VAC ASSIGNMENT Narshimha Reddy

```
              (RA1911042020043)
```

Aim: To write program for peridicting student grade using linear regression

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

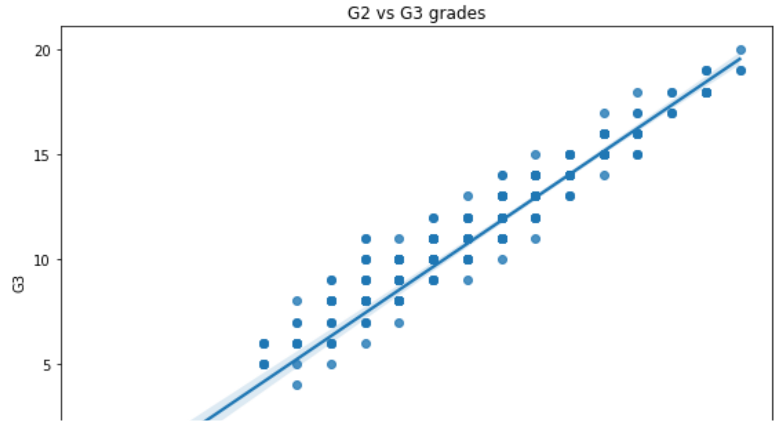
```
maths = pd.read_csv('/content/student-mat.csv')
print(maths.info())
maths.head()
```

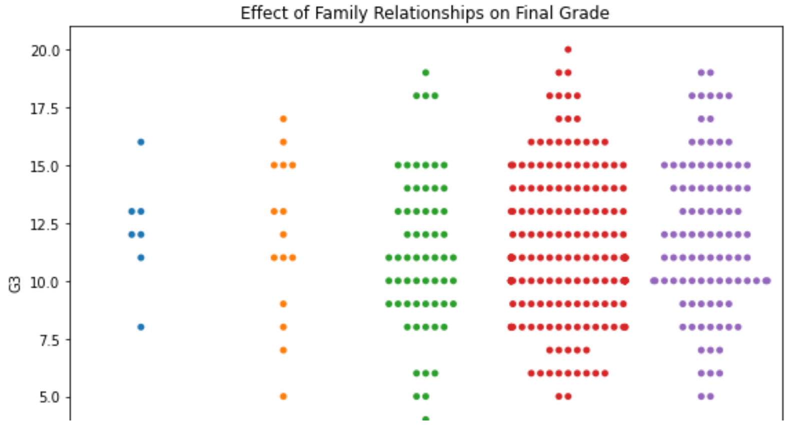
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   school      395 non-null    object
 1   sex         395 non-null    object
 2   age         395 non-null    int64 
 3   address     395 non-null    object
 4   famsize     395 non-null    object
 5   Pstatus     395 non-null    object
 6   Medu        395 non-null    int64 
 7   Fedu        395 non-null    int64 
 8   Mjob        395 non-null    object
 9   Fjob        395 non-null    object
 10  reason      395 non-null    object
 11  guardian    395 non-null    object
 12  traveltime  395 non-null    int64 
 13  studytime   395 non-null    int64 
 14  failures    395 non-null    int64 
 15  schoolsup   395 non-null    object
 16  famsup      395 non-null    object
 17  paid        395 non-null    object
 18  activities  395 non-null    object
 19  nursery     395 non-null    object
 20  higher      395 non-null    object
 21  internet    395 non-null    object
 22  romantic    395 non-null    object
 23  famrel      395 non-null    int64 
 24  freetime    395 non-null    int64 
 25  goout       395 non-null    int64 
 26  Dalc        395 non-null    int64 
 27  Walc        395 non-null    int64 
 28  health      395 non-null    int64 
 29  absences    395 non-null    int64 
```

```
fig, axes = plt.subplots(2, 2, figsize=(16,12))

sns.regplot('G2', 'G3', data=maths, ax=axes[0, 0]).set_title('G2 vs G3 grades')
sns.swarmplot('failures', 'G3', data=maths, ax=axes[1, 0]).set_title('Effect of Failures on Final Grade')
sns.swarmplot('famrel', 'G3', data=maths, ax=axes[0, 1]).set_title('Effect of Family Relationships on Final Grade')
sns.swarmplot('studytime', 'G3', data=maths, ax=axes[1, 1]).set_title('Effect of Studytime on Final Grade')
plt.tight_layout()
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following var
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following var
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarning: 37.8% of the points ca
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following var
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarning: 22.1% of the points ca
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following var
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarning: 17.2% of the points ca
  warnings.warn(msg, UserWarning)
```





```
maths = maths.select_dtypes('int64')
maths = maths[['famrel', 'studytime', 'failures', 'absences', 'G1', 'G2', 'G3']]
print(maths.info())

# set our prediction of a students final score (G3)
predict = 'G3'

# split-up X & y and make sure that they are np array's
# sklearn needs numpy array's as inputs
X = np.array(maths.drop(predict, axis=1))
y = np.array(maths[predict])
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 395 entries, 0 to 394
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   famrel     395 non-null    int64
 1   studytime  395 non-null    int64
 2   failures   395 non-null    int64
 3   absences   395 non-null    int64
 4   G1         395 non-null    int64
 5   G2         395 non-null    int64
 6   G3         395 non-null    int64
dtypes: int64(7)
memory usage: 21.7 KB
None
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.8, random_state=42
)
```

```
linear = LinearRegression()
linear.fit(X_train, y_train)
```

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

```
print("The R^2 is: ", linear.score(X_test, y_test))
coeff = linear.coef_
intercept = linear.intercept_

for i in range(len(coeff)):
    print(maths.columns[i], ': ', coeff[i])
print('The intercept of our slope is: ', intercept)
```

```
The R^2 is:  0.8316968124174093
famrel :  0.5675565229094424
studytime :  -0.46968540367724887
failures :  -0.5180844666140698
absences :  0.029015504354730215
```

```
G1 :  0.11790811767680964
G2 :  0.954990328569435
The intercept of our slope is:  -2.3293128685712166
```

```
Result: Student grades are predicted using linear regression is
implemented
```

Result: Student grades are predicted using linear regression is implemented

✓ 0s completed at 1:08 PM