```
import io
import sys
import pandas as pd
from google.colab import files


uploaded = files.upload()
```

Choose Files 51.Old Faith…er Data.csv

- **51.Old Faithful Geyser Data.csv**(application/vnd.ms-excel) - 4284 bytes, last modified: 5/11/2021 - 100% done

Saving 51.Old Faithful Geyser Data.csv to 51.Old Faithful Geyser Data.csv

```
df2 = pd.read_csv(io.BytesIO(uploaded['51.Old Faithful Geyser Data.csv']))



print("There are",len(df2.columns),"columns:")
```

```
There are 3 columns:
```

```
for x in df2.columns:
 sys.stdout.write(str(x)+", ")
```

```
Unnamed: 0, waiting, duration,
```

```
print("\n************")
print("Dataset Info:")
print(df2.info())
print("\n************")
print(df2)
print("\n************")
```

```
************
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  299 non-null    int64  
 1   waiting     299 non-null    int64  
 2   duration    299 non-null    float64
dtypes: float64(1), int64(2)
memory usage: 7.1 KB
None

************
     Unnamed: 0  waiting  duration
```

```
0             1       80  4.016667
1             2       71  2.150000
2             3       57  4.000000
3             4       80  4.000000
4             5       75  4.000000
..          ...      ...       ...
294         295       52  4.083333
295         296       85  2.066667
296         297       58  4.000000
297         298       88  4.000000
298         299       79  2.000000

[299 rows x 3 columns]

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

```
waiting=df2.iloc[:,1:2].values
print(waiting)
```

```
[ 76]
[ 59]
[ 80]
[ 89]
[ 45]
[ 93]
[ 72]
[ 71]
[ 54]
[ 79]
[ 74]
[ 65]
[ 78]
[ 57]
[ 87]
[ 72]
[ 84]
[ 47]
[ 84]
[ 57]
[ 87]
[ 68]
[ 86]
[ 75]
[ 73]
[ 53]
[ 82]

[ 93]
[ 77]
[ 54]
[ 96]
[ 48]
[ 89]
[ 63]
[ 84]
[ 76]
[ 62]
```

```
[ 82]
[ 83]
[ 50]
[ 85]
[ 78]
[ 78]
[ 81]
[ 78]
[ 76]
[ 74]
[ 81]
[ 66]
[ 84]
[ 48]
[ 93]
[ 47]
[ 87]
[ 51]
[ 78]
[ 54]
[ 87]
[ 52]
[ 85]
[ 58]
```

```
duration=df2.iloc[:,2].values
print(duration)
```

```
[4.0166667 2.15      4.        4.        4.        2.        4.3833333
 4.2833333 2.0333333 4.8333333 1.8333333 5.45      1.6166667 4.8666667
 4.3833333 1.7666667 4.6666667 2.        4.7333333 4.2166667 1.9
 4.9666667 2.        4.        2.        4.        2.8333333 4.5
 4.0666667 3.7166667 3.5166667 4.4666667 2.2166667 4.8833333 2.6
 4.15      2.2       4.7666667 1.8333333 4.6       2.2666667 4.1333333
 2.        4.        2.        4.        1.8833333 4.2666667 2.0833333
 4.4666667 2.5       4.        1.7666667 4.3333333 2.1833333 4.4833333
 3.8833333 3.3333333 3.7333333 4.        1.95      5.2666667 2.
 4.        2.        4.        2.        4.        3.5333333 2.1666667
 4.5       2.0166667 4.15      4.2       4.3333333 1.9333333 4.65
 3.8166667 4.0333333 4.1666667 4.6666667 1.8166667 4.        3.
 4.        2.        4.45      2.05      4.25      1.9166667 4.6666667
 1.7333333 4.3833333 1.7666667 4.6       1.8666667 4.45      1.6333333
 5.0333333 1.8166667 5.1       1.6333333 4.2833333 2.        4.
 2.        4.5333333 2.        4.        2.9333333 4.7333333 3.9
 1.95      4.1166667 1.8       4.6666667 1.8333333 4.7       2.1166667
 4.7833333 1.8166667 4.1       4.65      4.        2.        4.
 4.        4.2166667 4.1333333 3.9333333 3.75      4.4166667 2.4666667
 4.1666667 3.8       4.3166667 3.8666667 4.6833333 1.7       4.9666667
 4.2666667 4.5833333 4.        4.        4.        4.        1.9833333
 4.6       0.8333333 4.9166667 1.7333333 4.5833333 1.7       4.75
 1.8333333 4.5       1.8666667 4.45      4.45      4.        4.8
 4.        4.        2.        4.        1.9333333 4.5833333 2.
 3.7       2.8666667 4.8333333 3.45      4.3833333 1.8       4.4
 2.4833333 4.5166667 2.1       4.35      4.3666667 1.7833333 4.9166667
 1.8166667 4.        4.        4.        3.8666667 1.85      4.7
 2.0166667 4.4666667 1.8666667 4.1666667 1.9       4.25      3.25
```

```
      4.2166667 1.8833333 4.9833333 1.85      4.        1.9666667 4.7666667
      4.        2.        4.        4.        2.3833333 4.4166667 4.2166667
      4.3666667 2.        4.45      1.75      4.5       1.6166667 4.7
      2.5666667 3.7       4.2333333 1.9333333 4.35      4.        4.
      4.        4.2166667 4.        4.1333333 1.8833333 4.4666667 1.95
      4.2166667 1.7166667 4.45      4.25      3.9666667 4.3833333 1.9666667
      4.45      4.2666667 1.9166667 4.4166667 3.        4.        2.
      4.        3.2833333 1.8333333 4.6166667 1.8333333 4.6166667 4.6
      4.25      1.9333333 4.9833333 1.9666667 4.3       4.2       4.5333333
      4.4       4.6166667 2.        4.        4.        3.9166667 2.
      4.5       1.8       4.        2.75      4.7333333 3.9666667 1.95
      4.9666667 1.85      4.8       4.        4.        4.        4.
      4.        4.        4.        2.        4.        1.9333333 4.3333333
      1.6666667 4.7666667 1.95      4.6833333 1.9333333 4.4166667 2.1333333
      4.0833333 2.0666667 4.        4.        2.       ]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(waiting, duration, test_size = 0.2, rando
```

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

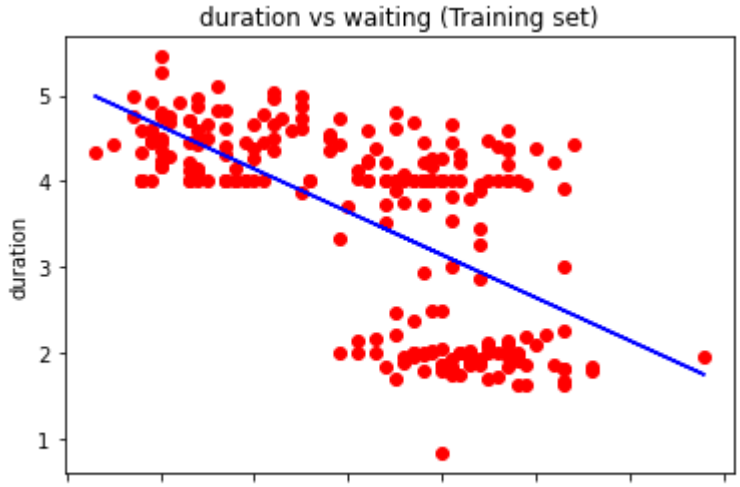
```
y_pred = regressor.predict(X_test)
y_pred_train = regressor.predict(X_train)
```

```
# print r_square_score
from sklearn.metrics import r2_score
```

```
print("R_square score: ", r2_score(y_test,y_pred))
```

```
     R_square score:  0.3975819834361388
```

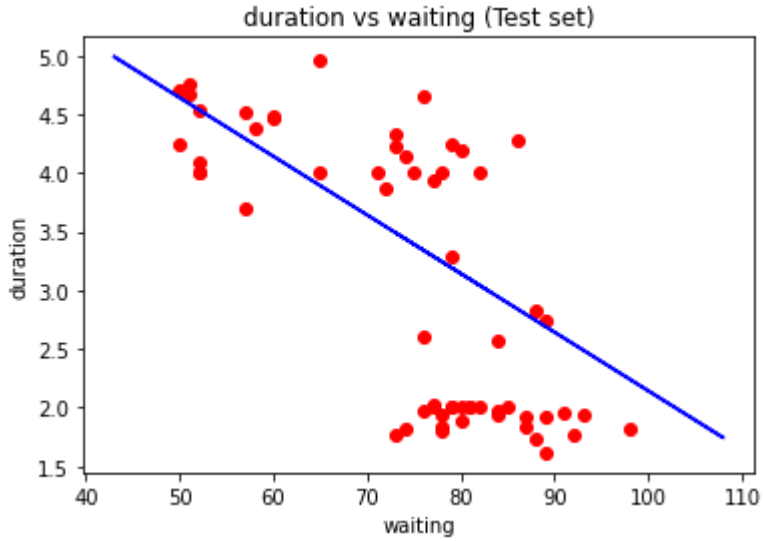
```
# Visualising the Test set results
import matplotlib.pyplot as plt
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('duration vs waiting (Training set)')
plt.xlabel('waiting')
plt.ylabel('duration')
plt.show()
```

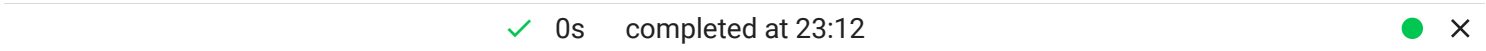



```
X_future_waiting = [[63],[97]]
print ("Duration of new passengers joining would be :", regressor.predict(X_future_waiting))
```

```
Duration of new passengers joining would be : [3.99135718 2.29242954]
```

```
# Visualising the Test set results
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('duration vs waiting (Test set)')
plt.xlabel('waiting')
plt.ylabel('duration')
plt.show()
```




0s
completed at 23:12