9th_april_Batch

April 9, 2020

https://www.youtube.com/watch?v=M5ILgNI0iXw&t=272s

[1]:
```
print("Hello world")
```

```
Hello world
```

[3]:
```
name = input("Enter your name : ")
print(name)
```

```
Enter your name : sachin yadav
sachin yadav
```

[5]:
```
num = int(input("Enter a number: "))
if num % 2 == 0:
    print("This is a Even Number")
else:
    print('This is odd Number')
```

```
Enter a number: 34
This is a Even Number
```

[6]:
```
num = int(input("Enter a number: "))
print("Even") if num % 2 == 0 else print("Odd")
```

```
Enter a number: 13
Odd
```

[9]:
```
for var in range(1, 11):
    print("Hello World", var)
```

```
Hello World 1
Hello World 2
Hello World 3
Hello World 4
Hello World 5
Hello World 6
Hello World 7
Hello World 8
Hello World 9
Hello World 10
```

[10]:

```
"sachin " * 10
```

[10]:

```
'sachin sachin sachin sachin sachin sachin sachin sachin sachin sachin '
```

[11]:

```
for var in range(1, 11):
    print("*"*var)
```

```
*
**
***
****
*****
******
*******
********
*********
**********
```

[18]:

```
print("Hello", "world", "how" , "are", "You?", sep='\n')
```

```
Hello
world
how
are
You?
```

[20]:

```
print("Hello", end=' ')
print(" World")
```

```
Hello  World
```

[12]:

```
for var in range(1, 11):
    print(" "*(10-var), "*"*var, sep="")
```

```
         *
        **
       ***
      ****
     *****
    ******
   *******
  ********
 *********
**********
```

[13]:
```
def add(x, y):
    return x + y
```

[14]:
```
x = add(5, 6)
```

[15]:
```
print(x)
```

```
11
```

[25]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

[22]:
```
tip  = sns.load_dataset('tips')
```

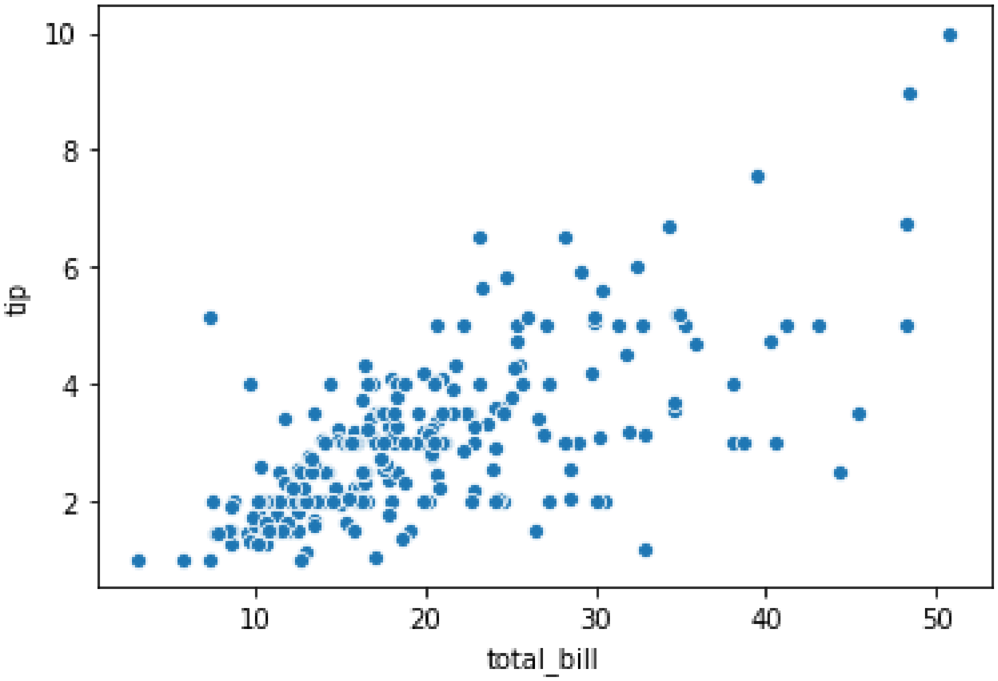
[23]:
```
tip.head()
```

[23]:
```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

[24]:
```
tip.shape
```

[24]:
```
(244, 7)
```

[27]:
```
sns.scatterplot('total_bill', 'tip', data=tip)
plt.show()
```



[37]:
```
url = "https://www.worldometers.info/coronavirus/"
from requests import get
import bs4

page = get(url)
```

[38]:
```
page.status_code
```

[38]:
```
200
```

[ ]:
```

```

[41]:
```
import pandas as pd

data = pd.read_html(page.content)
```

[42]:
```
data
```

[42]:
```
[              Country,Other  TotalCases NewCases  TotalDeaths NewDeaths  \
 0                    World     1538837  +20,877      89975.0    +1,520
 1                      USA      435941   +1,014      14865.0       +77
 2                    Spain      152446   +4,226      15238.0      +446
 3                    Italy      139422      NaN      17669.0       NaN
 4                  Germany      113615     +319       2349.0       NaN
```

```
..                     …         …      …          …       …
208       Papua New Guinea           2      NaN          NaN       NaN
209  Saint Pierre Miquelon           1      NaN          NaN       NaN
210            Timor-Leste           1      NaN          NaN       NaN
211                  China       81865      +63       3335.0        +2
212                 Total:     1538837  +20,877      89975.0    +1,520

     TotalRecovered  ActiveCases  Serious,Critical  Tot Cases/1M pop  \
0          340521.0      1108341           48512.0             197.0
1           22941.0       398135            9281.0            1317.0
2           52165.0        85043            7371.0            3261.0
3           26491.0        95262            3693.0            2306.0
4           46300.0        64966            4895.0            1356.0
..              …          …               …               …
208             NaN            2               NaN               0.2
209             NaN            1               NaN             173.0
210             NaN            1               NaN               0.8
211         77370.0         1160             176.0              57.0
212        340521.0      1108341           48512.0             197.4

     Deaths/1M pop  TotalTests  Tests/ 1M pop
0             11.5         NaN            NaN
1             45.0   2242703.0         6775.0
2            326.0    355000.0         7593.0
3            292.0    807125.0        13349.0
4             28.0   1317887.0        15730.0
..             …         …            …
208            NaN        72.0            8.0
209            NaN         NaN            NaN
210            NaN         NaN            NaN
211            2.0         NaN            NaN
212           11.5         NaN            NaN

[213 rows x 12 columns],
             Country,Other  TotalCases NewCases  TotalDeaths NewDeaths  \
0                    World     1517960  +84,384      88455.0    +6,414
1                    China       81802      +62       3333.0        +2
2                      USA      434927  +31,935      14788.0    +1,940
3                    Spain      148220   +6,278      14792.0      +747
4                    Italy      139422   +3,836      17669.0      +542
..                     …         …      …          …       …
208       Papua New Guinea           2      NaN          NaN       NaN
209            South Sudan           2      NaN          NaN       NaN
210  Saint Pierre Miquelon           1      NaN          NaN       NaN
211            Timor-Leste           1      NaN          NaN       NaN
212                 Total:     1517960  +84,384      88455.0    +6,414
```

```
     TotalRecovered  ActiveCases  Serious,Critical  Tot Cases/1M pop  \
0          330266.0      1099239           48092.0             195.0
1           77279.0         1190             189.0              57.0
2           22891.0       397248            9279.0            1314.0
3           48021.0        85407            7069.0            3170.0
4           26491.0        95262            3693.0            2306.0
..              …           …                …                …
208             NaN            2               NaN               0.2
209             NaN            2               NaN               0.2
210             NaN            1               NaN             173.0
211             NaN            1               NaN               0.8
212        330266.0      1099239           48092.0             194.7

     Deaths/1M pop  TotalTests  Tests/ 1M pop
0             11.3         NaN            NaN
1              2.0         NaN            NaN
2             45.0   2209041.0         6674.0
3            316.0    355000.0         7593.0
4            292.0    807125.0        13349.0
..             …          …             …
208            NaN        72.0            8.0
209            NaN         NaN            NaN
210            NaN         NaN            NaN
211            NaN         NaN            NaN
212           11.3         NaN            NaN

[213 rows x 12 columns]]
```

[44]:
```python
type(data[0])
```

[44]:
```
pandas.core.frame.DataFrame
```

[45]:
```python
import pandas as pd

data = pd.read_html(page.content)
df = data[0]
```

[48]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 213 entries, 0 to 212
Data columns (total 12 columns):
Country,Other       213 non-null object
TotalCases          213 non-null int64
NewCases            94 non-null object
TotalDeaths         162 non-null float64
NewDeaths           55 non-null object
```

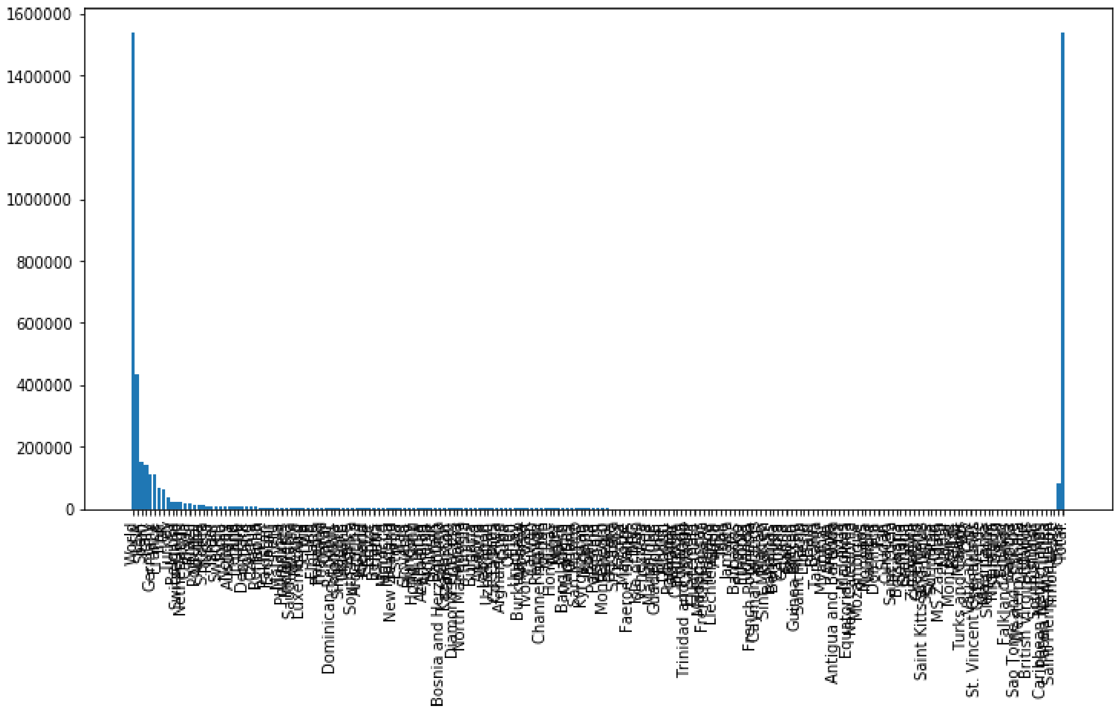
```
TotalRecovered      182 non-null float64
ActiveCases         213 non-null int64
Serious,Critical    128 non-null float64
Tot Cases/1M pop    211 non-null float64
Deaths/1M pop       160 non-null float64
TotalTests          150 non-null float64
Tests/ 1M pop       150 non-null float64
dtypes: float64(7), int64(2), object(3)
memory usage: 20.1+ KB
```

ml + signal processing –> computer vision

```
1. programming (intermediate)
2. maths (stats, probability, hypothesis testing, algebra, vector & matrix,
    differential equations)
3. Algortithms
4. Case Studies
5. Data Engineering
6. Data Analysis (trends of past data) # patterns, hypothesis
   Data Visulation
7. Data Analytics (prediction future)
8. Machin Learning
9. Deep Learning
10. machine Vision
11. Artifical Intelligence
```

[50]:

```python
plt.figure(figsize=(12, 6))
plt.bar(df['Country,Other'], df['TotalCases'])
plt.xticks(rotation=90)
plt.show()
```



```
[51]: import pandas as pd

      data = pd.read_html(page.content)
      df = data[0]
      cd = df[['Country,Other', 'TotalCases']]
```

```
[53]: cd.shape
```

```
[53]: (213, 2)
```

```
[56]: len(cd['TotalCases'] > 10000)
```

```
[56]: 213
```

```
[61]: sum(cd['TotalCases'] > 5000)
```

```
[61]: 31
```

```
[63]: import pandas as pd

      data = pd.read_html(page.content)
      df = data[0]
      cd = df[['Country,Other', 'TotalCases']]
```

```
d = cd[cd['TotalCases'] > 5000]
```

[66]:

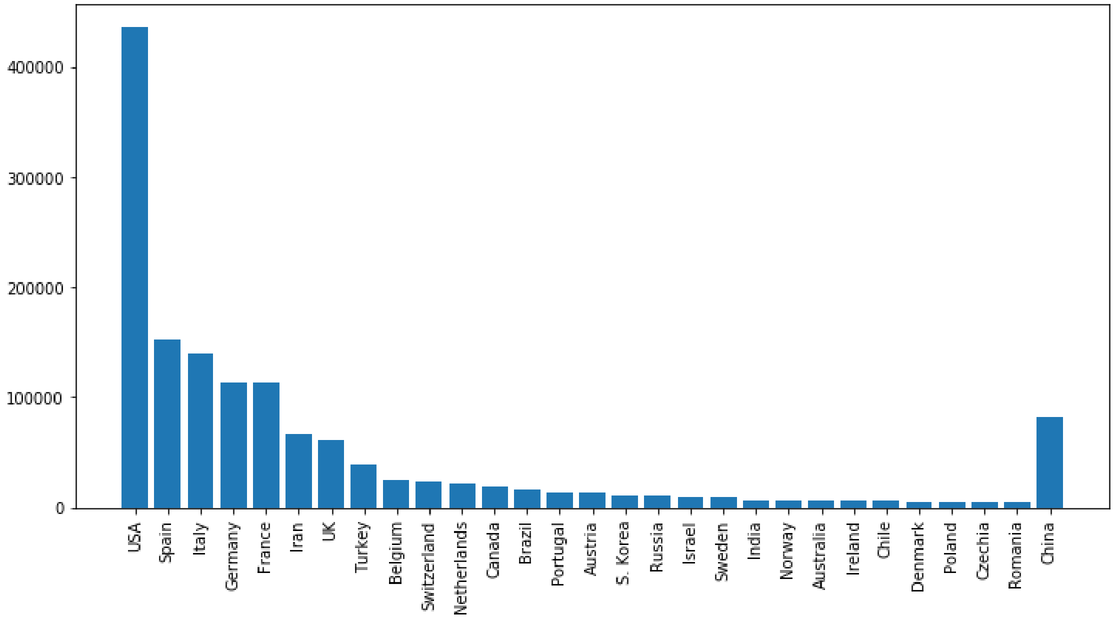
```
d = d[1:-1]
```

[ ]:

```

```

[68]:

```
import pandas as pd

data = pd.read_html(page.content)
df = data[0]
cd = df[['Country,Other', 'TotalCases']]

d = cd[cd['TotalCases'] > 5000]
plt.figure(figsize=(12, 6))
plt.bar(d['Country,Other'], d['TotalCases'])
plt.xticks(rotation=90)
plt.show()
```



[70]:

```
cd[cd['Country,Other'] == 'India']
```

[70]:

```
   Country,Other  TotalCases
20         India        6588
```

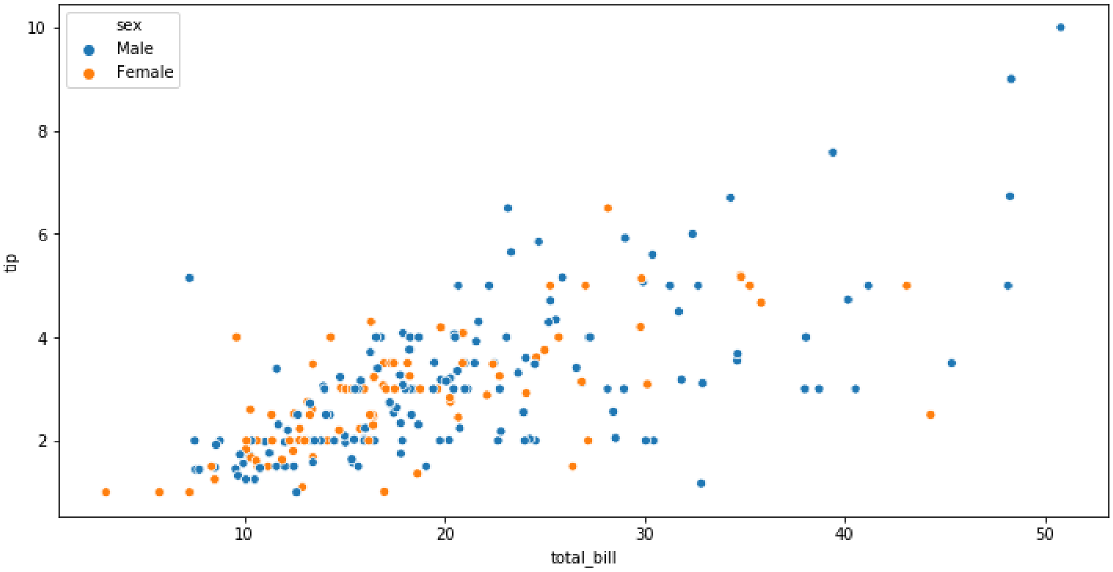
[72]:

```
tip.head()
```

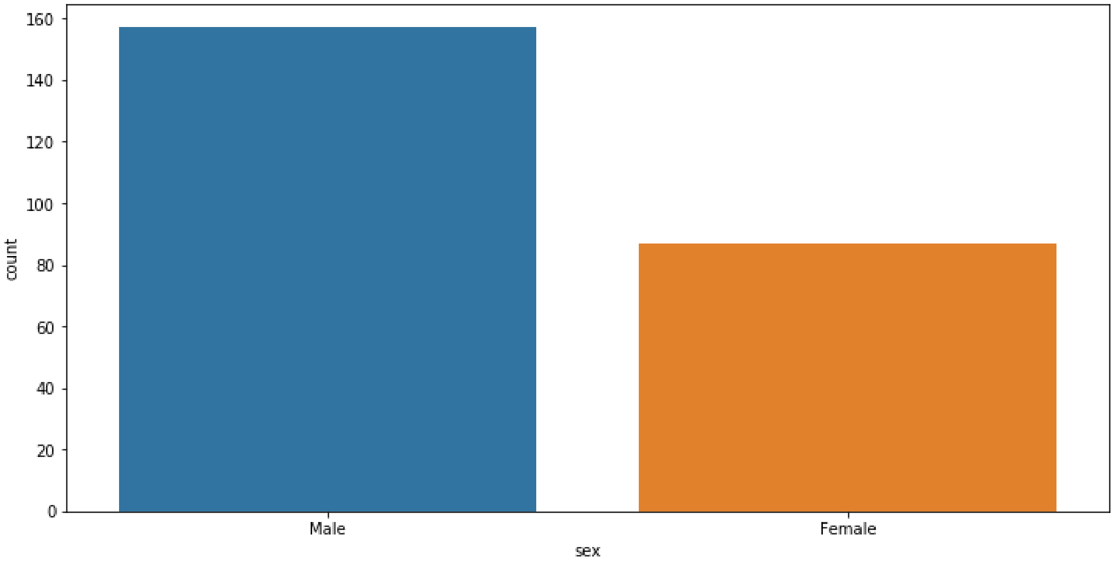
[72]:
```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

[76]:
```
plt.rcParams['figure.figsize'] = 12,6
```

[77]:
```
sns.scatterplot('total_bill', 'tip',data=tip, hue='sex')
plt.show()
```

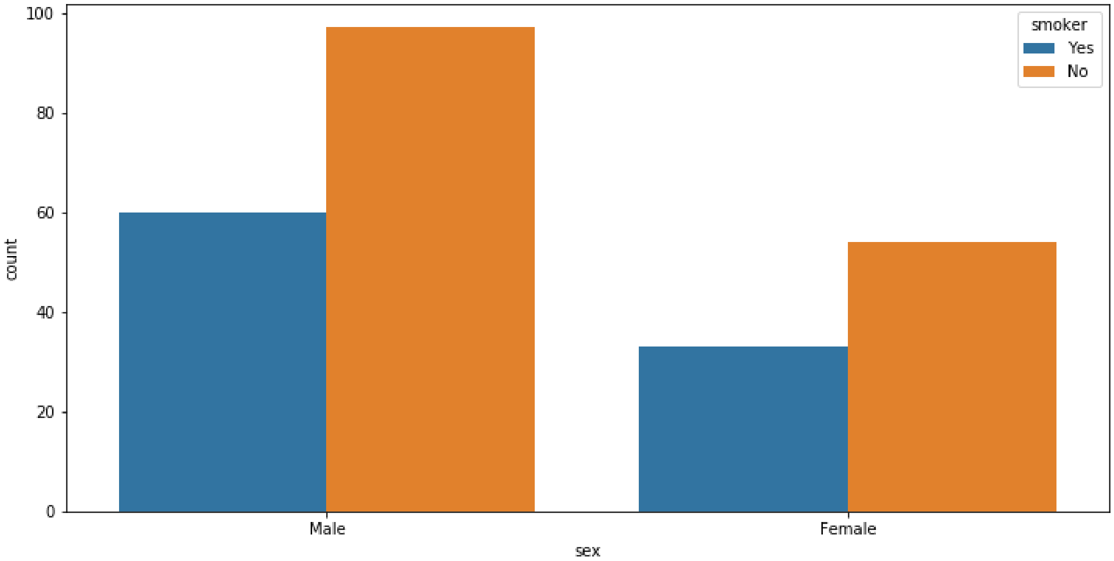


[78]:
```
sns.countplot('sex', data=tip)
plt.show()
```



[81]:

```
sns.countplot('sex', data=tip, hue='smoker')
plt.savefig('myplot.png')
plt.show()
```



[80]:

```
import seaborn as sns
```

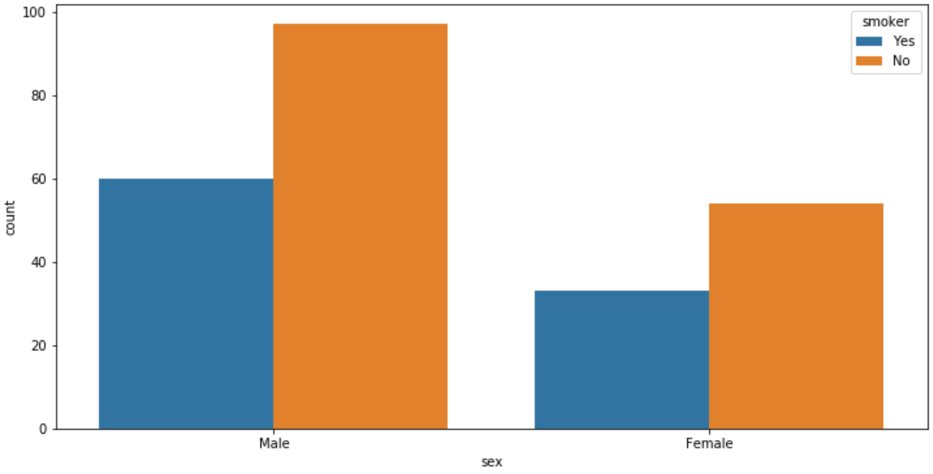
[82]:

```
from PIL import Image
```

[84]:

```
Image.open('myplot.png')
```

[84]:



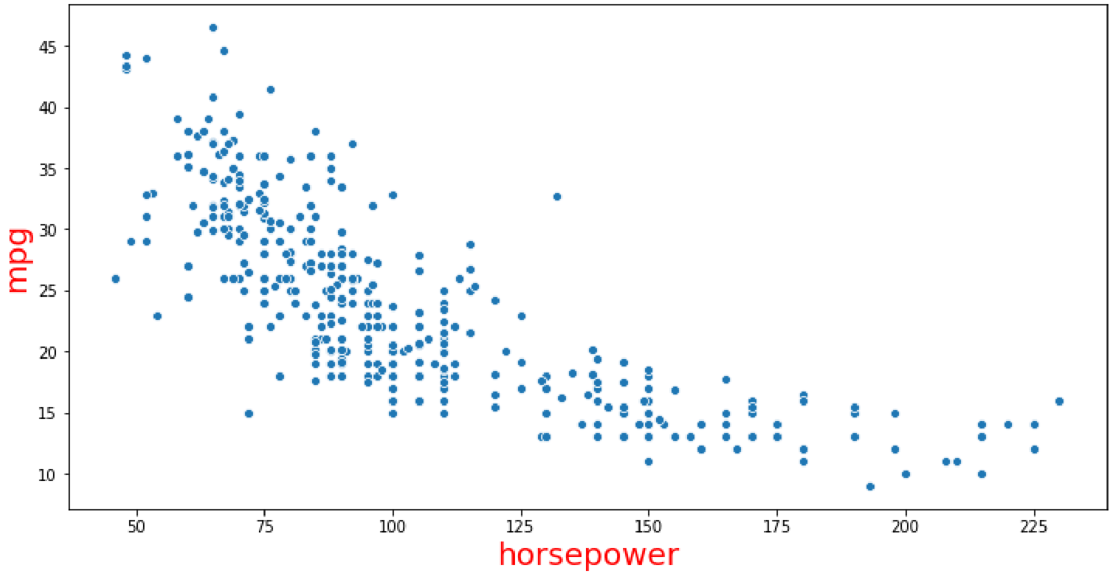
[86]:
```
mpg = sns.load_dataset('mpg')
```

[87]:
```
mpg.head()
```

[87]:
```
    mpg  cylinders  displacement  horsepower  weight  acceleration  \
0  18.0          8         307.0       130.0    3504          12.0
1  15.0          8         350.0       165.0    3693          11.5
2  18.0          8         318.0       150.0    3436          11.0
3  16.0          8         304.0       150.0    3433          12.0
4  17.0          8         302.0       140.0    3449          10.5

   model_year origin                       name
0          70    usa  chevrolet chevelle malibu
1          70    usa          buick skylark 320
2          70    usa         plymouth satellite
3          70    usa              amc rebel sst
4          70    usa                ford torino
```

[96]:
```
plt.rcParams['axes.labelsize'] = 20
plt.rcParams['axes.labelcolor'] = 'red'
```

[108]:
```
#y = -0.45 * mpg['horsepower'] + 20
sns.scatterplot('horsepower', 'mpg', data=mpg)
#plt.plot(mpg['horsepower'], y, 'r')
plt.show()
```



```
horse power -->     Model --> mpg
```

Linear Regression

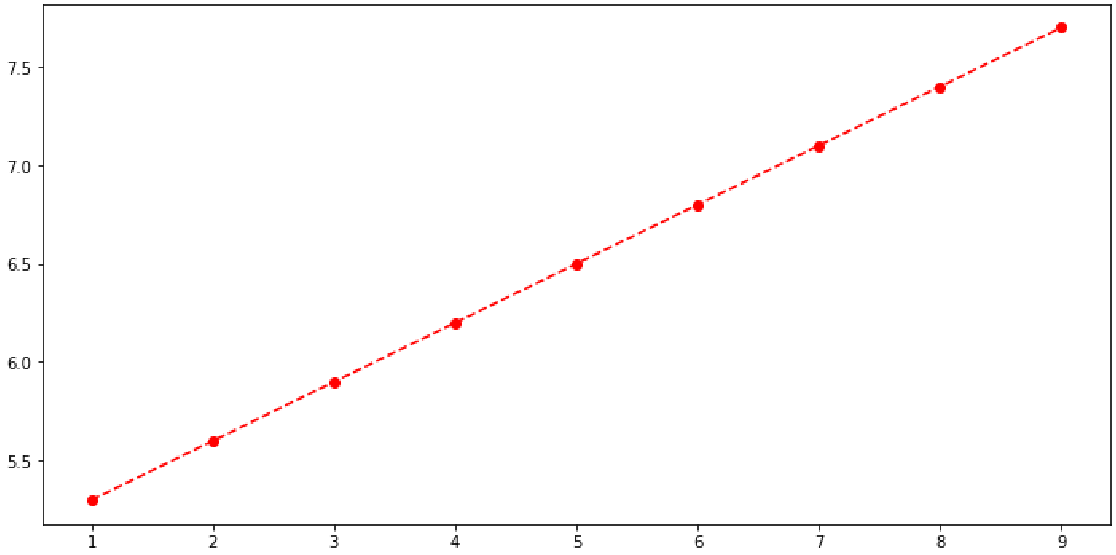
[94]:

```
#y = mx + c
import numpy as np
x = np.array([ 1, 2, 3, 4, 5,6, 7, 8, 9])
m = 0.3 # slope, coeffcient
c = 5 # intercept
y = m*x + c
print(y)
```

```
[5.3 5.6 5.9 6.2 6.5 6.8 7.1 7.4 7.7]
```

[95]:

```
plt.plot(x,y,'ro--')
plt.show()
```



[109]:
```
from sklearn.linear_model import LinearRegression
```

[110]:
```
model = LinearRegression()
```

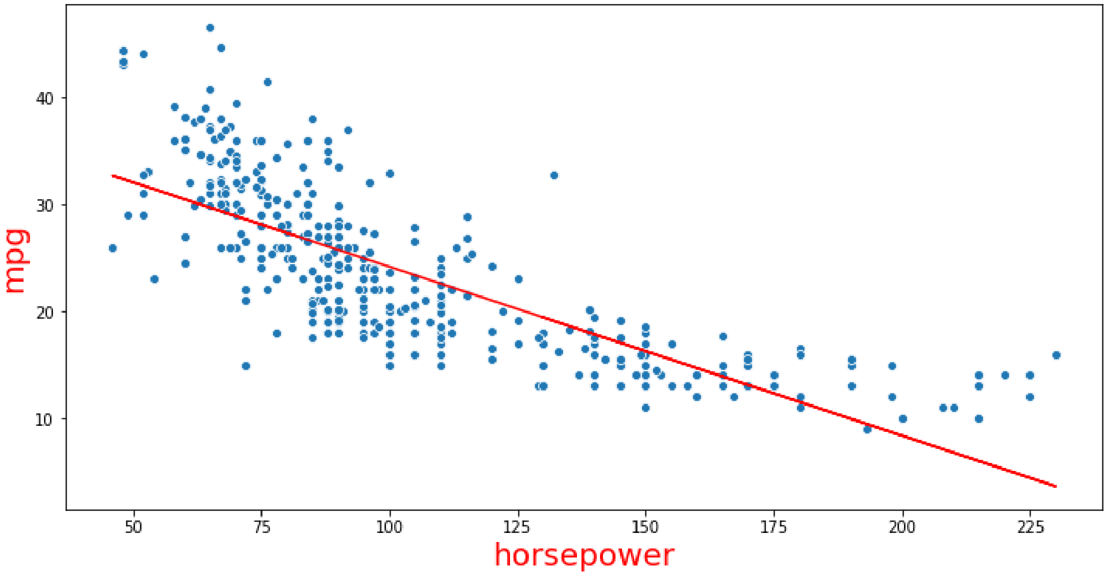
[113]:
```
mpg.dropna(inplace=True)
```

[114]:
```
model.fit(mpg[['horsepower']], mpg['mpg']) # learning phase
```

[114]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

[115]:
```
y_pred = model.predict(mpg[['horsepower']])
```

[116]:
```
#y = -0.45 * mpg['horsepower'] + 20
sns.scatterplot('horsepower', 'mpg', data=mpg)
plt.plot(mpg['horsepower'], y_pred, 'r')
plt.show()
```



[117]:
```
from sklearn.metrics import r2_score
```

[118]:
```
r2_score(mpg['mpg'], y_pred)
```

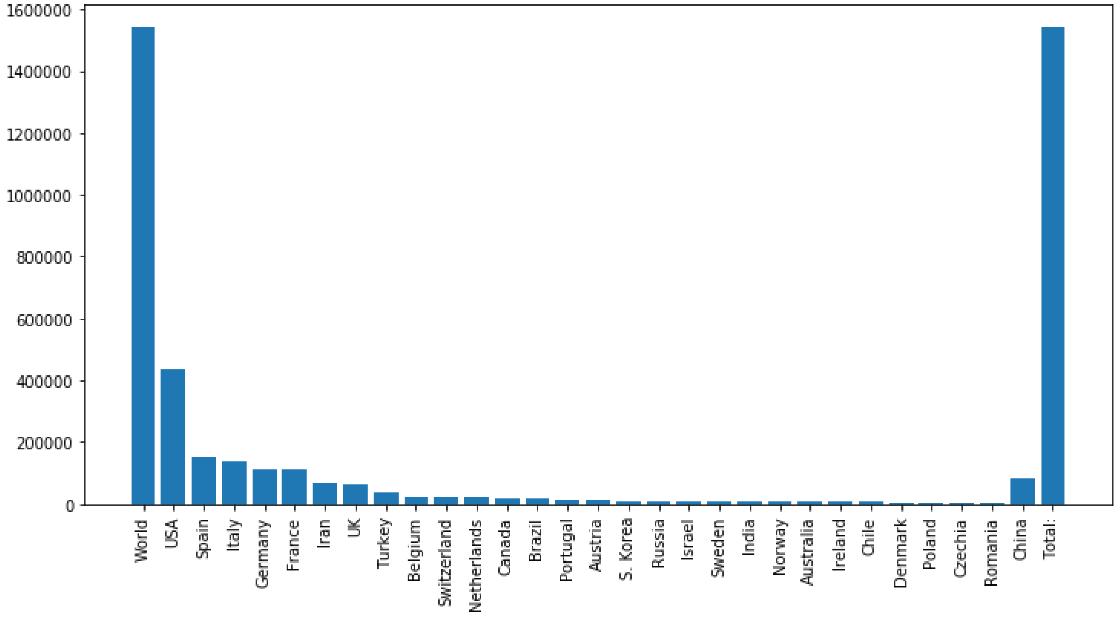
[118]:
```
0.6059482578894348
```

https://github.com/sachinyadav3496

your code will be available at https://github.com/sachinyadav3496/kiit

sachinyadav3496@gmail.com

[120]:
```
url = "https://www.worldometers.info/coronavirus/"
from requests import get
import bs4

page = get(url)
import pandas as pd

data = pd.read_html(page.content)
df = data[0]
cd = df[['Country,Other', 'TotalCases']]

d = cd[cd['TotalCases'] > 5000]
plt.figure(figsize=(12, 6))
plt.bar(d['Country,Other'], d['TotalCases'])
plt.xticks(rotation=90)
plt.show()
```


1600000
1400000
1200000
1000000
800000
600000
400000
200000
0
World
USA
Spain
Italy
Germany
France
Iran
UK
Turkey
Belgium
Switzerland
Netherlands
Canada
Brazil
Portugal
Austria
S. Korea
Russia
Israel
Sweden
India
Norway
Australia
Ireland
Chile
Denmark
Poland
Czechia
Romania
China
Total:

```
[ ]:
```