s11

March 29, 2022

1 VV4

1.1 B

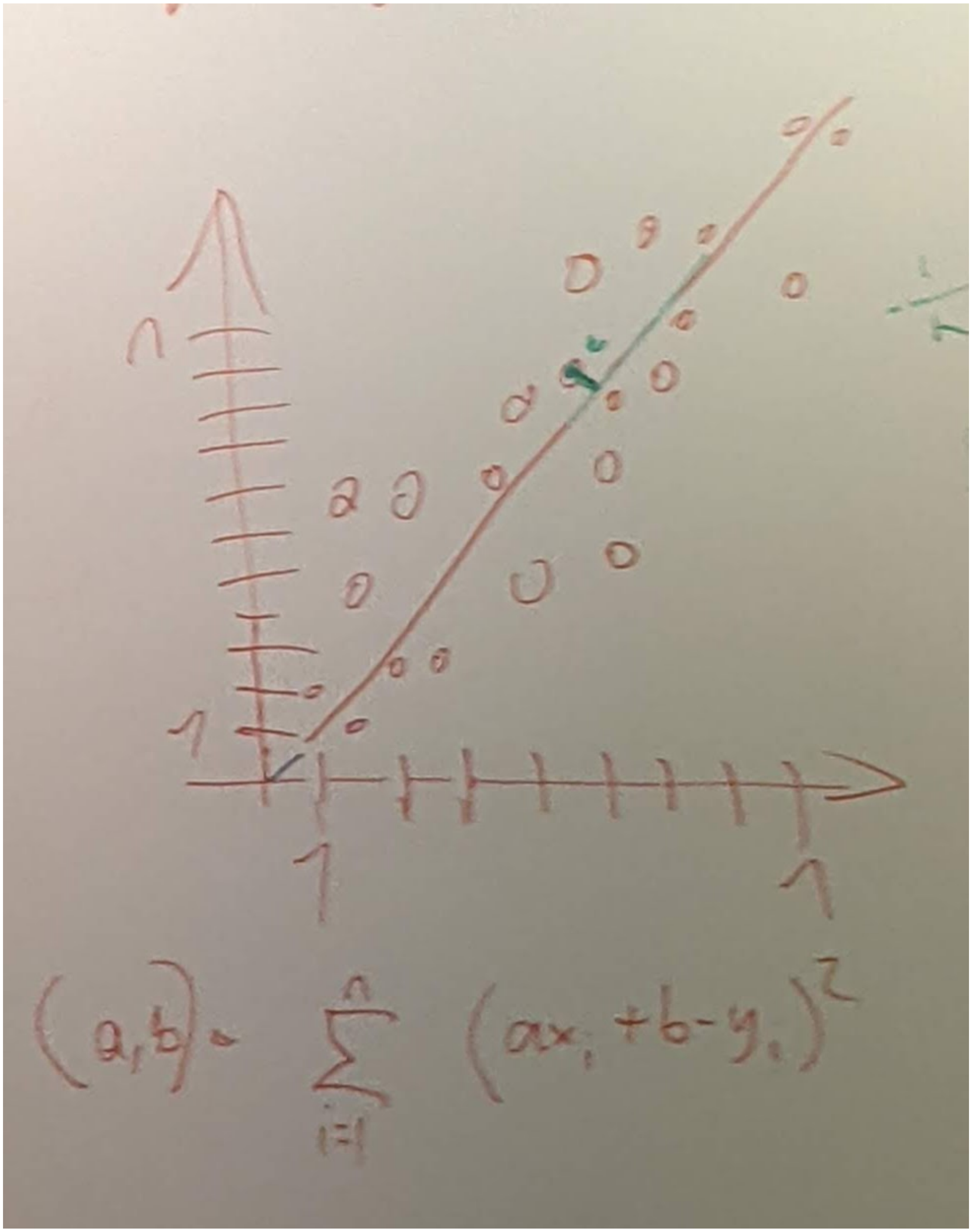
1.1.1 a)

gagnasafn: $(x_1, y_1), ..., (x_n, y_n)$
formúla: $y = ax + b$
$S(a, b) = \sum_{i=1}^{n}(ax_i + b - y_i)^2$

[ ]:
```
# bara til þess að setja inn mynd
# ehv fucky með export í markdown
from IPython.display import Image
Image(filename='./imgs/bb.jpg')
```

[ ]:

jafna bestu parabólu gagnasafnsins:
$S(a,b,c) = \sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)^2$



1.1.2 b)

jafna bestu parabólu gagnasafnsins:
$S(a,b,c) = \sum_{i=1}^{n}(ax_i^2 + bx_i + c - y_i)^2$

1.2 C

1.2.1 a)

BYRJUM ÞETTA
$D_x(x^2+y)^6 + e^{y-x}$
$\frac{\delta}{\delta x}(e^{-x+y}) + \frac{\delta}{\delta x}(x^2+y)^6$

KEÐJUREGLAN
$\frac{\delta}{\delta x}(x^2+y)^6 + (e^{-x+y}(\frac{\delta}{\delta x}(-x+y))$

SKIPTA SAMLAGANINGU NÐUR Í BÚTA
$\frac{\delta}{\delta x}(x^2+y)^6 + (-(\frac{\delta}{\delta x}(x)) + \frac{\delta}{\delta x}(y))$
$\frac{\delta}{\delta x}(x^2+y)^6 + (-(1) + \frac{\delta}{\delta x}(y))$
$\frac{\delta}{\delta x}(x^2+y)^6 + (-(1) + 0)$
$-e^{-x+y} + \frac{\delta}{\delta x}(x^2+y)^6$

OKOK KEÐJUREGLA AFTUR
$-e^{-x+y} + (6(x^2+y)^5(\frac{\delta}{\delta x}(x^2+y))$

SKIPTA NIÐUR
$-e^{-x+y} + (6(x^2+y)^5(\frac{\delta}{\delta x}(x^2) + \frac{\delta}{\delta x}(y))$
$-e^{-x+y} + (6(x^2+y)^5(2x + \frac{\delta}{\delta x}(y))$
$-e^{-x+y} + (6(x^2+y)^5(2x+0)$

EINFALDA
$-e^{y-x} + 12(x^2+y)^5$

LOKASVAR: $-e^{y-x} + 12(x^2+y)^5$

1.2.2 b)

finnum fyrst f_x og f_y
$f_x : 3x^2 + y$
$f_y : -3y^2 + x$

þá er ég með $\nabla f(x,y) = (\binom{3x^2+y}{-3y^2+x})$

1.2.3 c)

þar sem $f = x^3 + xy - y^3$ er auðvelt að finna að $f(1,2) = 1+2+8 = 11$

nú vitum við líka $\nabla f(x,y) = (\binom{3x^2+y}{-3y^2+x})$ þannig $\nabla f(1,2) = (\binom{5}{11})$

setjum inn í margvíðu Taylor-setninguna: $f(1,2) + \nabla f(1,2) * (x - (1,2)) = 11 + \binom{5}{11} * (x_1 - 1, x_2 - 2) = 5x_1 + 11x_2$

1.3 D

1.3.1 a)

$a = (1,2,3)$, $b = (4,2,0)$, $c = (1,0,1)$

$c_1 \begin{pmatrix}1\\2\\3\end{pmatrix} + c_2 \begin{pmatrix}4\\2\\0\end{pmatrix} + c_3 \begin{pmatrix}1\\0\\1\end{pmatrix}$

1. $c_1 + 4c_2 + c_3 = 0$
2. $2c_1 + 2c_2 = 0$
3.

1.4 $3c_1 + c_3 = 0$

4. hér eru c_1 og c_3 andstæður og c_2 hlýtur þá að vera 0
5. þar sem c_2 er 0 þá getur c_1 ekki verið neitt nema 0
6. c_1 er 0 þannig c_3 verður að vera 0

öll c-in eru 0 þannig vigrarnir eru línulega óháðir

1.4.1 b)

[ ]:
```python
import numpy as np
import numpy.linalg as la
from math import acos, degrees

a = np.array([1,2,3])
b = np.array([4,2,0])
theta = degrees(acos(a@b/(la.norm(a)*la.norm(b))))
print(theta)
```

```
61.43917478218875
```

hornið á milli vigranna a og b er 61.44 gráður

1.4.2 c)

$a = (1, 2, 3)$, $b = (4, 2, 0)$, $d = (x_1, x_2, x_3)$

$c_1 \begin{pmatrix}1\\2\\3\end{pmatrix} + c_2 \begin{pmatrix}4\\2\\0\end{pmatrix} + c_3 \begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}$

1. $c_1 + 4c_2 + x_1c_3 = 0$
2. $2c_1 + 2c_2 + x_2c_3 = 0$
3.

1.5 $3c_1 + x_3c_3 = 0$

4. byrjum á að láta x_1 vera -4, þá vitum við að c_2 og c_3 eru andstæður og $c_1 = 0$
5. nú þar sem $c_1 = 0$ þá þarft x_2 að vera andstæðan við 2, þ.e. -2

6. núna þarf $0 + 1 * x_3 = 0$ og þá getur x_3 bara verið 0

þá endum við með vigurinn $d = (-4, -2, 0)$

1.5.1 d)

ef hlutur kostar y og aðilar sem leggja í púkk eru x þá myndu jöfnunar hljóða svona:
$8x - y = 3$
$7x - y = -4$

fylki jöfnunnar væri $\begin{pmatrix} 8 & -1 \\ 7 & -1 \end{pmatrix}$ og lausnarfylkið væri $\begin{pmatrix} 3 \\ -4 \end{pmatrix}$

leysti dæmið með numpy fyrir neðan, bara til gamans

[ ]:
```
import numpy as np
import numpy.linalg as la

A = np.array([[-1,8],[-1,7]])
b = np .array([3,-4])
sol = la.solve(A,b)
print(f'Gripurinn kostar {int(sol[0])} og það eru {int(sol[1])} kaupendur')
```

```
Gripurinn kostar 53 og það eru 7 kaupendur
```

1.6 E

1.6.1 a)

[ ]:
```
import numpy as np
import numpy.linalg as la

upph = np.array([1000,600,400])
leslie = np.array([
    [0.2,0.6,0.8],
    [0.8,0.0,0.0],
    [0.0,0.5,0.0]
])
```

1.6.2 b)

[ ]:
```
sea = sum(la.matrix_power(leslie,1)@upph) # stofn eftir ár
print(sea)
```

```
1980.0
```

1.6.3 c)

test nr. 1 fyrsta hugmyndin mín var að brute forcea þetta með while-lykkju, á meðan stofninn er stærri en 100, halda áfram. Lykkjan var ennþá að ganga eftir mínútu svo sú lausn gekk ekki upp.

```
[ ]: j = 1
     while(sum(la.matrix_power(leslie,j)@upph) > 100): j=+1
```

test nr.2 önnur hugmyndin var byggð á útkomuni úr test nr. 1. Svo virðist vera sem stofninn minnki ekki niður fyrir 100 í býsna langann tíma afh er það?

```
[ ]: years = [sum(la.matrix_power(leslie,i)@upph) for i in range(1,1000,100)]
     print(years)
```

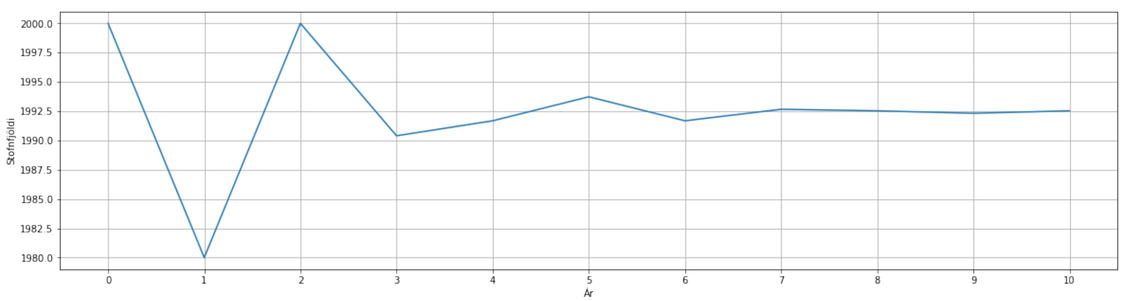
```
[1980.0, 1992.452830188685, 1992.4528301886908, 1992.4528301886962,
1992.4528301887021, 1992.4528301887076, 1992.4528301887133, 1992.452830188719,
1992.4528301887246, 1992.4528301887303]
```

Eins og sést, eftir fyrstu 100 árin er stofninn búinn að ná jafnvægi, ~1992.45 þá má með fullri vissu segja að stofninn mun ekki deyja út, bara til gamans þá plotta ég fyrstu 100 árin

```
[ ]: import matplotlib.pyplot as plt

     stofn = [sum(la.matrix_power(leslie,i)@upph) for i in range (11)]
     years = [j for j in range(11)]

     plt.figure(figsize=(20,5))
     plt.plot(years,stofn)
     plt.xticks(years)
     plt.grid(True)
     plt.xlabel('Ár')
     plt.ylabel('Stofnfjöldi')
     plt.show()
```



upprunalega var grafið upp í 100 ár en fjöldinn var orðinn stabíll eftir 10 þannig ég stytti skalann til að gera lesanlegra