Hailey Thanki

In [81]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import seaborn as sns
```

1. Write a Python function (called mymin) that takes a NumPy array as input and returns a tuple containing two elements. The first element is the minimum value of the array and the second element is the number of times the minimum value appears in the array. Please write your own code to find the minimum value. Do not simply use the built-in min() function.

In [11]:

```
def mymin(arr):
  curr_min = arr[0]
  for num in arr:
    if num < curr_min:
      curr_min = num
  return curr_min
```

1. Read in the file /public/bmort/python/numbers.pkl to a NumPy array called numbers. What is the output of mymin(numbers)?

In [10]:

```
arr = np.load('/Users/haileythanki/Desktop/numbers.pkl', allow_pickle=True)
print (mymin(arr))
```

```
40
```

1. Load the file /public/bmort/python/weather.csv into a Pandas data frame. Perform the following tasks and answer the questions below:

   a. The amount of rain recorded in the PREC column is measured in inches. Append a column to the data frame that represents the amount of rainfall in centimeters. Remember to retain the T notation to indicate a trace of rain. Make sure to show all code to append this column to the data frame. Refer to the Pandas documentation as needed.

   b. What was the minimum high (HI) temperature during the month of July?

In [64]:

```
df_weather = pd.read_csv("/Users/haileythanki/Desktop/weather.csv")
df_weather
```

Out[64]:

	MONTH	DAY	HI	LO	AVG	DEP	HDD	CDD	PREC
0	JUN	1	82	48	65	3	0	0	0
1	JUN	2	87	63	75	13	0	10	0
2	JUN	3	83	57	70	7	0	5	0.38

```
     MONTH  DAY  HI  LO  AVG  DEP  HDD  CDD  PREC
 3     JUN    4  73  53   63    0    2    0     0
 4     JUN    5  65  49   57   -6    8    0     0
...    ...  ...  ...  ...  ...  ...  ...  ...   ...
87     AUG   27  78  62   70    2    0    5     T
88     AUG   28  72  55   64   -4    1    0     0
89     AUG   29  77  51   64   -4    1    0     0
90     AUG   30  88  62   75    7    0   10  0.47
91     AUG   31  75  69   72    5    0    7  0.24
```

92 rows × 9 columns

In [66]:

```
df_weather['PREC_cm'] = ((pd.to_numeric(df_weather['PREC'], errors='coerce'))*2.
df_weather
```

Out[66]:

```
     MONTH  DAY  HI  LO  AVG  DEP  HDD  CDD  PREC  PREC_cm
 0     JUN    1  82  48   65    3    0    0     0      0.0
 1     JUN    2  87  63   75   13    0   10     0      0.0
 2     JUN    3  83  57   70    7    0    5  0.38   0.9652
 3     JUN    4  73  53   63    0    2    0     0      0.0
 4     JUN    5  65  49   57   -6    8    0     0      0.0
...    ...  ...  ...  ...  ...  ...  ...  ...   ...      ...
87     AUG   27  78  62   70    2    0    5     T        T
88     AUG   28  72  55   64   -4    1    0     0      0.0
89     AUG   29  77  51   64   -4    1    0     0      0.0
90     AUG   30  88  62   75    7    0   10  0.47   1.1938
91     AUG   31  75  69   72    5    0    7  0.24   0.6096
```

92 rows × 10 columns

In [69]:

```
mymin(df_weather.HI)
```

Out[69]:

```
65
```

1. Using the mtcars data set (located at /public/bmort/python/mtcars.csv) perform the following tasks and answer the questions below:

   a. Using Scikit-Learn's LinearRegression(), construct a model to predict the fuel efficiency of the set of cars (mpg)in the data set as a function of the horsepower (hp) of the engines in the cars.

b. Plot the data points in the data set and your linear model using Pyplot.

c. What is the predicted fuel efficiency of a car that has a horsepower of 130?

In [73]:

```
df_cars = pd.read_csv("/Users/haileythanki/Desktop/mtcars.csv")
df_cars
```

Out[73]:

	car	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
5	Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
7	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
9	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
10	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
17	Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
18	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
19	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
21	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
22	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
23	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
24	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
25	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
26	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
27	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
29	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6

	car	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

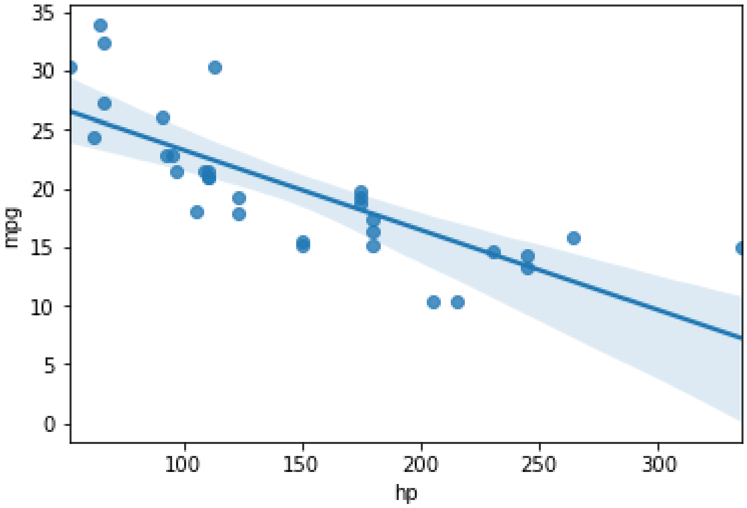
In [89]:

```
X=df_cars.hp
y=df_cars.mpg
X=X.values.reshape(len(X), 1)
y=y.values.reshape(len(y), 1)

reg = LinearRegression()
regmodel_fit = reg.fit(X, y)
```

In [87]:

```
sns.regplot(x='hp',y='mpg',data=df_cars)
plt.show()
```



In [92]:

```
z=np.array(130).reshape(-1,1)
reg.predict(z)
```

Out[92]:

```
array([[21.22918439]])
```

In [ ]: