



MAS-223-G

Semesterprosjekt

Gruppe 51

OBLIGATORISK OPPGAVE 1A

JOHANSSON, INGE JOHAN

1 Oppgave_1b_2

I denne oppgaven måles 30 terning kast med høydene *20cm*, *25cm*, *30cm*, *35cm*, *40cm* og *50cm*. Det blir utført 5 kast i vær høyde, for å sikre en variasjon for oppgave b så kastes terningen i forskjellig rekkefølge. Figur 1.1 viser en tabell av alle terningkastene og figur 1.2 viser kun di 5 første kastene.

Hoyde	Nummer	Lengde
40	5	118
40	4	94
20	4	47
20	1	41
20	2	23
20	2	48
40	2	87
40	1	105
50	3	116
30	4	63
35	2	78
30	3	50
35	4	99
20	2	32
50	2	129
30	2	55
50	1	69
50	6	150
50	4	150
45	6	145
45	6	142
30	2	88
45	2	148
35	6	98
45	1	123
35	4	102
45	5	139
35	3	103
30	5	87
40	5	150

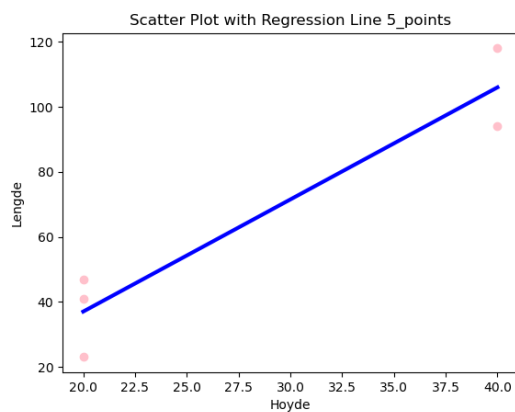
Figur 1.1: Lste av alle kastene

Hoyde	Nummer	Lengde
40	5	118
40	4	94
20	4	47
20	1	41
20	2	23

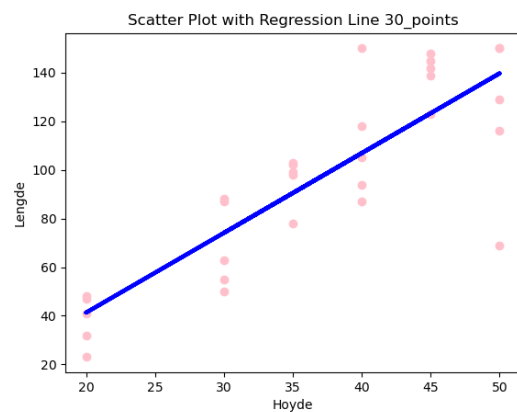
Figur 1.2: Liste med di 5 øverste kastene

2 Resultater

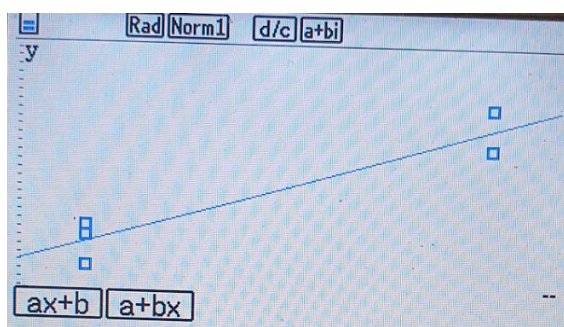
Figur 2.5 viser plottet av de fem første kastene å regresjons linjen mellom punktene. Figur 2.6 Viser plottet av hele datasettet å legger regresjons linjen mellom alle kastene. Legg inn formel for regresjons linje



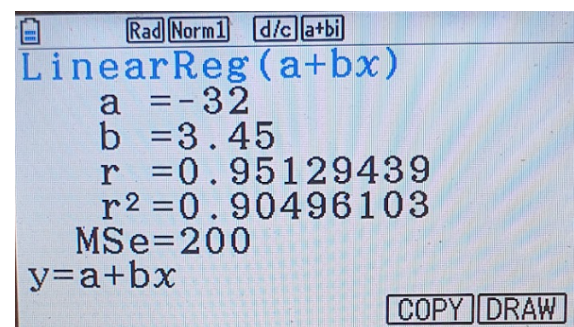
Figur 2.1: Scatterplot og regresjonslinjen for de 5 første kastene



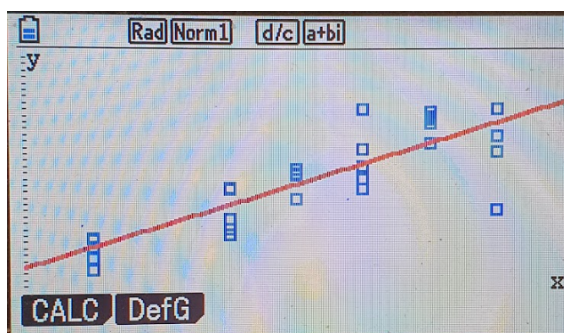
Figur 2.2: Scatterplot og regresjonslinjen for alle kastene



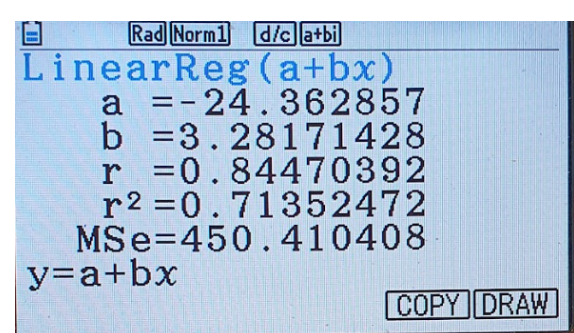
Figur 2.3: Graf på kalkulasjonene med 5 punkter



Figur 2.4: Kalkulasjonene på kalkulator med 5 punkter



Figur 2.5: Graf på kalkulasjonene med 30 punkter



Figur 2.6: Kalkulasjonene på kalkulator med 30 punkter

Under så er alle kalkulasjonene som ble vist på kalkualtor som stemmer overens med kalkulasjonene i matlab skriptet.

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = 12611.49 \quad (2.1)$$

$$SE = \sqrt{\frac{SSE}{n-p}} = 21.22 \quad (2.2)$$

```
alpha (5_points): [-32.]
beta (5_points): [3.45]
SSE (5_points): 600.0
SE (5_points): 14.142135623730951
alpha (30_points): [-24.36285714]
beta (30_points): [3.28171429]
SSE (30_points): 12611.491428571428
SE (30_points): 21.222874644196185
```

Figur 2.7: Kalkulasjonene fra Python skriptet med både 5 punkter og 30 punkter

oblig_1b/terningkast_2.py

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 # Importing csv files
6 p1 = pd.read_csv("oblig_1b/Terning_20.csv")
7 # Function to perform linear regression, plot, calculate SSE, and calculate SE
8 def linear_regression_and_plot(df, x_col, y_col, num_points, file_suffix):
9     # Create a new DataFrame with columns "1" and the independent variable
10    X = pd.DataFrame({
11        "Intercept": np.ones(num_points),
12        x_col: df[x_col].iloc[0:num_points]
13    })
14
15    # Define y as the dependent variable
16    y = df[y_col].iloc[0:num_points]
17
18    # Convert the DataFrames to numpy arrays
19    X_np = X.values
20    y_np = y.values.reshape(-1, 1)
21
22    # Calculating regression line
23    # Calculate the dot product of the matrix and its transpose
24    XtX = np.dot(X_np.T, X_np)
25    XtX_inv = np.linalg.inv(XtX)
26    Xty = np.dot(X_np.T, y_np)
27
28    # Calculate beta coefficients
29    beta = np.dot(XtX_inv, Xty)
30    alpha = beta[0]
31    print(f'alpha ({file_suffix}): {alpha}')
32    print(f'beta ({file_suffix}): {beta[1]}')
33
34    # Calculate regression line and residuals
35    regression_line = np.dot(X_np, beta)
36    residuals = y_np - regression_line
37
38    # Calculate SSE (Sum of Squared Errors)
39    SSE = np.sum(residuals**2)
40    print(f'SSE ({file_suffix}): {SSE}')
41
42    # Calculate SE (Standard Error of the regression)
43    # Degrees of freedom: num_points - number of parameters
44    dof = num_points - len(beta)
45    SE = np.sqrt(SSE / dof)
46    print(f'SE ({file_suffix}): {SE}')
47
48    # Plotting the scatter plot and regression line
49    plt.scatter(df[x_col][0:num_points], df[y_col][0:num_points], color="pink",
50    marker="o")
51    plt.plot(X_np[:, 1], regression_line, color="blue", linewidth=3)
52    plt.xlabel(x_col)
53    plt.ylabel(y_col)
54    plt.title(f'Scatter Plot with Regression Line {file_suffix}')
55    plt.savefig(f'oblig_1b/plot_{file_suffix}.png')
56    plt.show()
```

```
57     return alpha, beta[1], SSE, SE
58
59 # Perform linear regression, plot, calculate SSE, and calculate SE for the first
   subset
60 alpha_5, beta_5, SSE_5, SE_5 = linear_regression_and_plot(p1, "Hoyde", "Lengde", 5,
   "5_points")
61
62 # Perform linear regression, plot, calculate SSE, and calculate SE for the all data
   subset
63 alpha_30, beta_30, SSE_30, SE_30 = linear_regression_and_plot(p1, "Hoyde", "Lengde"
   , 30, "30_points")
64
```