RegressionPlots

October 31, 2020

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
chickweight = pd.read_csv('chick_weight.csv')
chickweight = chickweight.drop(['Unnamed: 0'], axis=1)
weight = 0
time = 1
chick = 2
diet = 3
```

[3]:
```
chickweight.head()
```

[3]:

	weight	Time	Chick	Diet
0	42	0	1	1
1	51	2	1	1
2	59	4	1	1
3	64	6	1	1
4	76	8	1	1

[4]:
```
#chick1_0 = chickweight.iloc(Chick==1)
adjchickweight = chickweight.copy()
allChicks = chickweight.Chick.unique()

initialweight = {}
for chick in allChicks:
    initialweight[chick]=chickweight.loc[(chickweight.Chick==chick) &␣
 ↪(chickweight.Time==0)].weight

initialweight

for i in range(len(adjchickweight)):
    adjchickweight.weight[i] = chickweight.weight[i]-initialweight[chickweight.
 ↪Chick[i]]

chickweight = adjchickweight.copy()
```

[5]:
```
chickweight.head
```

[5]:
```
<bound method NDFrame.head of      weight  Time  Chick  Diet
0           0     0      1     1
1           9     2      1     1
2          17     4      1     1
3          22     6      1     1
4          34     8      1     1
..        …   …     …   …
573       134    14     50     4
574       164    16     50     4
575       193    18     50     4
576       223    20     50     4
577       223    21     50     4

[578 rows x 4 columns]>
```

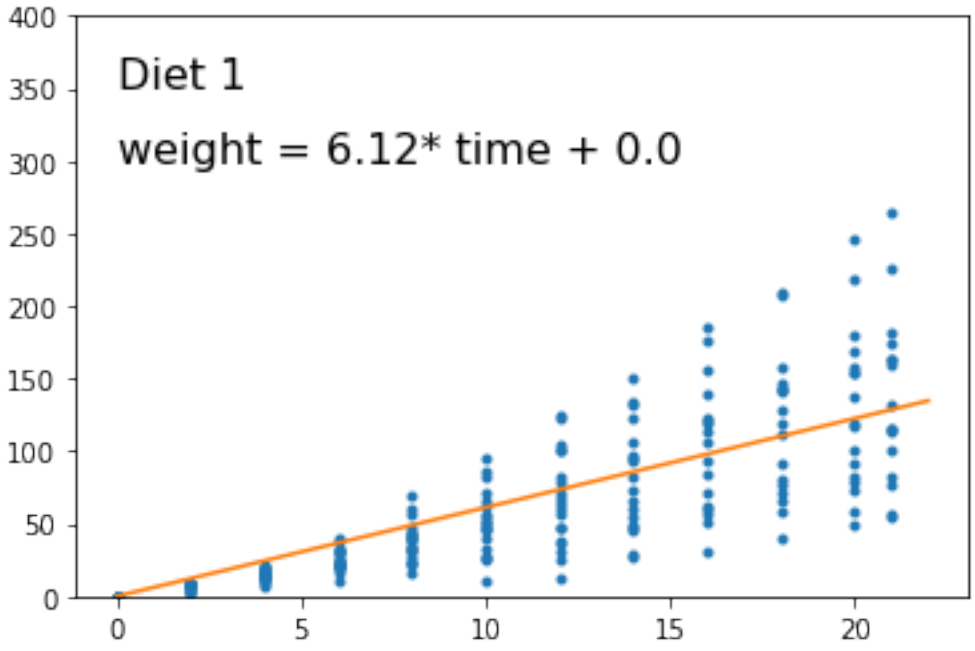
[6]:
```
def get_line_and_plot(X, Y, diet):
    z = np.polyfit(X, Y, 1)
    p = np.poly1d(z)
    xp = np.linspace(0, 22, 100)
    plt.plot(X,Y,'.',xp, p(xp),'-')
    polyfittext = "weight = "+ str(round(z[0],2)) + "* time + " +␣
↪str(round(z[1],2))
    plt.text(0,350,"Diet "+ str(diet), fontsize=16)
    plt.text(0,300,polyfittext, fontsize=16)
    plt.ylim(0,400)
    plt.show()
    return z
```

[7]:
```
def get_line_and_plot_nointercept(X, Y, diet):
    #z = np.polyfit(X, Y, 1)
    A = np.vstack([X,np.zeros(len(X))]).T
    z = np.linalg.lstsq(A, Y, rcond=None)[0]
    p = np.poly1d(z)
    xp = np.linspace(0, 22, 100)
    plt.plot(X,Y,'.',xp, p(xp),'-')
    polyfittext = "weight = "+ str(round(z[0],2)) + "* time + " +␣
↪str(round(z[1],2))
    plt.text(0,350,"Diet "+ str(diet), fontsize=16)
    plt.text(0,300,polyfittext, fontsize=16)
    plt.ylim(0,400)
    plt.show()
    return z
```

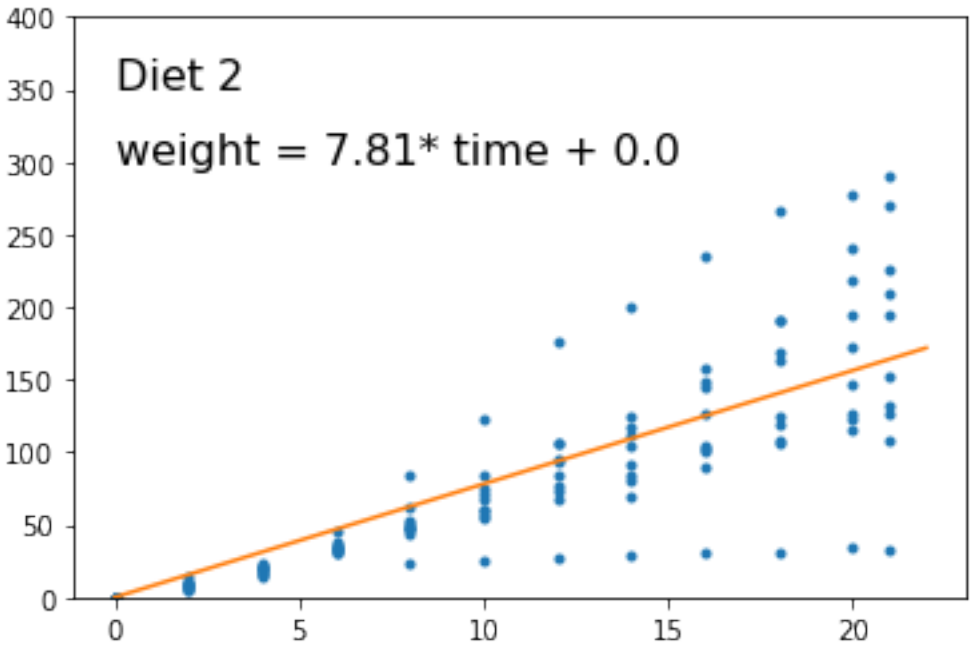
[8]:
```
def plot_all(data):
    uniqueDiets = chickweight.Diet.unique()
    for diet in uniqueDiets:
        df_this_diet = chickweight[:][chickweight.Diet == diet]
```

```
        df_this_diet_values = df_this_diet.values
        X = np.array(df_this_diet_values[:,1], dtype=float)
        Y = np.array(df_this_diet_values[:,0], dtype=float)
        z = get_line_and_plot_nointercept(X, Y, diet)
        print("Regression Line for diet ", diet, ": weight = (",round(z[0],␣
↪2),"* time) + ", round(z[1],2))
```

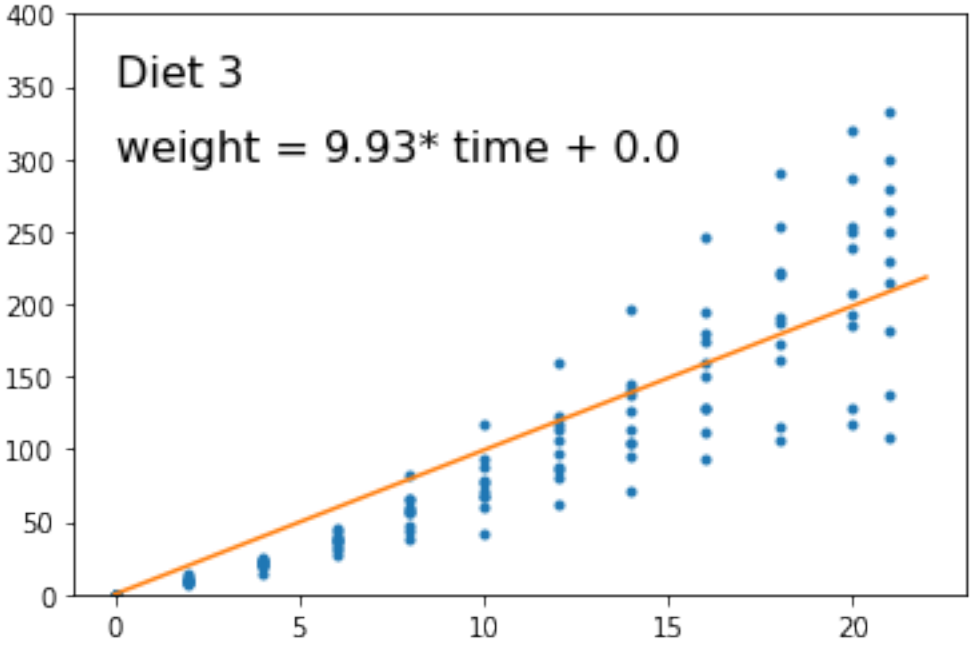
[9]:
```
plot_all(chickweight)
```



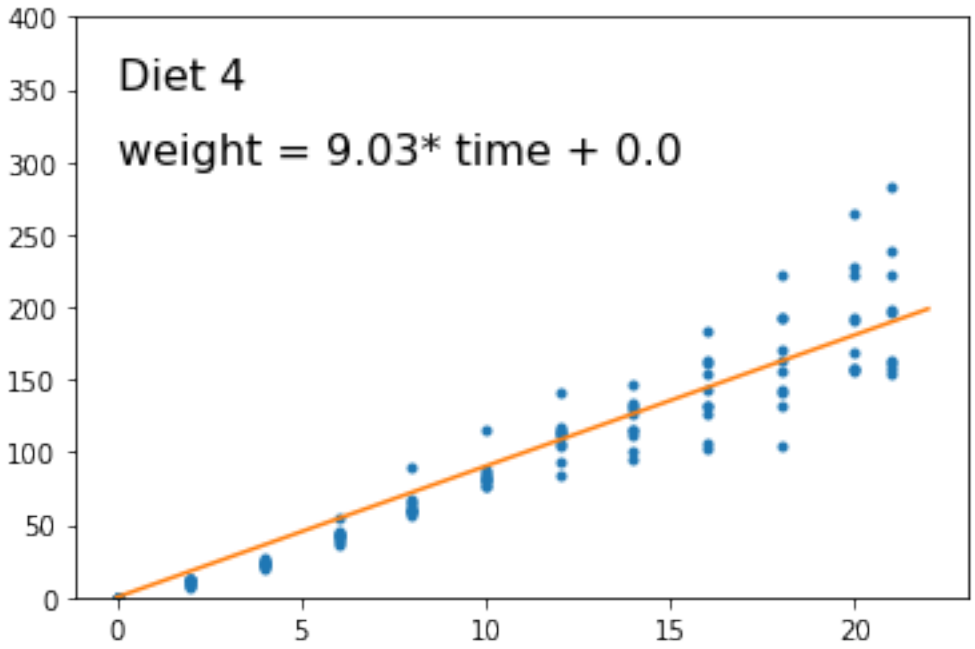

```
Regression Line for diet  1 : weight = ( 6.12 * time) +  0.0
```




```
Regression Line for diet  2 : weight = ( 7.81 * time) +  0.0
```




```
Regression Line for diet  3 : weight = ( 9.93 * time) +  0.0
```




```
Regression Line for diet  4 : weight = ( 9.03 * time) +  0.0
```

[9]:
```
def get_curve_and_plot(X, Y):
    z = np.polyfit(X, Y, 2)
    print(z)
    p = np.poly1d(z)
    xp = np.linspace(0, 22, 100)
    plt.plot(X,Y,'.',xp, p(xp),'-')
    plt.show()
    return z
```

[10]:
```
def plot_all_curves(data):
    uniqueDiets = chickweight.Diet.unique()
    for diet in uniqueDiets:
        df_this_diet = chickweight[:][chickweight.Diet == diet]
        df_this_diet_values = df_this_diet.values
        X = np.array(df_this_diet_values[:,1], dtype=float)
        Y = np.array(df_this_diet_values[:,0], dtype=float)
        z = get_curve_and_plot(X, Y)
```

[10]:
```
#plot_all_curves(chickweight)
```

[ ]:
```

```