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


uploaded_files = files.upload()
```

```
df = pd.read_csv(io.BytesIO(uploaded_files['Book1.csv']))
```

```
df.head()
```

```
df.tail()
```

used data frame

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   date       30 non-null     object
 1   paperback  30 non-null     int64
```

```
 2   hardcover  30 non-null     int64
dtypes: int64(2), object(1)
memory usage: 848.0+ bytes
```

```
df.describe()
```

Statistics of data

```
df.shape
```

```
(30, 3)
```

```
df_new = pd.read_csv(
    io.BytesIO(uploaded_files['Book1.csv']),
    parse_dates=['date'],
).drop('paperback', axis=1)
```

```
df_new.head()
```

```
import numpy as np
```

```
df_new['Time'] = np.arange(len(df_new.index))
```

```
df_new.head()
```

date information is converted into time first index is always 0

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use("seaborn-whitegrid")
plt.rc(
    "figure",
    autolayout=True,
    figsize=(11, 4),
    titlesize=18,
    titleweight='bold',
)
plt.rc(
    "axes",
    labelweight="bold",
    labelsize="large",
    titleweight="bold",
    titlesize=16,
    titlepad=10,
)

fig, ax = plt.subplots()
ax.plot('Time', 'hardcover', data=df_new, color='0.75')
ax = sns.regplot(x='Time', y='hardcover', data=df_new, ci=None, scatter_kws=dict(color='0.
ax.set_title('Time Plot of Hardcover Sales');
```

prediction is blue line

Accuracy :: over fitting/under fitting not recommended

Recommended :: 80-85%

Accuracy:: increase number of data

```
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

Linear regression scikit learn is website to learn

What to model against what? x values are features Basic idea is in what time what was the sales Forecast data based on time

```
X = df_new.loc[:, ['Time']]  # features
```

```
y = df_new.loc[:, 'hardcover']  # target
```

```
model = LinearRegression()
model.fit(X, y)
```

```
     LinearRegression()
```

```
## predict hardcover sales
## x will never get hardcover
## in what time how much was sold is found out
## based on x values

y_pred = pd.Series(model.predict(X), index=X.index)
```

```
print(y_pred)
```

```
     0     134.045161
     1     138.214461
     2     142.383760
     3     146.553059
     4     150.722358
     5     154.891657
     6     159.060957
     7     163.230256
```

```
8     167.399555
9     171.568854
10    175.738154
11    179.907453
12    184.076752
13    188.246051
14    192.415350
15    196.584650
16    200.753949
17    204.923248
18    209.092547
19    213.261846
20    217.431146
21    221.600445
22    225.769744
23    229.939043
24    234.108343
25    238.277642
26    242.446941
27    246.616240
28    250.785539
29    254.954839
dtype: float64
```

```
print(metrics.r2_score(y,y_pred))
```

```
0.4945651126168752
```

Predict accuracy of predictions

Linear regression y = ax+b a,b is calculted by algo on variable coefficient not on variable intercept

```
print(model.coef_,model.intercept_)
```

```
[4.16929922] 134.0451612903226
```

Pen Paper Demonstration

1) Prepared data -->Csv

2) Manipulate data --> add time column

3) model fit

4) predictions

What happened 1) linear regression line

y = 3.7x + 5 y

for particular data use x = 0-31

More the data more the accuracy more the variations

Use kaggel to manipulate data

Increase the number of data it will increase number of accurate

Double-click (or enter) to edit