

ISTx1001: Industriell statistikk

30. oktober 2023

Vi starter klokka 15:15

Timen blir tatt opp :)

Thea Bjørnland, Institutt for matematiske fag, NTNU

I dag: Starte med steg 5 i Del 1 av prosjektet

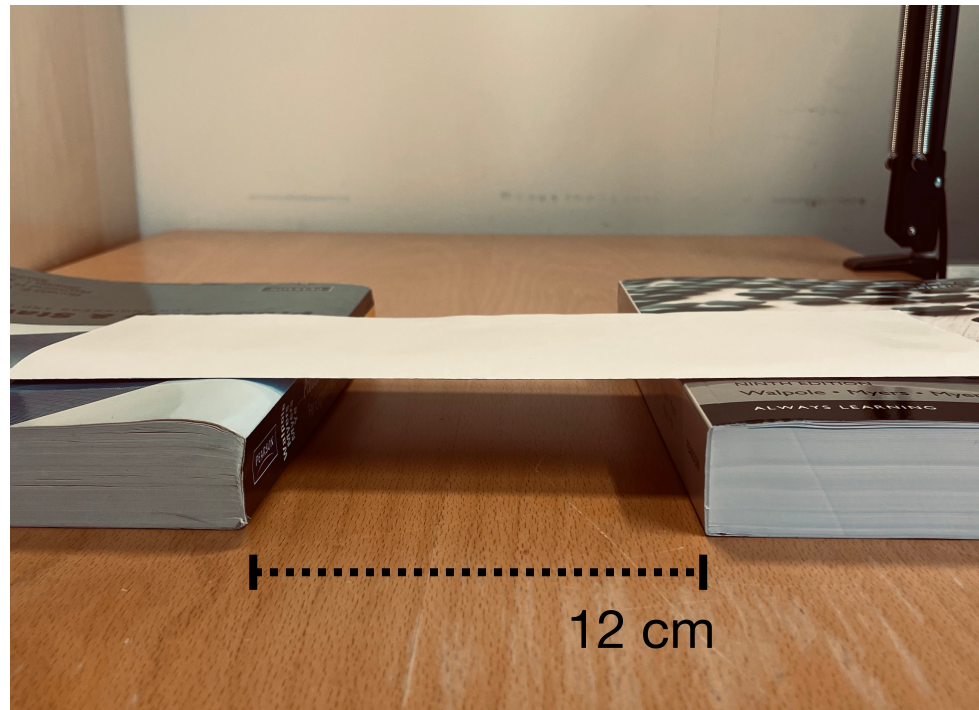
+ Python

Oppgavebeskrivelse

Målet med prosjektet er å planlegge, gjennomføre og diskutere et 2-nivå faktorielt forsøk. Dere står selv fritt til å velge problemstilling og eksperiment. Det kan være et laboratorieeksperiment eller et problem fra dagliglivet. Her er stegene gruppa skal gjennomføre

1. Bestem dere for en problemstilling som dere kan studere med et 2^k faktorielt forsøk. Hva er responsvariabelen og hva er mulige forklaringsvariabler? Hva er egnede nivåer for forklaringsvariablene? Merk at responsvariabelen Y må være en kontinuerlig stokastisk variabel.
2. Design et forsøk med 16, 24 eller 32 målinger. Bruk enten 3 eller 4 faktorer. Dersom dere bruker 3 faktorer, må dere ha minst ett gjentak.
3. Planlegg innsamling av data.
4. Samle inn data.
5. Analyser data.
6. Skriv en rapport.

Papirbro - hvor mye vekt tåler den?



4.35 gram

Designmatrise

$x_1(A)$	$x_2(B)$	$x_3(C)$
-1	-1	-1
+1	-1	-1
-1	+1	-1
+1	+1	-1
-1	-1	+1
+1	-1	+1
-1	+1	+1
+1	+1	+1

Resultater

$x_1 (A)$	$x_2 (B)$	$x_3 (C)$
-1	-1	-1
+1	-1	-1
-1	+1	-1
+1	+1	-1
-1	-1	+1
+1	-1	+1
-1	+1	+1
+1	+1	+1

y1	y2
8.70	4.35
17.40	17.40
0.00	4.35
8.70	8.70
8.70	8.70
30.45	34.80
4.35	4.35
13.05	21.75

Multipel lineær regresjon

Multipel lineær regresjon

Python: format på dataene

```
colnames = ["x1", "x2", "x3", "y"]
rep1 = pd.DataFrame(np.column_stack([X, y1]), columns=colnames)
rep2 = pd.DataFrame(np.column_stack([X, y2]), columns=colnames)

df = pd.concat([rep1, rep2])

print(df)
```

	x1	x2	x3	y
0	-1.0	-1.0	-1.0	8.70
1	1.0	-1.0	-1.0	17.40
2	-1.0	1.0	-1.0	0.00
3	1.0	1.0	-1.0	8.70
4	-1.0	-1.0	1.0	8.70
5	1.0	-1.0	1.0	30.45
6	-1.0	1.0	1.0	4.35
7	1.0	1.0	1.0	13.05
0	-1.0	-1.0	-1.0	4.35
1	1.0	-1.0	-1.0	17.40
2	-1.0	1.0	-1.0	4.35
3	1.0	1.0	-1.0	8.70
4	-1.0	-1.0	1.0	8.70
5	1.0	-1.0	1.0	34.80
6	-1.0	1.0	1.0	4.35
7	1.0	1.0	1.0	21.75

Python: regresjon

```
modell = smf.ols('y~x1+x2+x3 + x1*x2 + x2*x3 + x1*x3', data=df).fit()  
print(modell.summary())
```

OLS Regression Results

```
=====
Dep. Variable:          y      R-squared:          0.947
Model:                OLS     Adj. R-squared:       0.911
Method:             Least Squares   F-statistic:       26.72
Date:                Thu, 26 Oct 2023   Prob (F-statistic): 3.01e-05
Time:                  16:48:57   Log-Likelihood:    -35.260
No. Observations:      16     AIC:              84.52
Df Residuals:          9     BIC:              89.93
Df Model:              6
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	12.2344	0.731	16.745	0.000	10.582	13.887
x1	6.7969	0.731	9.303	0.000	5.144	8.450
x2	-4.0781	0.731	-5.582	0.000	-5.731	-2.425
x3	3.5344	0.731	4.837	0.001	1.882	5.187
x1:x2	-1.9031	0.731	-2.605	0.029	-3.556	-0.250
x2:x3	-0.8156	0.731	-1.116	0.293	-2.468	0.837
x1:x3	2.4469	0.731	3.349	0.009	0.794	4.100

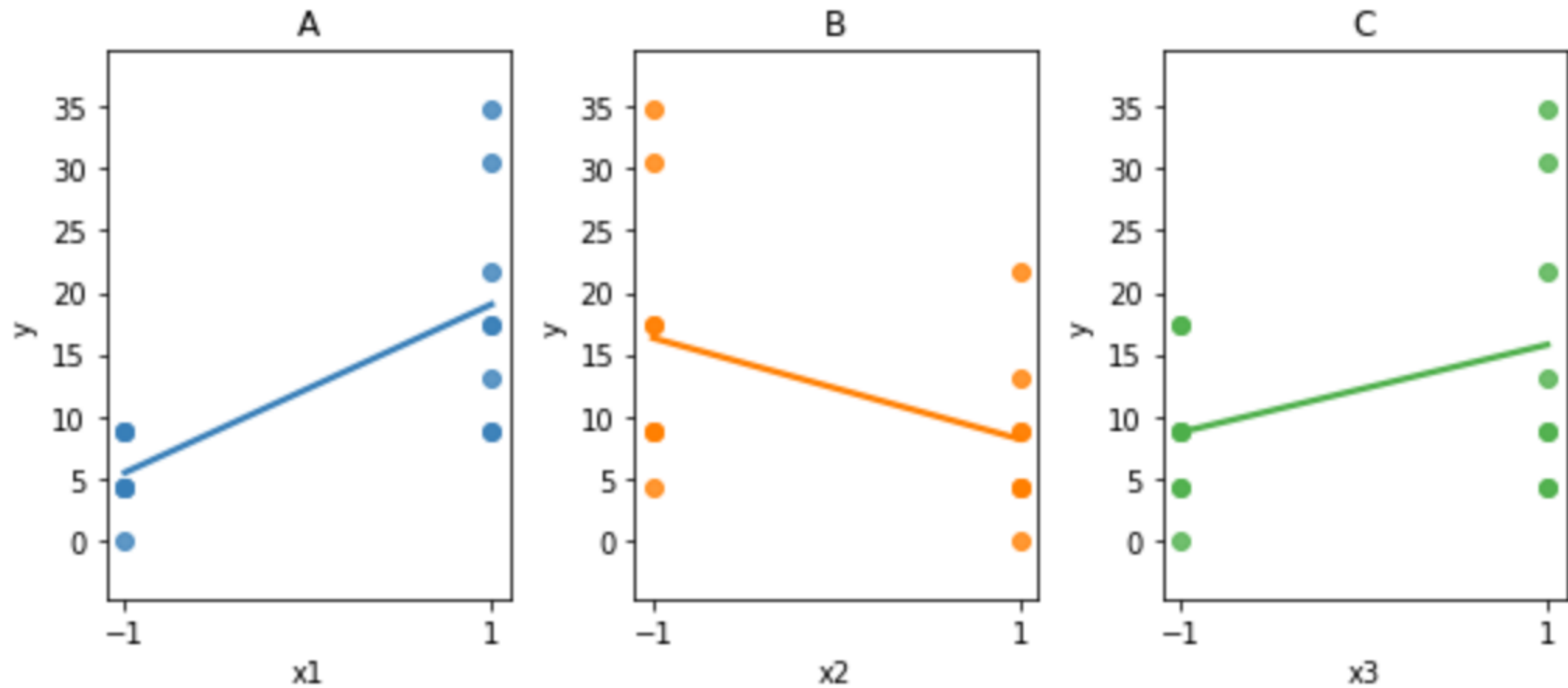
```
=====
Omnibus:              1.224   Durbin-Watson:          1.642
Prob(Omnibus):         0.542   Jarque-Bera (JB):        0.454
Skew:                  -0.412   Prob(JB):                0.797
Kurtosis:              3.037   Cond. No.                1.00
=====
```

Python: regresjon -> hovedeffekter

	coef	std err	t	P> t	[0.025	0.975]
Intercept	12.2344	0.731	16.745	0.000	10.582	13.887
x1	6.7969	0.731	9.303	0.000	5.144	8.450
x2	-4.0781	0.731	-5.582	0.000	-5.731	-2.425
x3	3.5344	0.731	4.837	0.001	1.882	5.187
x1:x2	-1.9031	0.731	-2.605	0.029	-3.556	-0.250
x2:x3	-0.8156	0.731	-1.116	0.293	-2.468	0.837
x1:x3	2.4469	0.731	3.349	0.009	0.794	4.100

Hovedeffektsplott

Hovedeffekter



Python: regresjon -> hovedeffekter

	coef	std err	t	P> t	[0.025	0.975]
Intercept	12.2344	0.731	16.745	0.000	10.582	13.887
x1	6.7969	0.731	9.303	0.000	5.144	8.450
x2	-4.0781	0.731	-5.582	0.000	-5.731	-2.425
x3	3.5344	0.731	4.837	0.001	1.882	5.187
x1:x2	-1.9031	0.731	-2.605	0.029	-3.556	-0.250
x2:x3	-0.8156	0.731	-1.116	0.293	-2.468	0.837
x1:x3	2.4469	0.731	3.349	0.009	0.794	4.100

Tips!

Det ligger flere notatbøker på huben
der jeg har gjort brukt Python for å estimere
effekter i 2-nivå faktorielle forsøk :)