

In [ ]:

class LinearRegression:

```
def __init__(self):
    import numpy as np
    import pandas as pd
    X = [i for i in range(1,8)]
    Y = [1.5,3.8,6.7,9.0,11.2,13.6,16.0]
    dataset = pd.DataFrame(data={
        'X' : X,
        'Y' : Y
    })
    print(dataset)
    print('\n')
    pass

def find_m(self):
    dataset['XY'] = dataset['X'] * dataset['Y']
    dataset['X2'] = dataset['X'] ** 2
    dataset
    n = len(dataset)
    sum_x = dataset['X'].sum()
    sum_y = dataset['Y'].sum()
    sum_xy = dataset['XY'].sum()
    sum_x2 = dataset['X2'].sum()
    sum_X_h2 = sum_x**2
    n,sum_x,sum_y,sum_xy,sum_x2,sum_X_h2
    numerator_m = (n*(sum_xy)) - (sum_x*sum_y)
    numerator_m
    denominator_m = ((n*sum_x2) - sum_X_h2)
    denominator_m
    m = numerator_m/denominator_m
    print('value of m',m)
    print('\n')
    pass

def find_b(self):
    numerator_b = sum_y - (m * sum_x)
    denominator_b = n
    b = numerator_b/denominator_b
    print('value of b',b)
    print('\n')
    pass

def find_outputs(self):
    inputs = [i for i in range(1,8)]
    outputs =[m * X+b for X in dataset['X']]
    outputs
    dataset['outputs'] = outputs
    print("dataset with outputs")
```

```

print(dataset)
print('\n')
pass

def visualization(self):
    import matplotlib.pyplot as plt
    plt.scatter(dataset['X'],dataset['Y']) #original data
    plt.plot(dataset['X'],dataset['outputs'],c='r') #predicted data
    dataset_xy = pd.DataFrame(data={
        "X" : [i for i in range(1,15)]
    })

    outputs = [m*X+b for X in dataset_XY['X']]
    dataset_xy['output'] = outputs
    dataset_xy
    print("data set with forecasted outputs")
    print(dataset_xy)
    print('\n')

    plt.scatter(dataset['X'],dataset['Y']) #original data
    plt.plot(dataset['X'],dataset['outputs'],c='r') #predicted data
    plt.plot(dataset_xy['X'][6:],dataset_xy['output'][6:],c='y') #new
data
    plt.show()
    pass

```

```
oj = LinearRegression() oj.find_m() oj.find_b() oj.find_outputs() oj.visualization()
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [60]: **class** LinearRegression:

```
    def __init__(self):
        import numpy as np
        import pandas as pd
        X = [i for i in range(1,8)]
        Y = [1.5,3.8,6.7,9.0,11.2,13.6,16.0]
        dataset = pd.DataFrame(data={
            'X' : X,
            'Y' : Y
        })
        print(dataset)
        print('\\n')
        pass

    def find_m(self):
        dataset['XY'] = dataset['X'] * dataset['Y']
        dataset['X2'] = dataset['X'] ** 2
        dataset
        n = len(dataset)
        sum_x = dataset['X'].sum()
        sum_y = dataset['Y'].sum()
        sum_xy = dataset['XY'].sum()
        sum_x2 = dataset['X2'].sum()
        sum_X_h2 = sum_x**2
        n,sum_x,sum_y,sum_xy,sum_x2,sum_X_h2
        numerator_m = (n*(sum_xy)) - (sum_x*sum_y)
        numerator_m
        denominator_m = ((n*sum_x2) - sum_X_h2)
        denominator_m
        m = numerator_m/denominator_m
        print('value of m',m)
        print('\\n')
        pass

    def find_b(self):
        numerator_b = sum_y - (m * sum_x)
        denominator_b = n
        b = numerator_b/denominator_b
        print('value of b',b)
        print('\\n')
        pass
```

```

def find_outputs(self):
    inputs = [i for i in range(1,8)]
    outputs =[m * X+b for X in dataset['X']]
    outputs
    dataset['outputs'] = outputs
    print("dataset with outputs")
    print(dataset_xy)
    print('\n')
    pass

def visualization(self):
    import matplotlib.pyplot as plt
    plt.scatter(dataset['X'],dataset['Y']) #original data
    plt.plot(dataset['X'],dataset['outputs'],c='r') #predicted data
    dataset_xy = pd.DataFrame(data={
        "X" : [i for i in range(1,15)]
    })

    outputs = [m*X+b for X in dataset_xy['X']]
    dataset_xy['output'] = outputs
    dataset_xy
    print("data set with forecasted outputs")
    print(dataset_xy)
    print('\n')
    plt.scatter(dataset['X'],dataset['Y']) #original data()
    plt.plot(dataset['X'],dataset['outputs'],c='r') #predicted data
    plt.plot(dataset_xy['X'][6:],dataset_xy['output'][6:],c='y') #new
    plt.show()
    pass

oj = LinearRegression()
oj.find_m()
oj.find_b()
oj.find_outputs()
oj.visualization()

```

Input In [60]

```
plt.scatter(dataset['X'],dataset['Y']) #original data
```

^

IndentationError: unexpected indent

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: