```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!cp "/content/drive/MyDrive/Colab Notebooks/steps_vs_cal.csv" "/content/"
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

```

```
dataset = pd.read_csv('steps_vs_cal.csv')
dataset = dataset[['step_count', 'calories_burned']]
dataset
# df.drop('date')
```

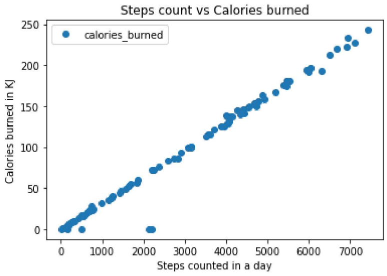
1 to 25 of 96 entries Filter

index	step_count	calories_burned
0	5464	181
1	6041	197
2	25	0
3	5461	174
4	6915	223
5	4545	149
6	4340	140
7	1230	38
8	61	1
9	1258	40
10	3148	101
11	4687	152
12	4732	150
13	3519	113
14	1580	49
15	2822	86
16	181	6
17	3158	99
18	4383	143
19	3881	125
20	4037	129
21	202	6
22	292	9
23	330	10
24	2209	72

Show 25 per page 1 2 3 4

Like what you see? Visit the data table notebook to learn more about interactive tables.

```
# step_count = df['step_count']
# calories_burned = df['calories_burned']

# step_count = step_count.to_numpy()
# calories_burned = calories_burned.to_numpy()

dataset.plot(x='step_count', y='calories_burned', style='o')
plt.title('Steps count vs Calories burned')
plt.xlabel('Steps counted in a day')
plt.ylabel('Calories burned in KJ')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values
plt.show()

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

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


```



```
LinearRegression()
```

```
print('intercept: ', regressor.intercept_)
print('coeficient: ', regressor.coef_)
print('pearson correlation: ', np.corrcoef(dataset['step_count'], dataset['calories_burned']))
```

```
intercept:  -4.501301128591351
coeficient:  [0.03321682]
pearson correlation:  [[1.        0.9892597]
 [0.9892597 1.       ]]
```

From the above graph we can recongnise a few outliers, but majority of the data follows a simple trend of a linear relation. We can simply say, there is a positive relationship betweeen steps walked and calories burned. And judging from the linear regression coeficient we can see that it is 0.9892, it shows that the linear model is workable with the current dataset.

```
y_pred = regressor.predict(X_test)
df = pd.DataFrame({ 'Actual': y_test, 'Predicted': y_pred})
df

```

1 to 20 of 20 entries Filter

index	Actual	Predicted
0	141	142.81528159788144
1	99	97.34145933124304
2	0	-3.6708807073598244
3	116	115.34497406354254
4	125	126.40617407434645
5	180	179.42021376576713
6	6	1.5109427211249002
7	21	17.122846640277594
8	220	217.25416815707547
9	25	22.038935533968232
10	35	34.12985686709926
11	47	43.563432852289395
12	39	36.38860041284901
13	113	112.3886773639583
14	38	36.35538359599975
15	17	13.402563153160358
16	9	5.198009391392876
17	72	68.87464729142631
18	131	130.62470981420262
19	1	-2.4750753007864263

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

The following table shows calories burned in kJ when a certain amount of steps are taken, in predicted values and actual values.

```
y_pred = regressor.predict(dataset[['step_count']].to_numpy()) #to predict other amount of steps walked, please replace what is in the initial ()s !
df = pd.DataFrame({ 'Steps count': dataset['step_count'].to_numpy(),'Actual': dataset['calories_burned'].to_numpy(), 'Predicted': y_pred})
df
```

1 to 25 of 96 entries Filter

index	Steps count	Actual	Predicted
0	5464	181	176.99538613577107
1	6041	197	196.1614894577947
2	25	0	-3.6708807073598244
3	5461	174	176.89573568522326
4	6915	223	225.19298738404882
5	4545	149	146.46913145130014
6	4340	140	139.65968399720163
7	1230	38	36.35538359599975
8	61	1	-2.4750753007864263
9	1258	40	37.285454467779054
10	3148	101	100.06523831288244
11	4687	152	151.18591944389522
12	4732	150	152.68067620211195
13	3519	113	112.3886773639583
14	1580	49	47.981269493241115
15	2822	86	89.23655602002334
16	181	6	1.5109427211249002
17	3158	99	100.39740648137506
18	4383	143	141.08800712171984
19	3881	125	124.41316506339079
20	4037	129	129.59498849187554
21	202	6	2.2084958749593824
22	292	9	5.198009391392876
23	330	10	6.460248431664796
24	2209	72	68.87464729142631

Show 25 per page 1 2 3 4

Like what you see? Visit the data table notebook to learn more about interactive tables.

✓ 0s completed at 5:16 PM