TRABALHO FINAL

METODOS NUMERICOS

"Curve Fitting"

Husi:

Naran: Abraão de Jesus Ximenes

NRE: 20170204002

Turma: A

DEPARTAMENTO ENGENHARIA INFORMATICA

FACULDADE ENGENHARIA CIENSIA E TECNOLOGIA

UNIVERSIDADE NASIONAL TIMOR LOROSA'E

Hera, 2022

1. **Curve Fitting**.

1.1 Definisaun Curve Fitting.

Curve Fitting hanesan metode ida ne'ebe uja hodi bele halo estimasaun kurva/Liña ka funsaun matematica ne'ebe reprezenta pontus ba dadus ne'ebe barak. Curve Fitting bele involve mos ho interpolation iha ne'ebé kecocokan ba data ne'ebé mak presiza ka funsaun ne'ebé konstrui tenke sesuai ho data ne'ebé refere. Relasaun mos ho Regression Linear ne'e haree liu ba iha kestaun statistical interference hanesan kurva hira mak la sesuai ho dadus ne'ebé mak iha ne'ebé observa liu husi nia error sira. Kurva ne'ebé mak iha fasiliza ita hodi utiliza ba haree dadus sira atu menyimpulkan valor funsaun ne'ebé mak la available iha dadus, no halo summarize relasaun entre variavel rua ba leten.

Implementasaun iha Python.

In [123]:
```
import numpy as np
from sklearn.linear_model import LinearRegression
from matplotlib import pyplot as plt
```

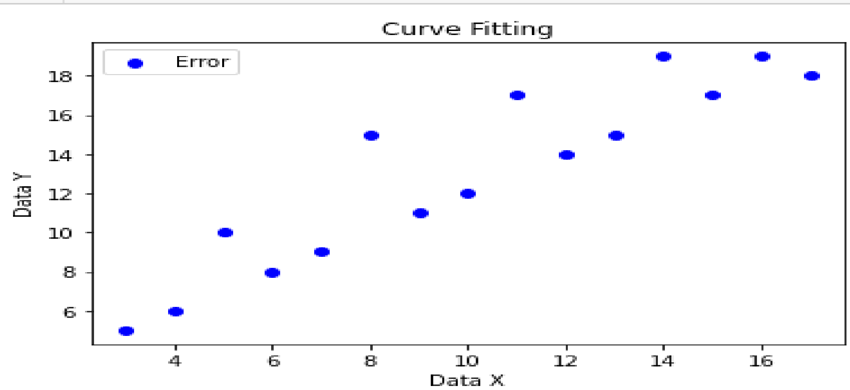
In [124]:
```
x = np.array([3,4,5,6,7,8,9,10,11,12,13,14,15,16,17]).reshape((-1, 1))
y = np.array([5,6,10,8,9,15,11,12,17,14,15,19,17,19,18]).reshape((-1, 1))

```

In [125]:
```
# Fitting Linear Regression to the dataset
from sklearn.linear_model import LinearRegression
lin = LinearRegression()

lin.fit(x, y)
```

Out[125]: LinearRegression()

In [126]:
```
# Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree = 4)
X_poly = poly.fit_transform(x)

poly.fit(X_poly, y)
lin2 = LinearRegression()
lin2.fit(X_poly, y)
```

In [136]:
```
plt.scatter(x, y, color = 'blue', label='Error')

plt.title('Curve Fitting')
plt.xlabel('Data X')
plt.ylabel('Data Y')
plt.legend()

plt.show()
```



2. Least Square Regression Method.

2.1 Definisaun Least Square Regression.

Least Square hanesan método neebé forma regresaun matemátika ida neebé uza hodi determina liña dadus neebé adekuadu tuir liña dadus lubuk ida, hodi fornese demonstrasaun vizuál ida kona-ba relasaun entre pontu dadus sira. Kada pontu dadus reprezenta relasaun entre variavel independente ida no variável la depende.

Least Square mak prosedimentu estatístiku ida atu hetan adekuadu liu ba pontu dadus lubuk ida, hodi hamenus kuantidade kompensasaun ka residuasaun pontu sira husi kurva ne 'ebé ajusta.

Deklarante regresaun uza atu prevee komportamentu hosi variavel dependente sira. Métodu ke' e liña-diak ne 'e fornese razaun jerál hodi halo kolokasaun liña ida ne' ebé di 'ak liu entre pontu dadus sira ne' ebé oras ne 'e estuda hela.

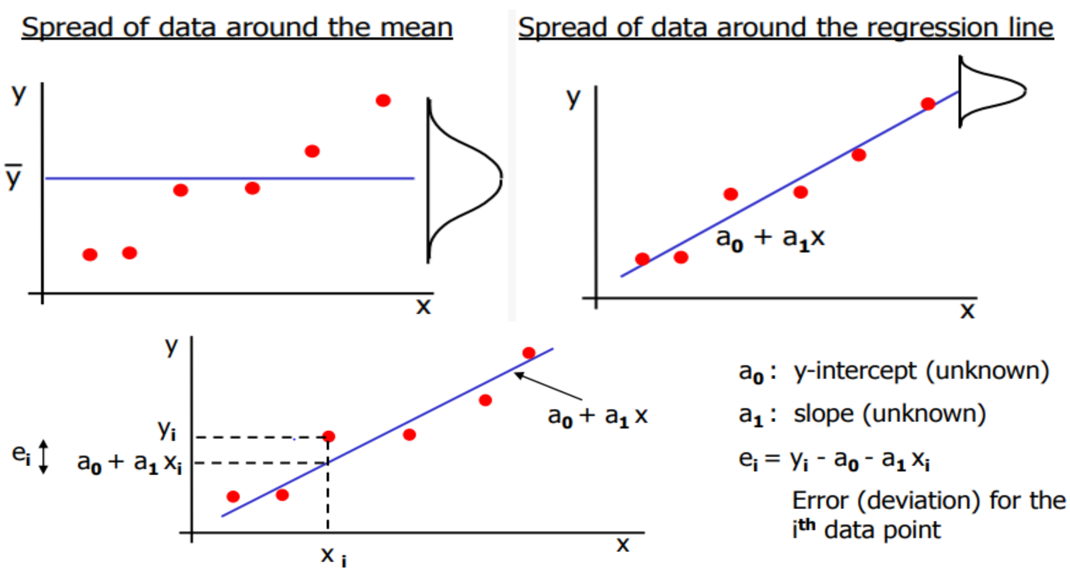
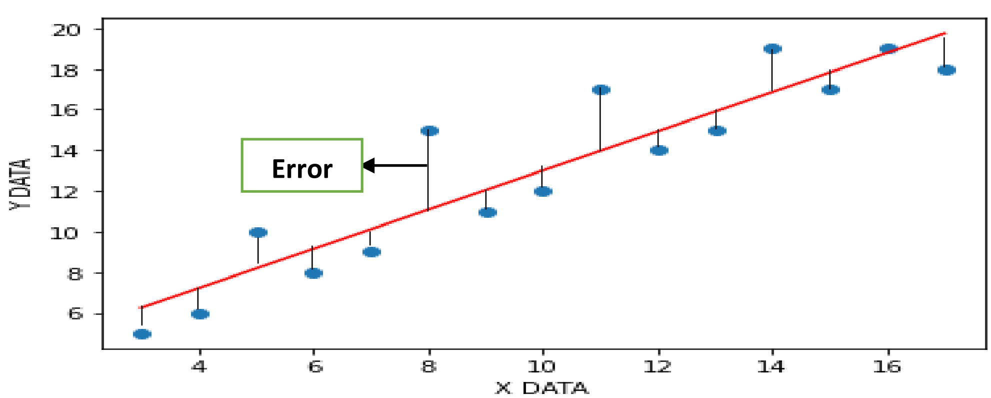


Chart Least Square iha Python

Error sira mak pontu sira ne'ebé mak iha straight line ne'e.

Ŷ=ax + b ⟶ Error=Ŷ-yi



Resultadu Least Square iha python

```
In [75]:
# Least Square
y_pred = model.predict(x)
print('predicted response:', y_pred, sep='\n')

predicted response:
[[ 6.25      ]
 [ 7.21428571]
 [ 8.17857143]
 [ 9.14285714]
 [10.10714286]
 [11.07142857]
 [12.03571429]
 [13.        ]
 [13.96428571]
 [14.92857143]
 [15.89285714]
 [16.85714286]
 [17.82142857]
 [18.78571429]
 [19.75      ]]
```

```
In [76]:
Error = y_pred - y
print(Error)

[[ 1.25      ]
 [ 1.21428571]
 [-1.82142857]
 [ 1.14285714]
 [ 1.10714286]
 [-3.92857143]
 [ 1.03571429]
 [ 1.        ]
 [-3.03571429]
 [ 0.92857143]
 [ 0.89285714]
 [-2.14285714]
 [ 0.82142857]
 [-0.21428571]
 [ 1.75      ]]
```

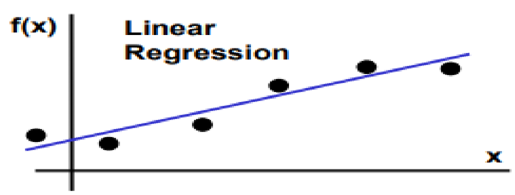
3. Linear, Polynomial Regression Method.

3.1 Definisaun Linear Regression Method.

Regresaun simples nee uza atu halo estimativa ba relasaun entre variabel kuantitativu rua. Bele uza regresaun simples bainhira Ita-boot hakarak hatene: Oinsá relasaun forte entre variavel rua (ezemplu relasaun entre udan no erozaun rai). Valór husi variavel dependente ho valór balu husi variavel independente (n.e., kuantidade erozaun rai iha nivel udan nian balu).

Formula: **Grafiku** :

$$y = b_0 + b_1 x_1$$



$$\boxed{a_1 = \frac{n\sum(x_i y_i) - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \qquad a_0 = \bar{y} - a_1\bar{x}}$$

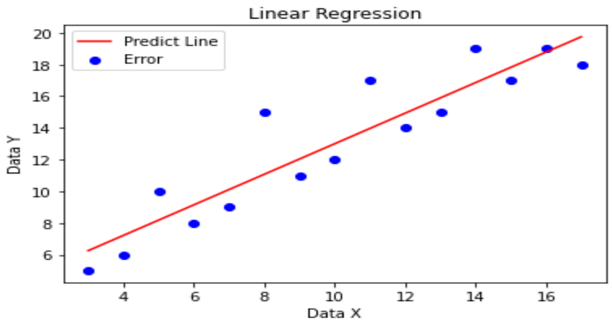
$$r^2 = \frac{S_t - S_r}{S_t} \rightarrow S_t = \sum_{i=1}^{n}(y_i - \bar{y})^2 \rightarrow S_r = \sum_{i=1}^{n}(y_i - a_0 - a_1 x_i)^2$$

r^2 = Coefficient Correlation.

Sr = Sum of the Square of the residual.

Resultadu Chart Linear Regression Python

```
In [68]:
# Visualising the Linear Regression results
plt.scatter(x, y, color = 'blue', label='Error')

plt.plot(x, lin.predict(x), color = 'red', label='Predict Line')
plt.title('Linear Regression')
plt.xlabel('Data X')
plt.ylabel('Data Y')
plt.legend()

plt.show()
```



Resultadu Slope no Intercept iha Python

```
In [74]:
print('intercept:', model.intercept_)
print('slope:', model.coef_)
```

```
intercept: [3.35714286]
slope: [[0.96428571]]
```

```
In [73]:
r_sq = model.score(x, y)
print('coefficient of determination:', r_sq)
```

```
coefficient of determination: 0.8508403361344538
```

3.2 Definisaun Polynomial Regression Method.

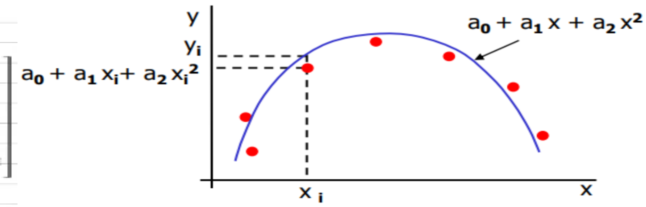
Regresaun polynomial hanesan forma ida husi regresaun hodi analiza ida ne’ebe iha relasaun entre
variável independente entre variável independente no variável dependente nee sai modelu ba nivel polynomial iha x. regresaun Polynomial ba relasaun naun linear entre valor x no kondisaun koresponde, denominadu E(y |x).
Maske regresaun polynomial ba fitar hanesan modelu naun linear ba dadus, nudár kalkulasaun estatistika ba problema ne 'e linear, katak regresaun ba funsaun E(y | x) ne' e linear iha parámetru ne 'ebé la hatene tuir dadus. Tanba

nee, regresaun polynomial konsidera hanesan kazu espesial ida husi regresaun linear oioin.

Formula: **Grafiku:**

$$y = a_0 + a_1 x + a_2 x^2$$

$$\underbrace{\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix}}_{A} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}}_{B}$$



Resultadu fó sai iha Python

```
In [89]:
# Visualising the Polynomial Regression results
plt.scatter(x, y, color = 'blue', label='Error')

plt.plot(x, lin2.predict(poly.fit_transform(x)), 'b--', label='Poly Line')
plt.title('Polynomial Regression')
plt.xlabel('DATA X')
plt.ylabel('DATA Y')
plt.legend()

plt.show()
```

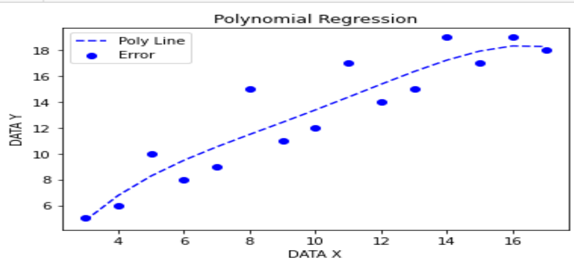
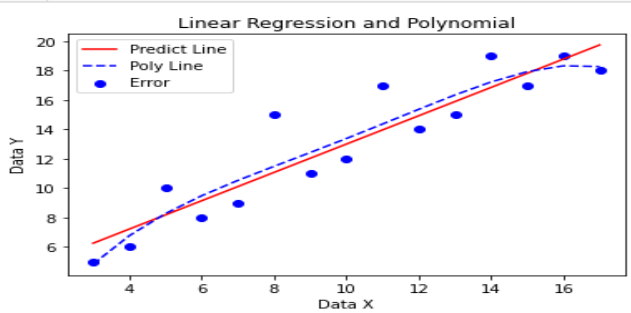
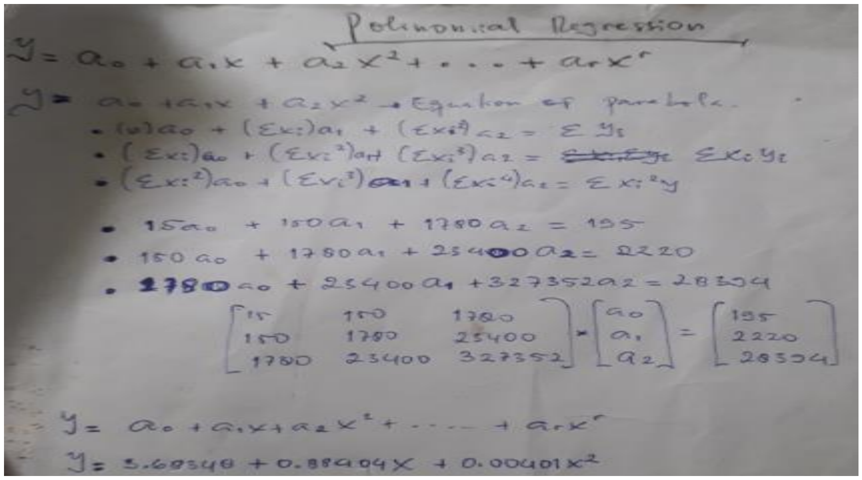


Chart Linear no Polynomial Regression implementa iha python.

```
In [90]:
# Visualising the Linear Regression results
plt.scatter(x, y, color = 'blue', label='Error')

plt.plot(x, lin.predict(x), color = 'red', label='Predict Line')
plt.plot(x, lin2.predict(poly.fit_transform(x)), 'b--', label='Poly Line')
plt.title('Linear Regression and Polynomial')
plt.xlabel('Data X')
plt.ylabel('Data Y')
plt.legend()
plt.show()
```





Polinomial Regression

$y = a_0 + a_1 x + a_2 x^2 + \dots + a_r x^r$

$y = a_0 + a_1 x + a_2 x^2 \rightarrow$ Equation of parabola.

- $(n) a_0 + (\Sigma x_i) a_1 + (\Sigma x_i^2) a_2 = \Sigma y_i$
- $(\Sigma x_i) a_0 + (\Sigma x_i^2) a_1 + (\Sigma x_i^3) a_2 = \Sigma x_i y_i$
- $(\Sigma x_i^2) a_0 + (\Sigma x_i^3) a_1 + (\Sigma x_i^4) a_2 = \Sigma x_i^2 y$

- $15 a_0 + 150 a_1 + 1780 a_2 = 195$
- $150 a_0 + 1780 a_1 + 25400 a_2 = 2220$
- $1780 a_0 + 25400 a_1 + 327352 a_2 = 28394$

$$\begin{bmatrix} 15 & 150 & 1780 \\ 150 & 1780 & 25400 \\ 1780 & 23400 & 327352 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 195 \\ 2220 \\ 28394 \end{bmatrix}$$

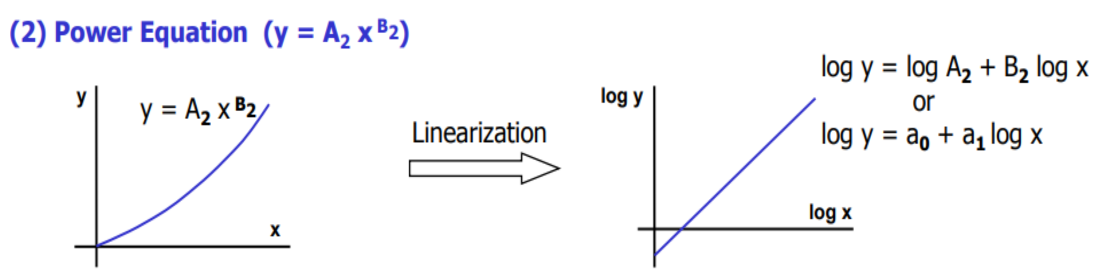
$y = a_0 + a_1 x + a_2 x^2 + \dots + a_r x^r$

$y = 5.69348 + 0.88404x + 0.00401x^2$

4. **Do the Linearization of Nonlinear Equation and then Fitting the Curve.**
   Non Linear Regression hanesan forma analizasaun ba regresaun ne'ebé mak observa dadus ne'ebé model tiha ona husi funsaun ne'ebé mak hanesan konbinasaun linear husi parametru model no depende ba model ida ka variavel independente barak.

4.1 Power Regression

Power Regression model parte ida husi non-linear regression model $\mathbf{y = ax^b}$ ne'ebé ho valor a no b ne'e konstanta ne'ebé ita kalcula hamutuk ho logaritmu.



$$y = ax^b \longrightarrow \log y = \log ax^b \longrightarrow \log y = \log a + b \log x$$

$$p = \log y; \quad A = \log a; \quad B = b; \quad q = \log x;$$

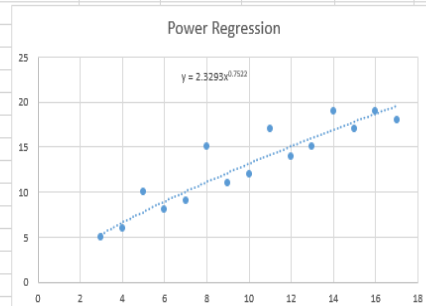
$$\overline{y} = \Sigma\, y_i / n\ , \ \overline{q} = \Sigma \log x_i / n\ , \ \overline{p} = \Sigma \log y_i / n$$

$$B = \frac{n \sum q_i p_i - \sum q_i \sum p_i}{n \sum q_i^2 - (\sum q_i)^2}, \quad A = \bar{p} - B\,\bar{q}$$

B = Slope B
A = Intercept A
Ȳ = Mean Y

Implementa iha excel.

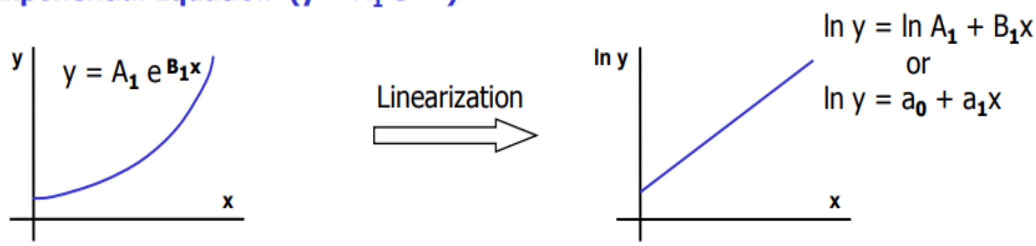
		Power Regressions					
X	Y	qi=log(x)	pi=log(y)	qi.pi	qi2	Dt2=(Yi-Y)^2	D2=(Yi-a-bxi)2
3	5	0.477121255	0.698970004	0.333493445	0.227644692	64	3.68064225
4	6	0.602059991	0.77815125	0.468493735	0.362476233	49	8.51764225
5	10	0.698970004	1	0.698970004	0.488559067	9	47.86564225
6	8	0.77815125	0.903089987	0.702740603	0.605519368	25	24.19164225
7	9	0.84509804	0.954242509	0.806428474	0.714190697	16	35.02864225
8	15	0.903089987	1.176091259	1.06211624	0.815571525	4	142.0506423
9	11	0.954242509	1.041392685	0.993741169	0.910578767	4	62.70264225
10	12	1	1.079181246	1.079181246	1	1	79.53964225
11	17	1.041392685	1.230448921	1.281380506	1.084498725	16	193.7246423
12	14	1.079181246	1.146128036	1.236879882	1.164632162	1	119.2136423
13	15	1.113943352	1.176091259	1.31009904	1.240869792	4	142.0506423
14	19	1.146128036	1.278753601	1.465615353	1.313609474	36	253.3986423
15	17	1.176091259	1.230448921	1.447120221	1.38319065	16	193.7246423
16	19	1.204119983	1.278753601	1.539772764	1.449904933	36	253.3986423
17	18	1.230448921	1.255272505	1.5445487	1.514004548	25	222.5616423
150	**195**	**14.25003852**	**16.22701579**	**15.97058138**	**14.27525063**	306	1781.649634



4.2 Exponential Regression

Exponencial Regression model parte ida husi non-linear regression model $\mathbf{y = ae^{bx}}$ ne'ebé ho valor a no b ne'e konstanta ne'ebé ita kalcula hamutuk ho logaritmu natural.

(1) Exponential Equation $(y = A_1 e^{B_1 x})$



$$y = a\,e^{bx} \longrightarrow \ln y = \ln ae^{bx} \longrightarrow \ln y = \ln a + bx$$

$$p = \ln y; \quad A = \ln a; \quad B = b; \quad q = x;$$

$$\bar{y} = \Sigma y_i / n \;, \; \bar{q} = \Sigma q_i / n \;, \; \bar{p} = \Sigma p_i / n \;:$$

$$B = \frac{n \Sigma q_i p_i - \Sigma q_i \Sigma p_i}{n \Sigma q_i^2 - (\Sigma q_i)^2} \;, \; A = \bar{p} - B \bar{q}$$

B = Slope B
A = Intercept A
$\bar{Y}$ = Mean Y

Implementa iha excel.

Exponencial Regressions						
X	**Y**	**pi=ln(y)**	**qi=X**	**qi.pi**	**qi2**	**qi=ln(x)**
3	5	1.609437912	3	4.828313737	9	1.098612289
4	6	1.791759469	4	7.167037877	16	1.098612289
5	10	2.302585093	5	11.51292546	25	1.098612289
6	8	2.079441542	6	12.47664925	36	1.098612289
7	9	2.197224577	7	15.38057204	49	1.098612289
8	15	2.708050201	8	21.66440161	64	1.098612289
9	11	2.397895273	9	21.58105746	81	1.098612289
10	12	2.48490665	10	24.8490665	100	1.098612289
11	17	2.833213344	11	31.16534678	121	1.098612289
12	14	2.63905733	12	31.66868796	144	1.098612289
13	15	2.708050201	13	35.20465261	169	1.098612289
14	19	2.944438979	14	41.22214571	196	1.098612289
15	17	2.833213344	15	42.49820016	225	1.098612289
16	19	2.944438979	16	47.11102367	256	1.098612289
17	18	2.890371758	17	49.13631988	289	1.098612289
150	**195**	**37.36408465**	**150**	**397.4664007**	**1780**	**16.47918433**