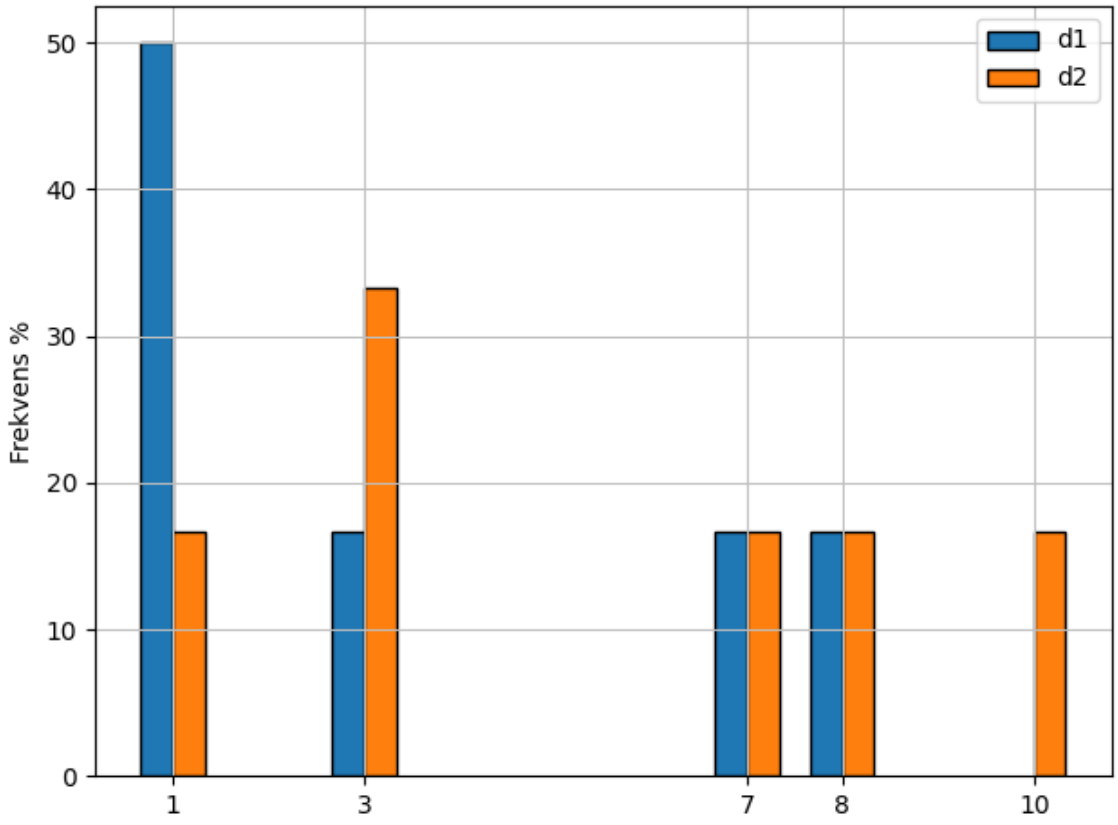
```
plot_sum([[1,1,1,3,7,8],[1,3,3,7,8,10]],label=["d1","d2"])
```



```
plot_bars([1,1,1,3,7,8])
```



```
plot_bars([[1,1,1,3,7,8],[1,3,3,7,8,10]],label=["d1","d2"])
```



Grupperet

```
group_mean([1,2,3,1],[1,2,3,4,5])
```

```
3.0714285714285716
```

```
group_percentile([1,2,3,1],[1,2,3,4,5],[0,25,50,75,100])
```

```
[1.0, 2.375, 3.166666666666667, 3.75, 5.0]
```

```
group_var([1,2,3,1],[1,2,3,4,5],ddof=1)
```

```
0.9523809523809524
```

```
group_std([1,2,3,1],[1,2,3,4,5],ddof=1)
```

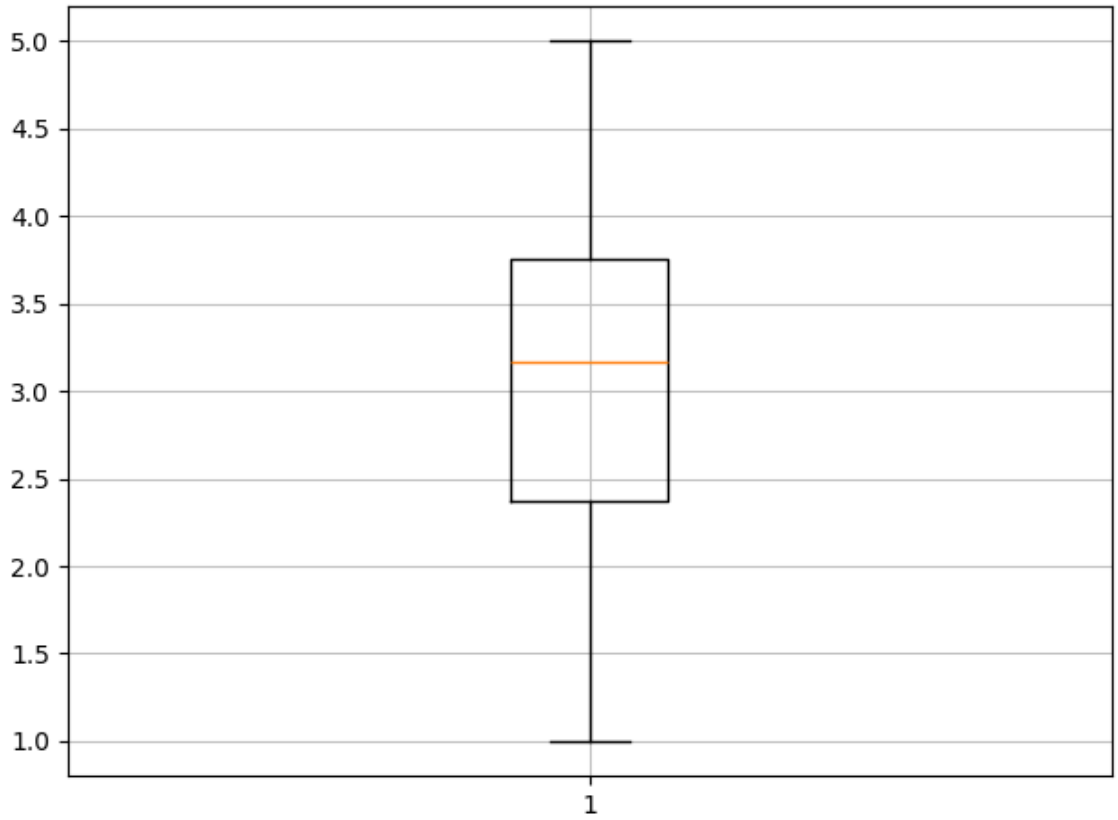
```
0.975900072948533
```

```
frekvenstabel([1,2,3,1],[1,2,3,4,5])
```

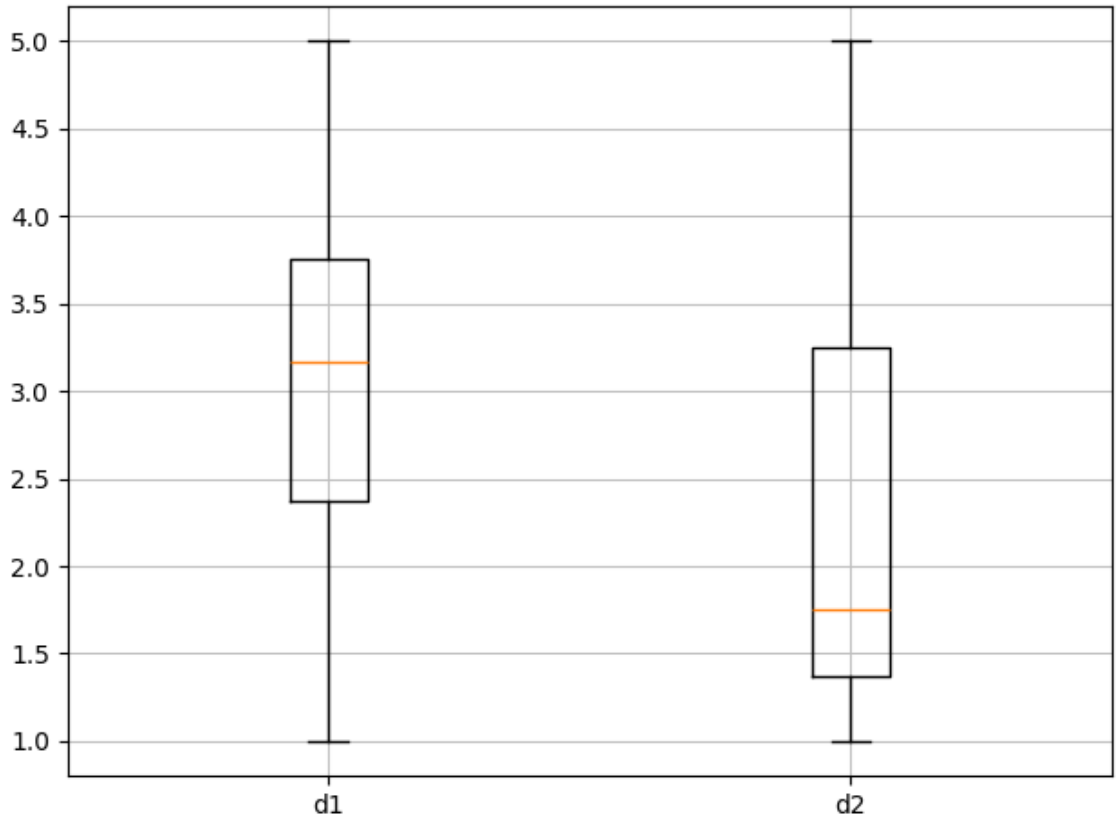
Observationsinterval	Hyppighed	Frekvens %	Kumuleret frekvens %
] 1 ; 2]	1	14.286	14.286
] 2 ; 3]	2	28.571	42.857
] 3 ; 4]	3	42.857	85.714
] 4 ; 5]	1	14.286	100.0

```
<gym_cas.stats.FrequencyTableGrouped at 0x2106e5a39e0>
```

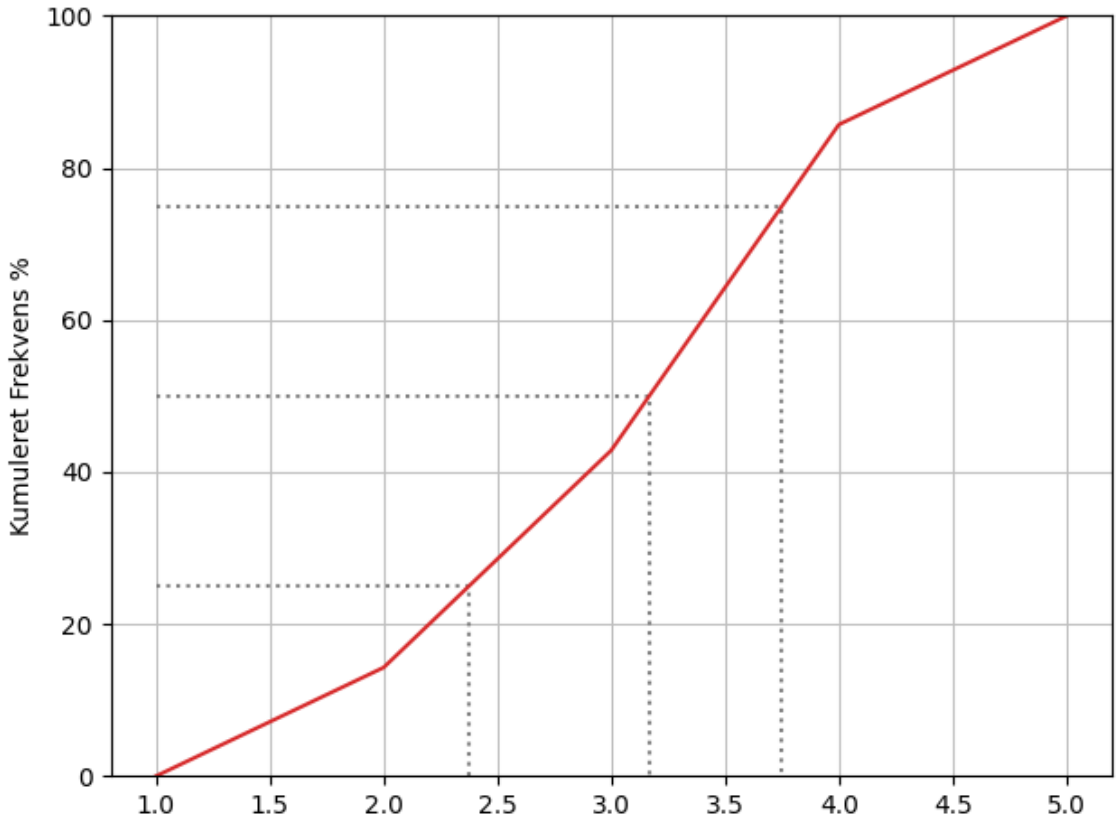
```
boxplot([1,2,3,1],[1,2,3,4,5])
```



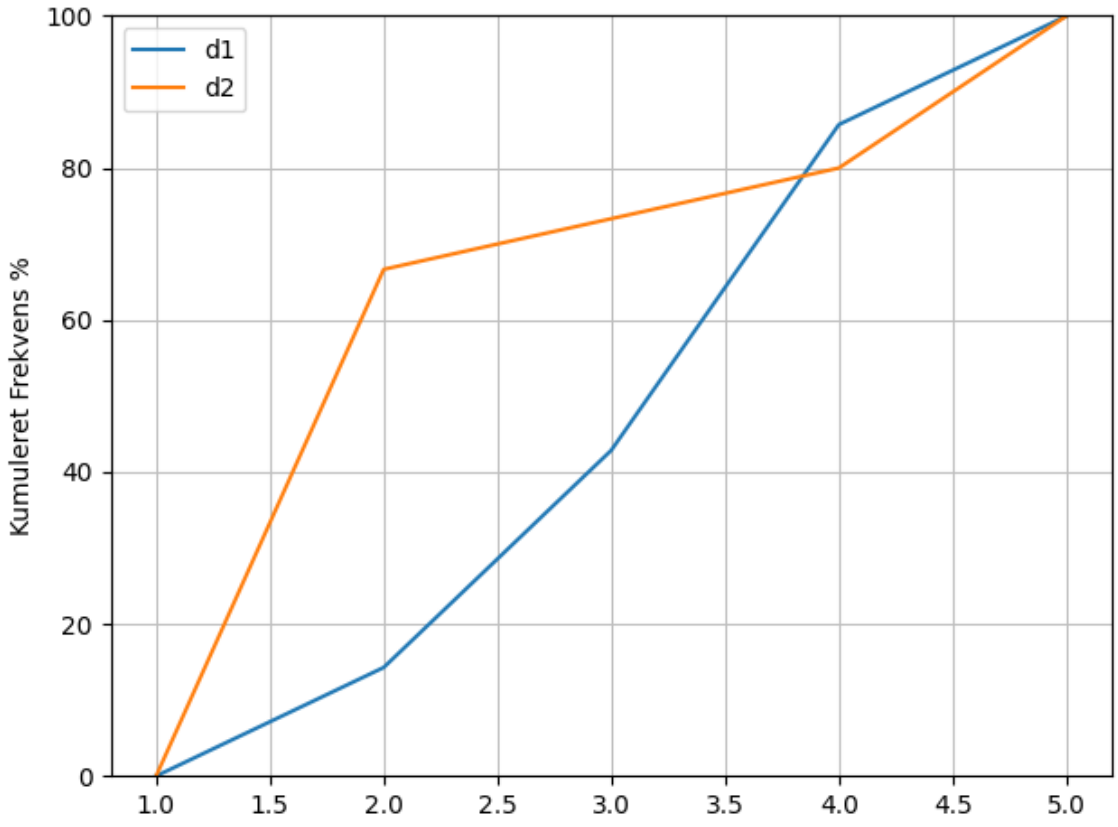
```
boxplot([[1,2,3,1],[10,1,1,3]],[1,2,3,4,5],["d1","d2"])
```



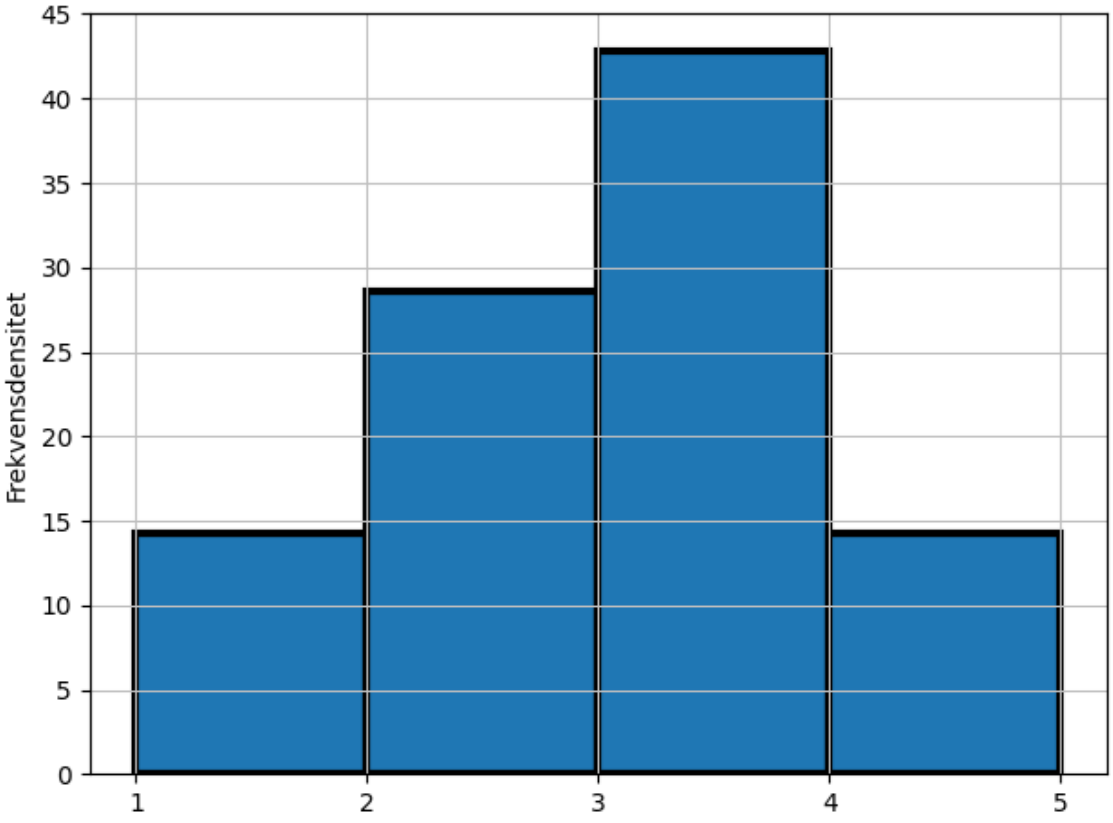
```
plot_sum([1,2,3,1],[1,2,3,4,5])
```



```
plot_sum([[1,2,3,1],[10,1,1,3]],[1,2,3,4,5],["d1","d2"])
```



```
plot_hist([1,2,3,1],[1,2,3,4,5])
```



B8. Funktioner

```
def f(x):
    return 3*x**2 + 2*x + 10
f(x)
```

$3x^2 + 2x + 10$

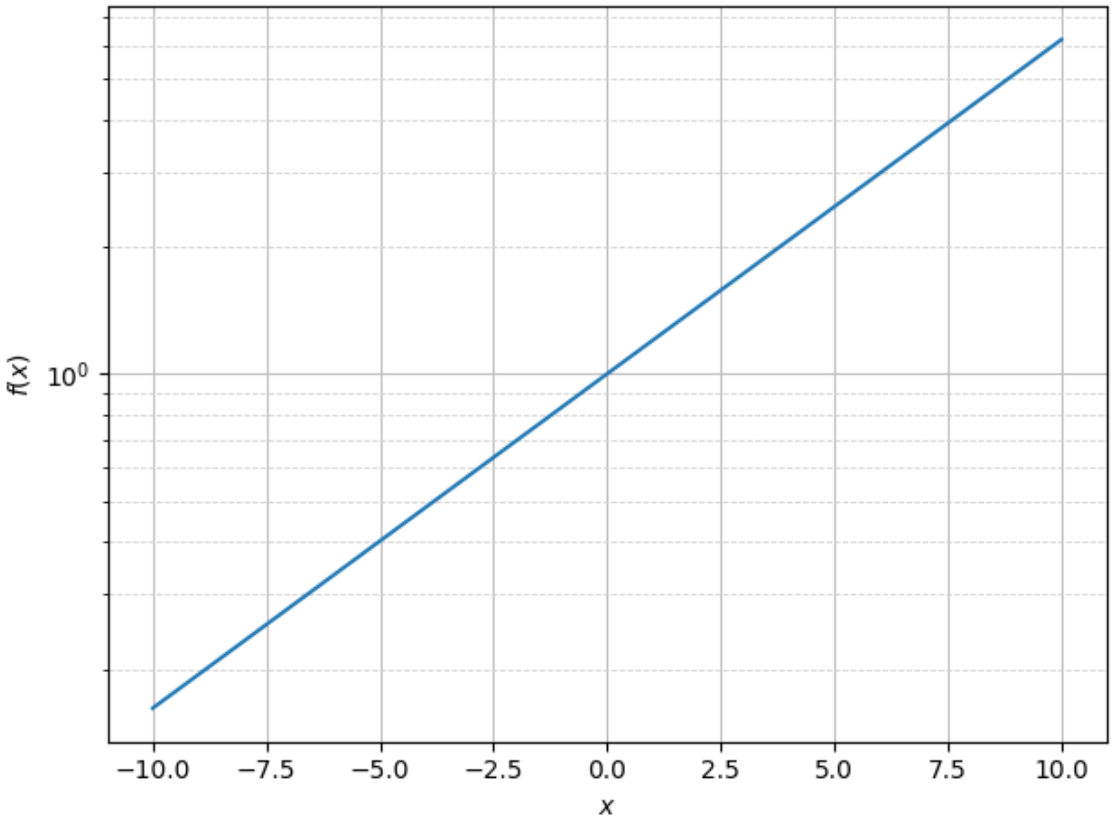
```
f(3)
```

43

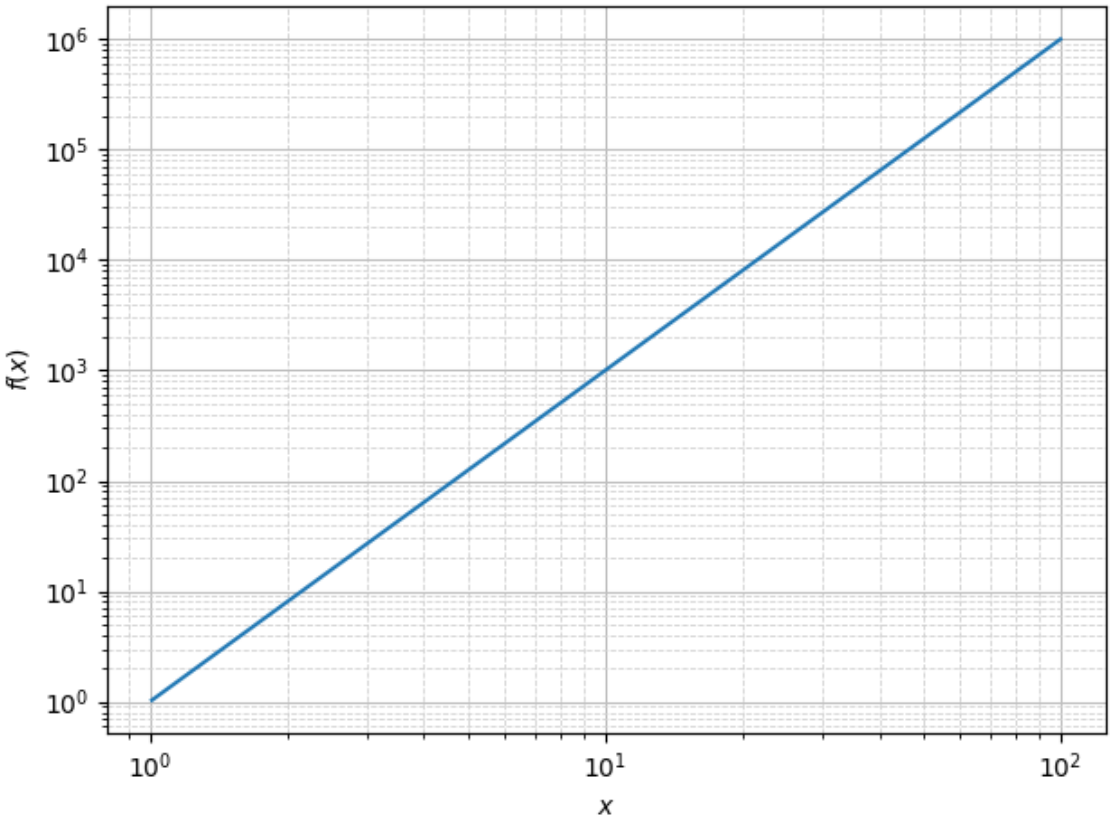
```
def f(x):
    return Piecewise((0, x < -1), (x**2, x <= 1), (5, True))
f(x)
```

$$\begin{cases} 0 & \text{for } x < -1 \\ x^2 & \text{for } x \leq 1 \\ 5 & \text{otherwise} \end{cases}$$

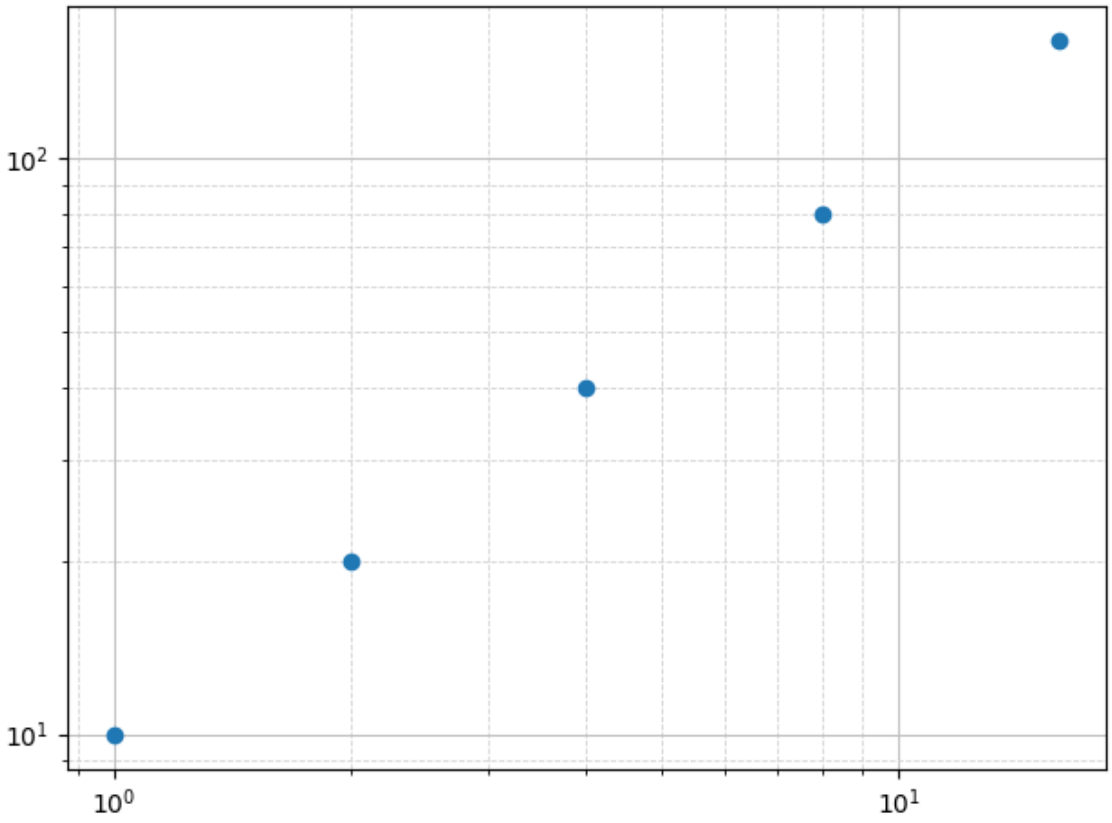
```
plot(1.2**x,yscale="log")
```



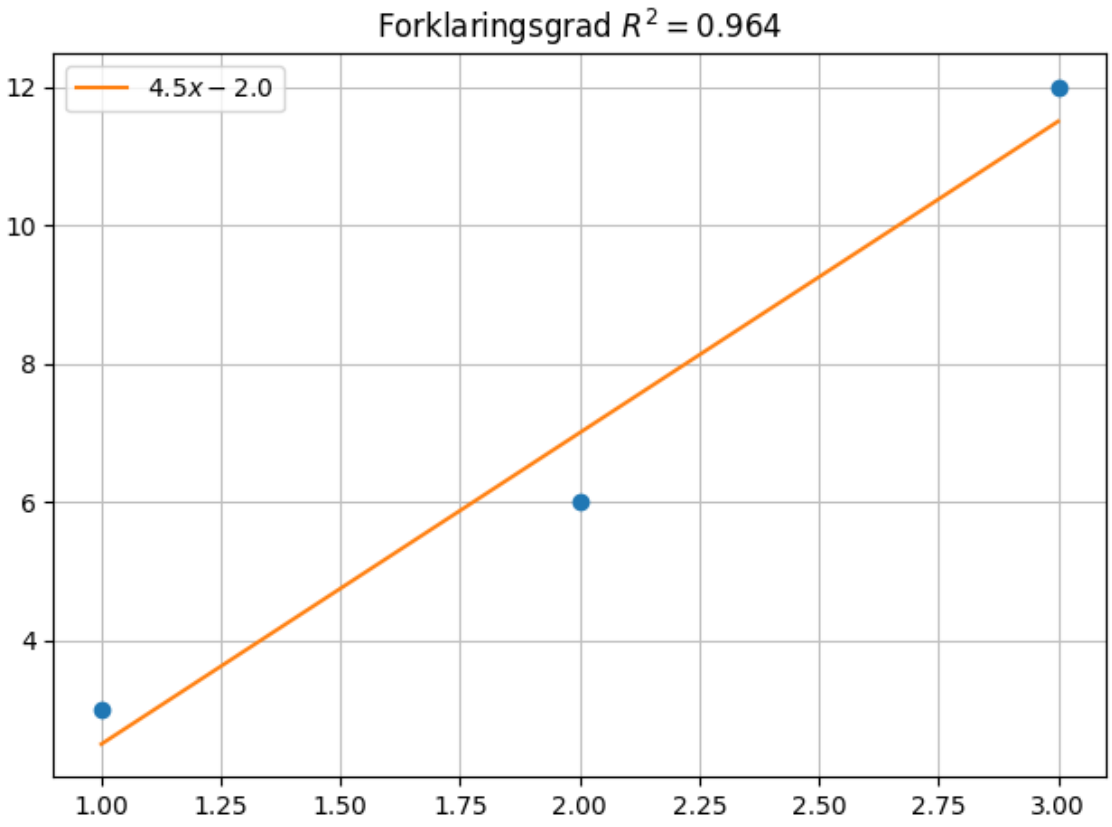
```
plot(x**3,(x,1.01,100),xscale="log", yscale="log")
```



```
plot_list([1,2,4,8,16],[10,20,40,80,160],is_point=True,xscale="log",yscale="log")
```

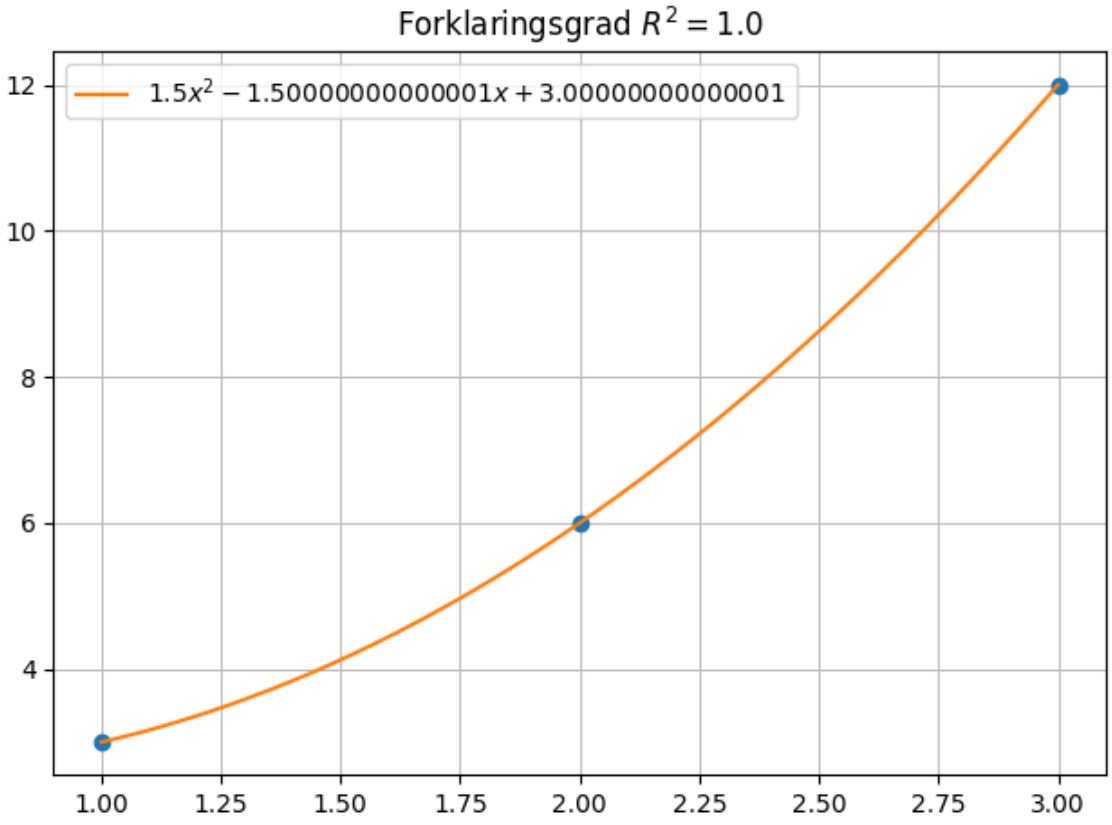


```
regression_poly([1,2,3],[3,6,12],1)
```



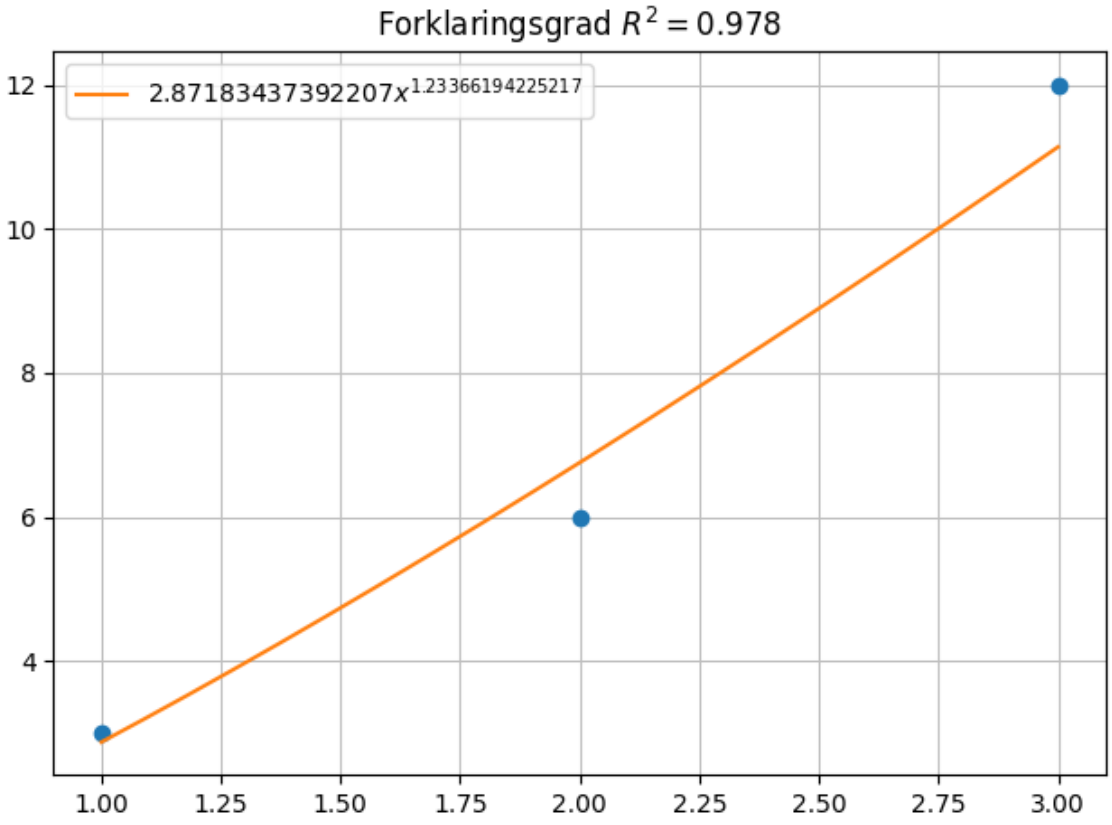
$(t \mapsto 4.5t - 2.0)$

```
regression_poly([1,2,3],[3,6,12],2)
```



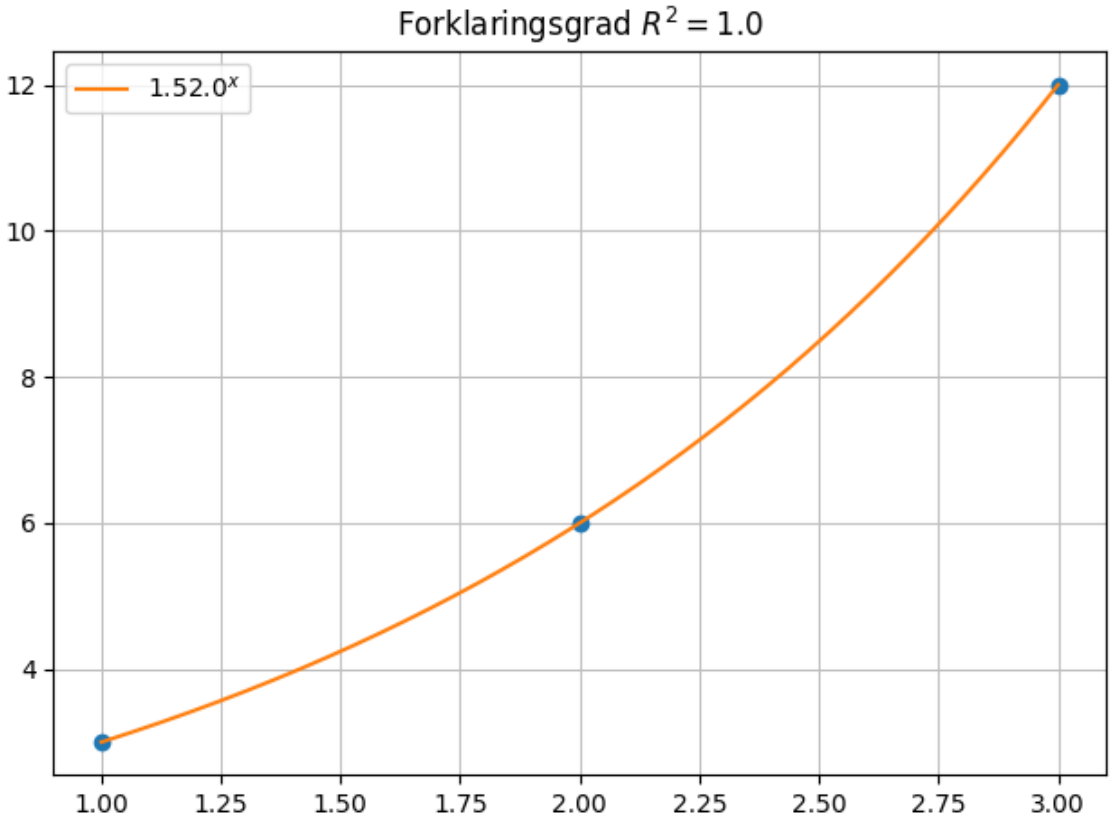
$(t \mapsto 1.5t^2 - 1.50000000000001t + 3.00000000000001)$

```
regression_power([1,2,3],[3,6,12])
```



$(t \mapsto 2.87183437392207t^{1.23366194225217})$

```
regression_exp([1,2,3],[3,6,12])
```



$\left(t \mapsto 1.52.0^t\right)$

B9. Differentialregning

```
limit(1/x,x,0,"+")
```

∞

```
diff(3*x**2)
```

$6x$

```
def df(xin):
    return diff(x**2).subs(x,xin)
df(3)
```

6

B10. Integralregning

```
integrate( 6*x )
```

$3x^2$

```
integrate( 6*x , (x,0,1))
```

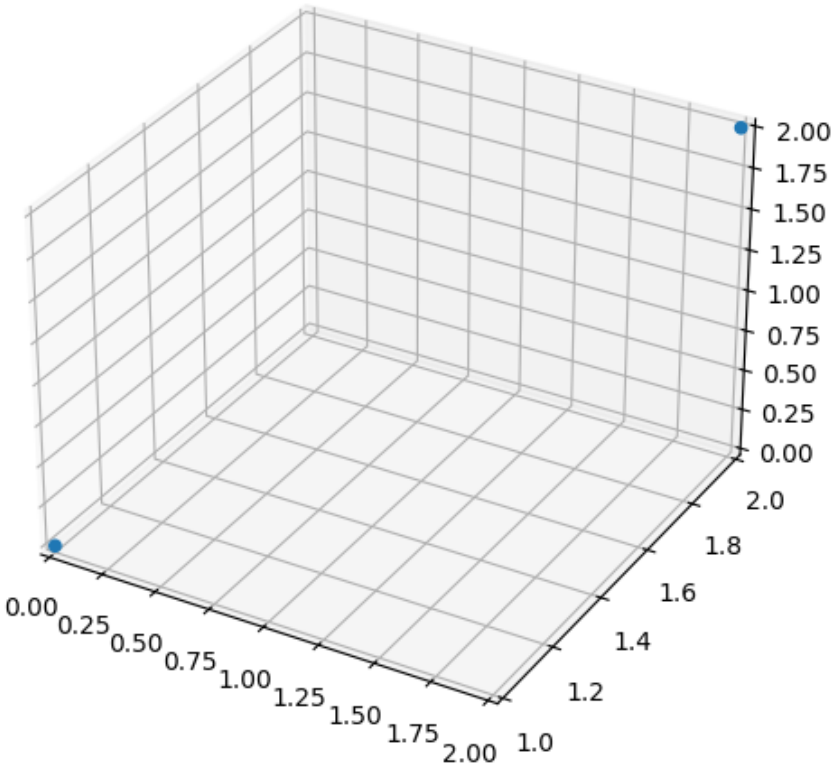
3

A1. Vektorer i rummet

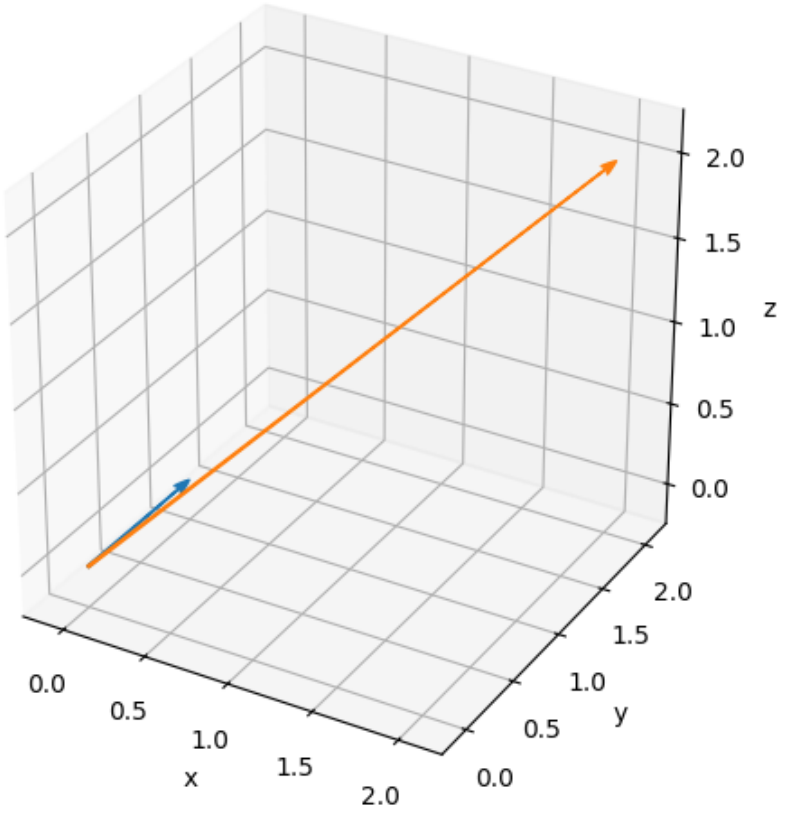
```
a = vector(0,1,0)
b = vector(2,2,2)
a.cross(b)
```

$$\begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix}$$

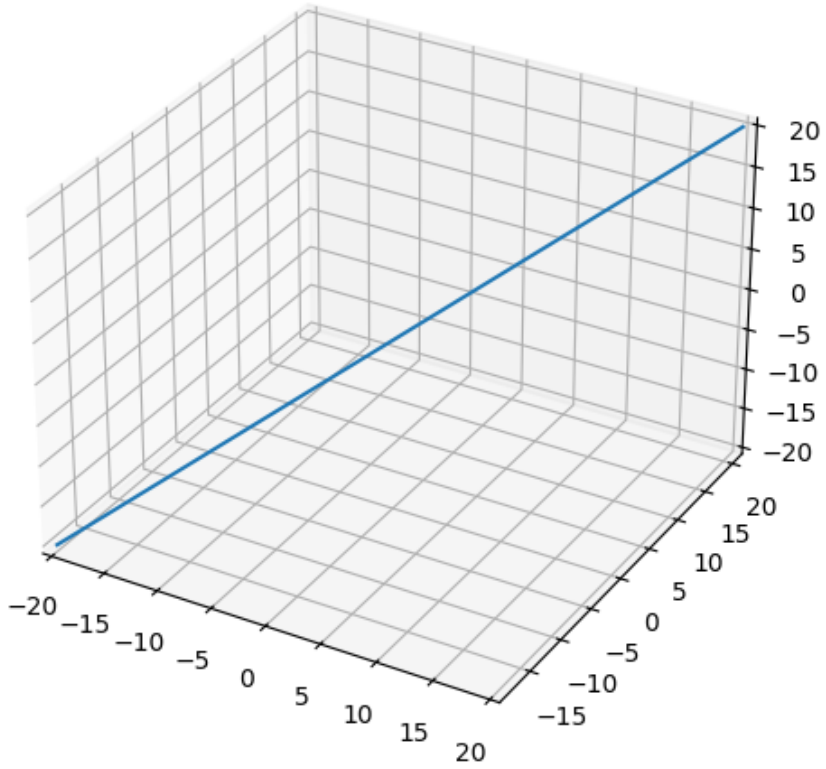
```
plot3d_list([0,2],[1,2],[0,2],is_point=True)
```



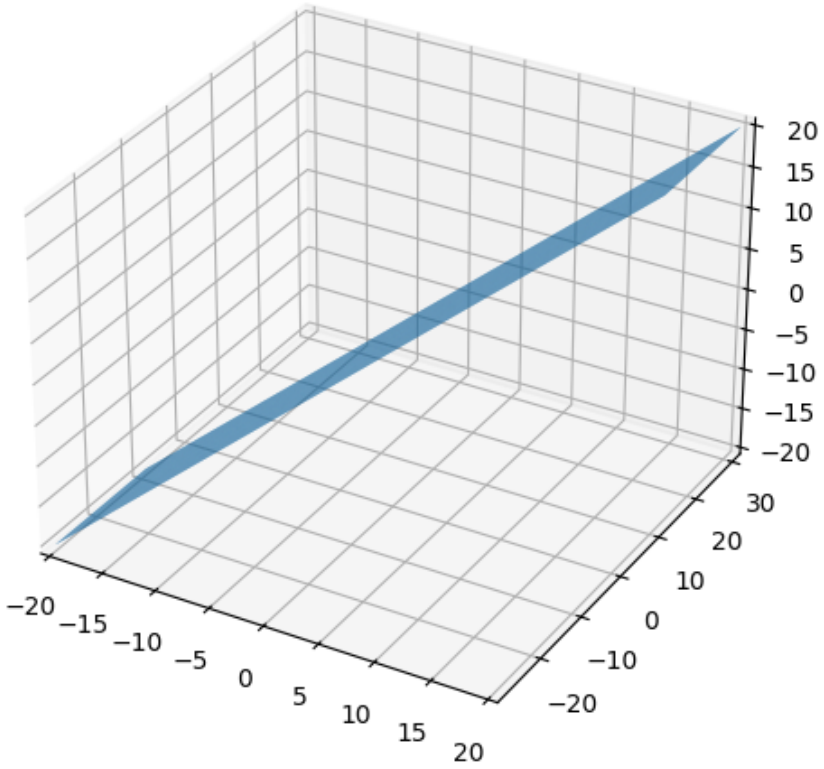
```
plot_vector([a,b])
```



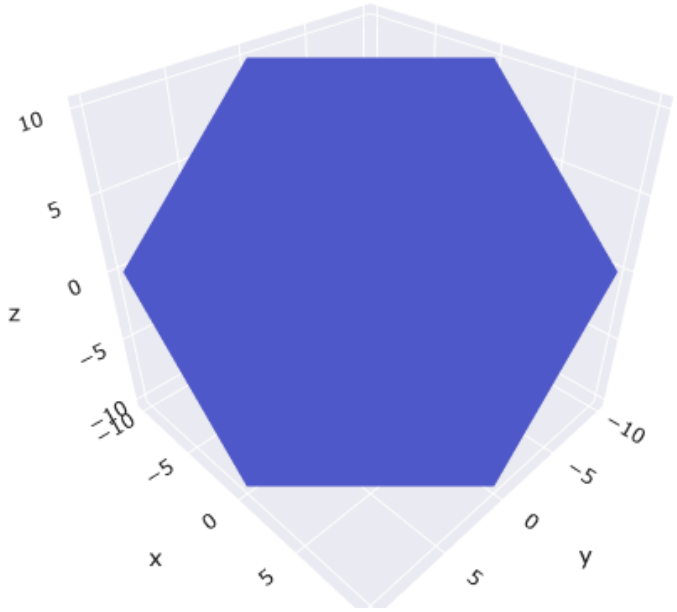
```
plot3d_line(a+t*b)
```



```
plot3d_plane(a+b*t+vector(0,1,0)*s)
```



```
# Kræver Plotly eller K3D. Output skal gemmes som billede og indsættes manuelt
from spb import PB
plot3d_implicit(x+y+z,backend=PB)
```



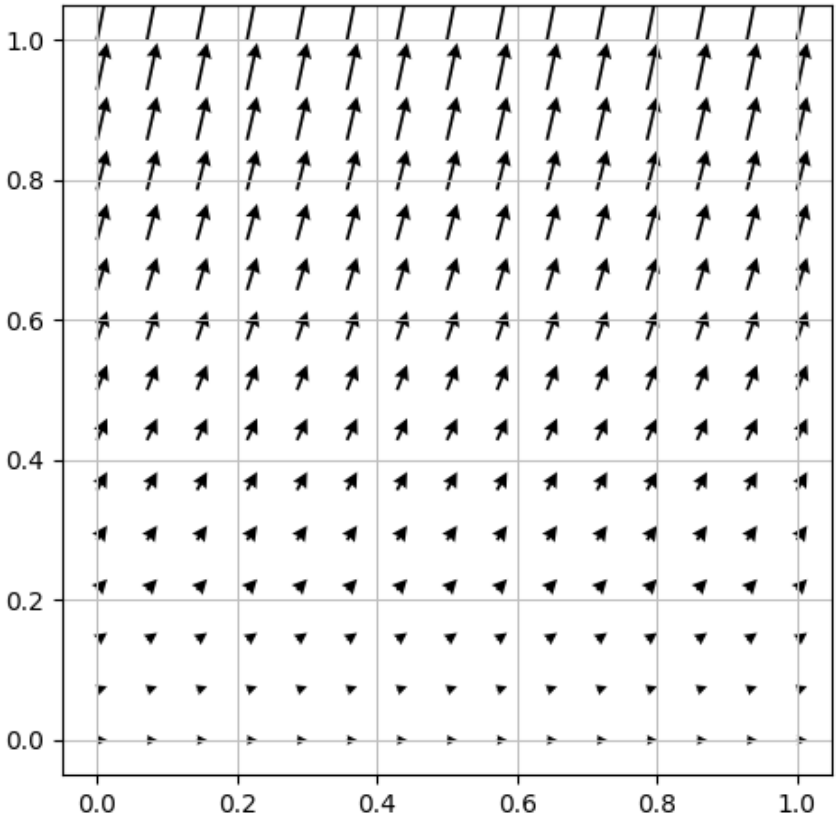
Figur 2: plot3d_implicit

A4. Differentialligninger

```
f = Function('f')
dsolve(diff(f(x),x)-5*f(x))
```

$f(x) = C_1 e^{5x}$

```
plot_ode(diff(f(x),x)-5*f(x), (x,0,1), (f,0,1))
```



A5. Diskret Matematik

```
X = [2*x for x in range(1,10)]
X
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18]
```

```
X = [1]
for i in range(1,10):
    X.append(X[i-1]+i)
X
```

```
[1, 2, 4, 7, 11, 16, 22, 29, 37, 46]
```