Q.1) Print first 10 naturals numbers using while loop ?

Ans:

```
i = 1
n = 11
while i<11 :
    Print(i)
    i = i+1
```

Q.2) Print the following pattern :

```
1 1 1 1
1 1 1
1 1
1
```

Ans :

```
n = 4
for i in range (n):
    for j in range (i, n):
        print ("1", end = " ")
    print ()
```

Q.3) calculate the sum of all numbers from 1 to a given number ?

Ans

```
n=0
for i in range (1,12)
  n+i=i
print ("The sum is", n)
```

Q.4) Write a program to print multiplication table of a given number.

Ans:

```
n = int(input("enter a number"))
for i in range(1,11):
    print("2", "x", j, "=", i*j)
    for j in range(1,11):
        print("2", "x", j, "=", i*j)
    print()
```

Ans:

```
for i in range(1,11):
    print("2", "x", i, "=", i*2)
```

Q.5) Display numbers from a list using loop?

Ans:

```
list = [1, 2, 3, 4, 5, 6, 7, 8, 9]
for list in list:
    print (list)
```

Output:

```
1
2
3
4
5
6
7
8
9
```

Q.6) Count the total number of digits in a number ?

Ans

```
n = '123456789'
len(n)
```

Q.7) Print the following pattern : Diamond shape ?

Ans

```
n = 5
for x in range(n):
    print (" " * (n-x), "*" * (2*x+1))
for x in range (n-2, -1, -1):
    print (" " * (n-x), "*" * (2*x+1))
```

Q. 8) Print list in reverse order using a loop ?

Ans

```
li = ['a', 'b', 'c', 'd', 'e']
for i in range (len(li) -1, -1, -1):
    print (li[i])
```

Q.9) Display numbers from -10 to -1 using for loop.?

Ans

```
for i in range (-10, 0):
    print (i)
```

Q.10) Use else block to display a message "Done" after successful execution of for loop ?

Ans

```
Name = "G"
for i in Name :
    if Name != "G":
        print ("hi")
    else :
        print ("Done")
```

Q.11) Write a program to display all prime numbers within a range?

Ans

```
for i in range (2,101):
    for j in range (2,101):
        if i%j==0:
            break
    if i==j:
        print(i,end=",")
```

Q.12) Display Fibonacci series up to 10 terms ?

Ans

```
n = int (input ("enter a number"))
n1 = 0
n2 = 1
For i in range (2, 10):
  n3 = n1+n2
   print (n3)
    n1 = n2
    n2 = n3
```

Q.13) Find the factorial of a given number?

Ans

```
n = int (input(" enter a number "))
fact = 1
for i in range (1, 6):
    fact = fact * i
print (fact)
```

Q.14) Reverse a given integer number?

```
n = [1, 4, 7, 5, 2, 3]
n.reverse()
print(n)
```

Q.15) Use a loop to display elements from given list present at odd index positions?

Ans

```
a = [10,20,30,40,50,60,70,80,90,100]
for i in range(1,9,2):
    print(a[i])
```

Q.16) Calculate the cube of all numbers from 1 to a given numbers.

Ans

```
def cube(num):
    return num * num * num
For i in {6}:
    result = cube(i)
    print('cube of', i, "=", result)
```

5.17) Find the sum of the series upto n terms ?

Ans

```
def sum(x):
    if x==0:
        return 0
    else:
        return ((x * (x+1)) /2)

sum(12)
```

Q.18) Print the following pattern :

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Ans

```
def pattern(n):
    num = 1
    for i in range(0, n):
        num = 1
        for j in range(0, i+1):
            print(num, end = " ")
            num = num + 1
        print("\r")
n = 5
pattern(n)
```

1) Given,
data = {2, 3, 1, 3, 2, 4}
and S.D = 1.5
From given data calculate mean i.e. $\mu = 2.5$
Now, the formula to calculate z-score:

$$z = \frac{x_i - \mu}{S.D.}$$

i) For $x = 2$

$$z_1 = \frac{2 - 2.5}{1.5} = -0.33$$

ii) For $x = 3$

$$z_2 = \frac{3 - 2.5}{1.5} = 0.33$$

Formula for Normalization is-

$$\text{Normalization} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

2)

One-Hot-Encoding is a technique used to convert categorical variables into a binary matrix, where each category is represented by a binary column.
In Pandas, the get_dummies function is commonly used for one-hot encoding.

3)

List all the transformers (function and power).
The question seems incomplete or ambiguous. If you're referring to mathematical transformations, it could include functions like square root, logarithm, exponential etc. with different powers.

4)

Linear regression assumes that the relationship between the independent and dependent variables is linear, and the residuals (the differences between actual and the predicted value) are normally distributed with constant variance.

5) Gradient desent is an optimization algorithm used to minimize the cost function in machine learning models it iteratively adjusts model parameters in the direction of steepest decrease of the cost The diagram typically shows a convergence towards the minimum point of cost function.

6) Pandas-Profiling is a library used for generating exploratory data analysis reports for a pandas Dataframe TO-Use-it :-

```
Python
Copy Code
import Pandas-Profiling
Profile =
Pandas-Profiling.ProfileReport(df)
Profile.to-file("output.html")
```

7) Draw the line for the following equation :

$$y = x^2$$

;

Since $y = x^2$

It represents a quadratic function, the graph is a parabola it opens upwards if the coiffic of x^2

It positive the specific shape and location of the parabola depend on the coifficient values.

8)

```
Python
copy code
# import necessary libraries.
 import seaborn as sns
 From sklearn.model_selection import train_test_split.
 From import train_test_split
 From sklearn.linear_model import
 Linear Regression
 From sklearn.matrics import
 mean_squared_error.

# Load the Dataset.
 mpg_data = sms.load_dataset ('mpg')

# check for missing values.
 print (mpg:data.isnull().sum())

# split the data into train and test sets
 x = mpg_data [['horse Power']]
 y = mpg_data ['mpg']
 x = train, x_test, y_train, y_test = train_test_split (x,y,
     test_size = 0,2'. random_state = 42)

# Create and fill the model
 model = LinearRegression()
 model.fit (x_train, y_train)

# Evaluate the model.
 mse = mean_squared_error (y_test, y_pred)
print f (f mean squared Error : {mse}')
```

This code imports the necessary libraries, loads the "mpg" dataset, checks for the missing values, splits the data into training and testing sets, builds a linear regression model, makes predictions and evaluates the model using mean squared error.