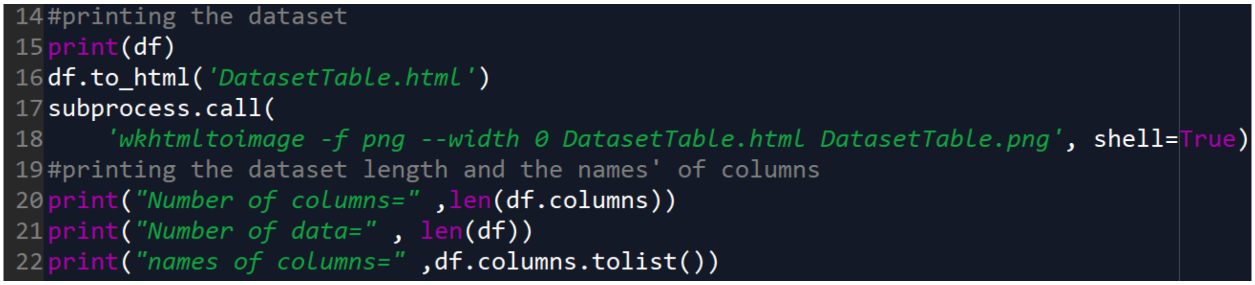
Mohammad Javad Ranjbar

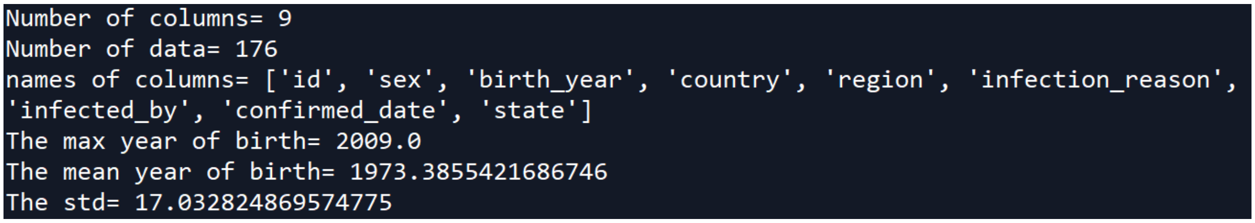
Data Mining

First question:

First extract the data from the file csv Open and display the requested values. We also convert the database into a table that is in the appendix.

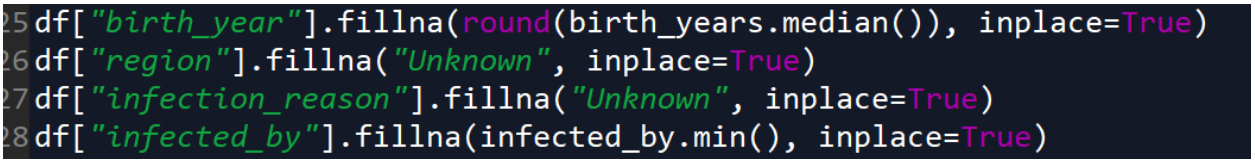


```
#printing the dataset
print(df)
df.to_html('DatasetTable.html')
subprocess.call(
    'wkhtmltoimage -f png --width 0 DatasetTable.html DatasetTable.png', shell=True)
#printing the dataset length and the names' of columns
print("Number of columns=" ,len(df.columns))
print("Number of data=" , len(df))
print("names of columns=" ,df.columns.tolist())
```



```
Number of columns= 9
Number of data= 176
names of columns= ['id', 'sex', 'birth_year', 'country', 'region', 'infection_reason',
'infected_by', 'confirmed_date', 'state']
The max year of birth= 2009.0
The mean year of birth= 1973.3855421686746
The std= 17.032824869574775
```

Now we give the appropriate value of data from data that is without value.



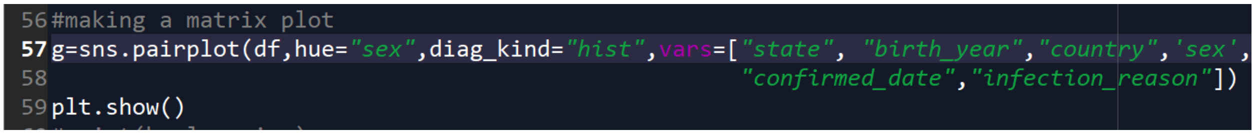
```
df["birth_year"].fillna(round(birth_years.median()), inplace=True)
df["region"].fillna("Unknown", inplace=True)
df["infection_reason"].fillna("Unknown", inplace=True)
df["infected_by"].fillna(infected_by.min(), inplace=True)
```

we have selected the year of birth as the average year of birth in the data.

For an unknown area and the cause of the infection in the absence of the amountunkown We have selected.

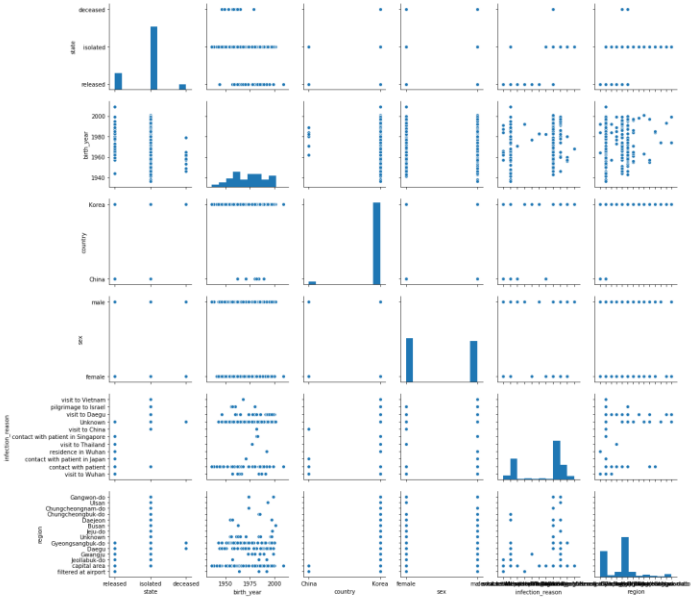
For data infected_by We chose the middle value.

Then we display the amount of data in the table.



```
#making a matrix plot
g=sns.pairplot(df,hue="sex",diag_kind="hist",vars=["state", "birth_year","country",'sex',
                                                   "confirmed_date","infection_reason"])
plt.show()
```

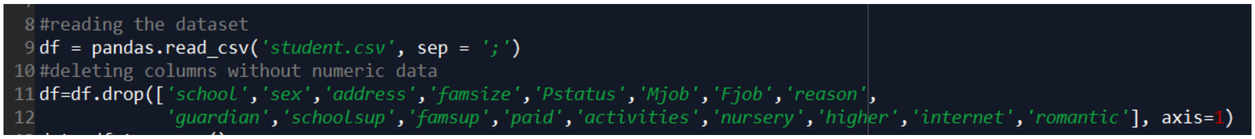
Result:



Yes, in some places it has data outlier Are ( data that do not follow the general trend ) that for the lack of adequate data or the data irrelevant for example, people from other countries . You can delete this data or add more data rates outlier Improved .

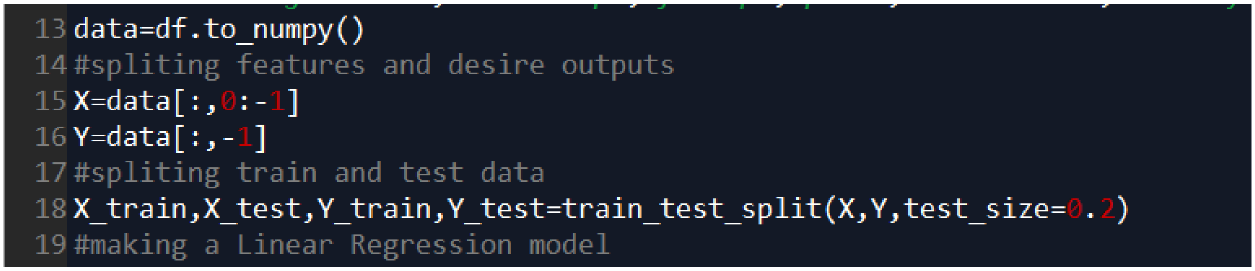
Second question:

First extract the data from the file csv Open and delete columns that have no numeric value.



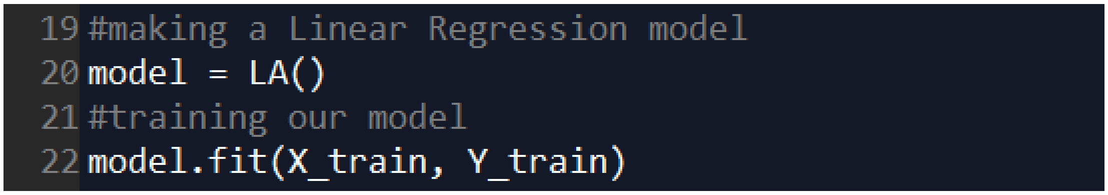
```
#reading the dataset
df = pandas.read_csv('student.csv', sep = ';')
#deleting columns without numeric data
df=df.drop(['school','sex','address','famsize','Pstatus','Mjob','Fjob','reason',
            'guardian','schoolsup','famsup','paid','activities','nursery','higher','internet','romantic'], axis=1)
```

The desired data ) student grades ( and input data or features from each other ,we separated the data into two categories divided into training and testing.



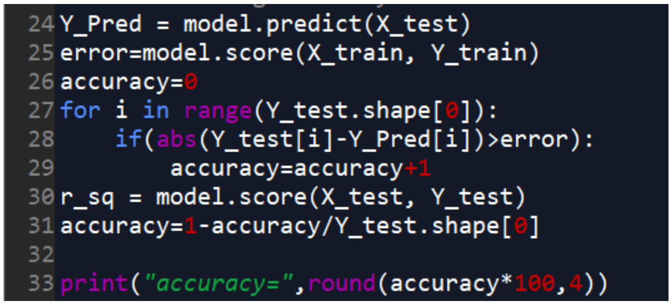
```
data=df.to_numpy()
#spliting features and desire outputs
X=data[:,0:-1]
Y=data[:,-1]
#spliting train and test data
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
#making a Linear Regression model
```

Now build your linear regression model and using the training data ,train We do.



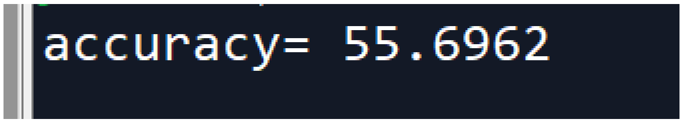
```
#making a Linear Regression model
model = LA()
#training our model
model.fit(X_train, Y_train)
```

Now using r2 score We calculate the accuracy value for the test data.



```
Y_Pred = model.predict(X_test)
error=model.score(X_train, Y_train)
accuracy=0
for i in range(Y_test.shape[0]):
    if(abs(Y_test[i]-Y_Pred[i])>error):
        accuracy=accuracy+1
r_sq = model.score(X_test, Y_test)
accuracy=1-accuracy/Y_test.shape[0]

print("accuracy=",round(accuracy*100,4))
```



```
accuracy= 55.6962
```