

Task2 #Need to include these data in excel #then select the column and o to home go to filter and then go to advanced select duplicates then highlight the duplicates #then execute the query:
`=IFERROR(INDEX(A3, MATCH(0, COUNTIF(B$1, A3:A12)+IF(COUNTIF(A3:A12, A3:A12)>1, 0, 1), 0)), "")`

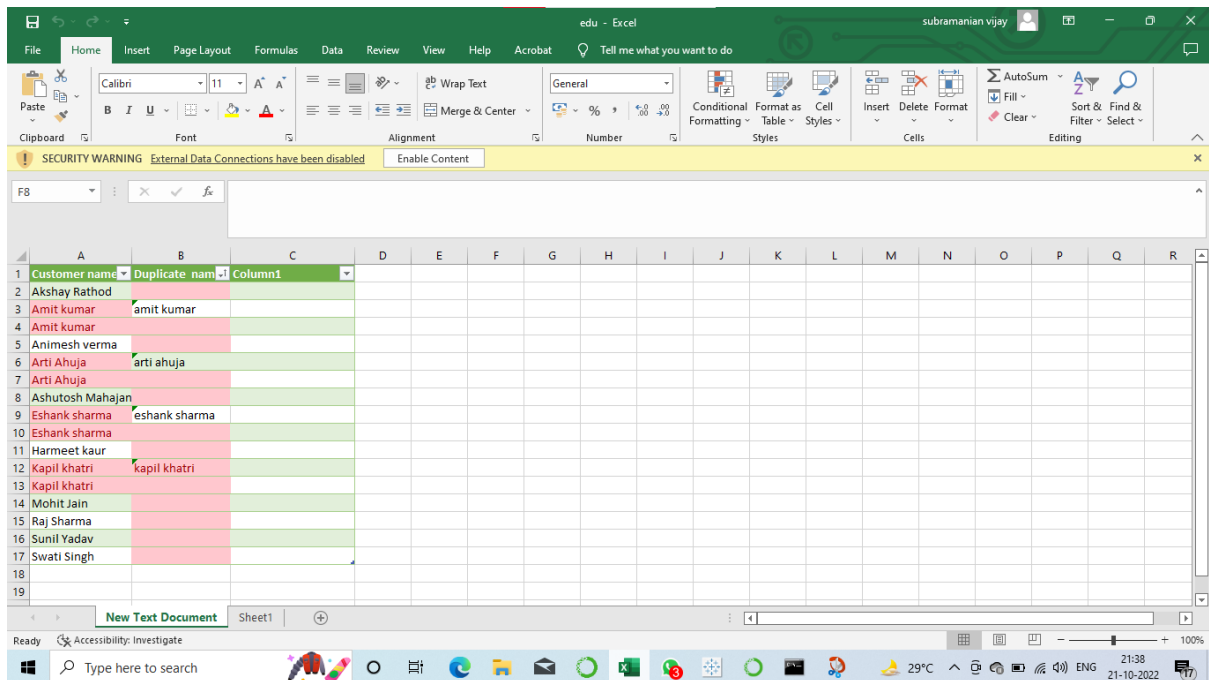
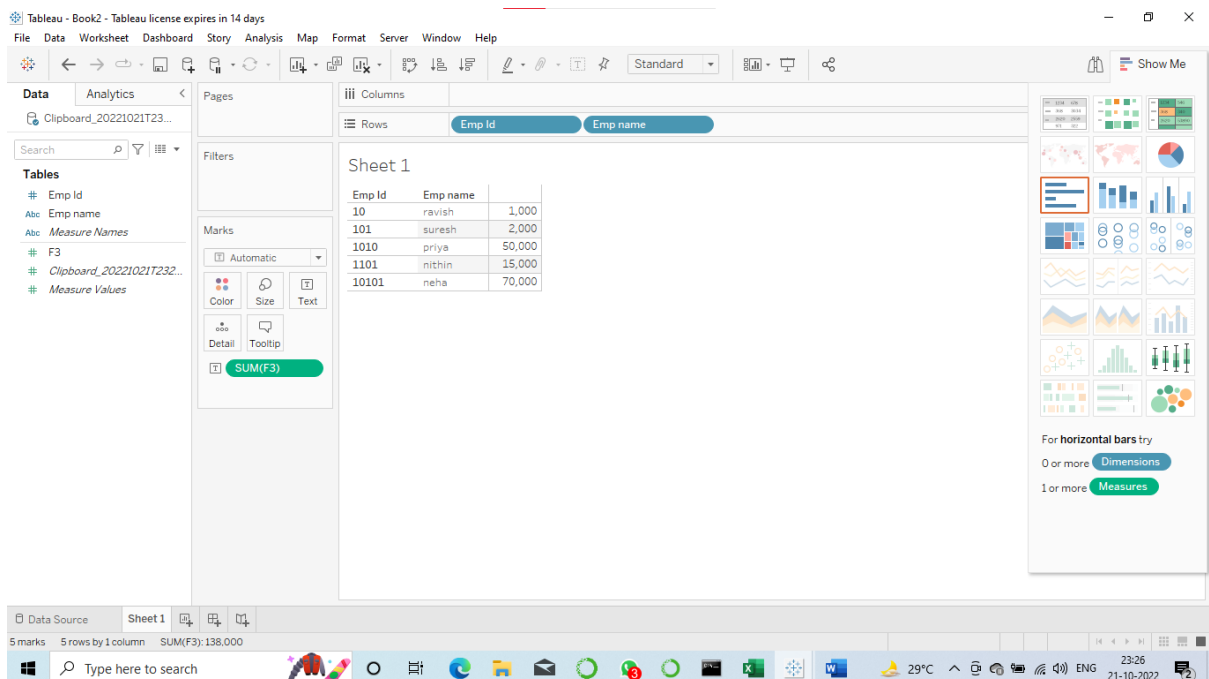


Tableau #Save the data in a csv file format #Import the data to the working sheet #Go to sheet 1 #Drag emp name and id #Right click on the id on the left side #a drop down box is appear #Rename that dialogue box type `LEFT("0000000", (7-LEN(STR[ID]))+STR([ID]))`



At St. Xavier's College, a Faculty has the following data in My SQL in database named as Class having table student related to Semester Examination

table student related to Semester Examination Enrollment No. Student Name Section Subject Id Marks

1	Tim	A	70
2	Jim	A	75
3	Kim	B	65
4	Tom	B	77
5	John	C	60
6	Joe	C	82
7	James	B	76
8	Henry	C	68
9	Matt	B	71
10	Paul	A	79

```
C:\Windows\system32\cmd.exe - sqlite3
main: "" r/w
sqlite> CREATE TABLE class_data_1 (enrollment_no varchar(10), student_name varchar(10), class_section varchar(10),subject_id varchar(10),marks INT, primary key (enrollment_no));
sqlite> insert into class_data_1 values(1,'TIM','A',1,70);
sqlite> insert into class_data_1 values(2,'JIM','A',2,75);
sqlite> insert into class_data_1 values(3,'Kim','B',3,65);
sqlite> insert into class_data_1 values(4,'TOM','B',4,77);
sqlite> insert into class_data_1 values(5,'JOHN','C',5,60);
sqlite> insert into class_data_1 values(6,'JOE','C',1,82);
sqlite> insert into class_data_1 values(7,'JAMES','B',2,76);
sqlite> insert into class_data_1 values(8,'HENRY','C',5,68);
sqlite> insert into class_data_1 values(9,'MATT','B',3,71);
sqlite> insert into class_data_1 values(10,'PAUL','A',4,79);
sqlite> select * from class_data_1;
1|TIM|A|1|70
2|JIM|A|2|75
3|Kim|B|3|65
4|TOM|B|4|77
5|JOHN|C|5|60
6|JOE|C|1|82
7|JAMES|B|2|76
8|HENRY|C|5|68
9|MATT|B|3|71
10|PAUL|A|4|79
sqlite> select Section,count(Section) as 'No.of Candidates greater than or equal to 75 marks'from class_data_1 group by Section;
Error: no such column: Section
sqlite> select section,count(section) as 'No.of Candidates greater than or equal to 75 marks'from class_data_1 group by Section;
Error: no such column: section
sqlite> select class_section,count(class_section) as 'No.of Candidates greater than or equal to 75 marks'from class_data_1 group by class_section;
Error: no such column: Section
sqlite> select class_section,count(class_section) as 'No.of Candidates greater than or equal to 75 marks'from class_data_1 group by class_section;
A|3
B|4
C|3
sqlite>
```

The Faculty needs a section-wise Number of candidates who have secured more than or equal to 75 marks in the Semester Exam. Note: Enrollment No. is declared as Primary Key

Fatal prediction using KNN Classifier Importing required libraries

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

In [2]: df=pd.read_csv("C:\\Users\\Hi\\OneDrive\\Desktop\\train.csv")
df.head()

Out[2]:
```

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variab
0	142.0	0.000	0.000	0.007	0.000	0.0	0.0	
1	122.0	0.000	0.000	0.006	0.002	0.0	0.0	
2	129.0	0.005	0.003	0.001	0.000	0.0	0.0	
3	136.0	0.006	0.000	0.008	0.000	0.0	0.0	
4	144.0	0.000	0.000	0.006	0.000	0.0	0.0	

5 rows x 22 columns

```
df.describe()
```

	baseline value	accelerations	fetal_movement	uterine_contractions	light_dec
count	1700.000000	1700.000000	1700.000000	1700.000000	17
mean	133.213529	0.003212	0.010211	0.004356	
std	9.873344	0.003888	0.050124	0.002943	
min	106.000000	0.000000	0.000000	0.000000	
25%	126.000000	0.000000	0.000000	0.002000	
50%	133.000000	0.002000	0.000000	0.004000	
75%	140.000000	0.006000	0.003000	0.006000	
max	159.000000	0.019000	0.481000	0.015000	

```
df.shape
```

```
(1700, 22)
```

```
df.isna().sum()
```

```
baseline value      0
accelerations       0
fetal_movement      0
uterine_contractions 0
light_decelerations 0
severe_decelerations 0
prolongued_decelerations 0
abnormal_short_term_variability 0
mean_value_of_short_term_variability 0
percentage_of_time_with_abnormal_long_term_variability 0
mean_value_of_long_term_variability 0
histogram_width     0
histogram_min       0
histogram_max       0
histogram_number_of_peaks 0
histogram_number_of_zeroes 0
histogram_mode      0
histogram_mean      0
histogram_median    0
histogram_variance  0
histogram_tendency  0
fetal_health        0
dtype: int64
```

```
df.info()
```

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

```
RangeIndex: 1700 entries, 0 to 1699
```

```
Data columns (total 22 columns):
```

#	Column	Non-Null Count	Dtype
0	baseline value	1700 non-null	float64
1	accelerations	1700 non-null	float64
2	fetal_movement	1700 non-null	float64
3	uterine_contractions	1700 non-null	float64
4	light_decelerations	1700 non-null	float64
5	severe_decelerations	1700 non-null	float64
6	prolongued_decelerations	1700 non-null	float64
7	abnormal_short_term_variability	1700 non-null	float64
8	mean_value_of_short_term_variability	1700 non-null	float64
9	percentage_of_time_with_abnormal_long_term_variability	1700 non-null	float64
10	mean_value_of_long_term_variability	1700 non-null	float64
11	histogram_width	1700 non-null	float64
12	histogram_min	1700 non-null	float64
13	histogram_max	1700 non-null	float64
14	histogram_number_of_peaks	1700 non-null	float64
15	histogram_number_of_zeroes	1700 non-null	float64
16	histogram_mode	1700 non-null	float64
17	histogram_mean	1700 non-null	float64
18	histogram_median	1700 non-null	float64
19	histogram_variance	1700 non-null	float64
20	histogram_tendency	1700 non-null	float64
21	fetal_health	1700 non-null	float64

```
dtypes: float64(22)
```

```
df.describe()
```

	baseline value	accelerations	fetal_movement	uterine_contractions	light_dec
count	1700.000000	1700.000000	1700.000000	1700.000000	17
mean	133.213529	0.003212	0.010211	0.004356	
std	9.873344	0.003888	0.050124	0.002943	
min	106.000000	0.000000	0.000000	0.000000	
25%	126.000000	0.000000	0.000000	0.002000	
50%	133.000000	0.002000	0.000000	0.004000	
75%	140.000000	0.006000	0.003000	0.006000	
max	159.000000	0.019000	0.481000	0.015000	

```
df.shape
```

```
(1700, 22)
```

```
df.isna().sum()
```

baseline_value	0
accelerations	0
fetal_movement	0
uterine_contractions	0
light_decelerations	0
severe_decelerations	0
prolongued_decelerations	0
abnormal_short_term_variability	0
mean_value_of_short_term_variability	0
percentage_of_time_with_abnormal_long_term_variability	0
mean_value_of_long_term_variability	0
histogram_width	0
histogram_min	0
histogram_max	0
histogram_number_of_peaks	0
histogram_number_of_zeroes	0
histogram_mode	0
histogram_mean	0
histogram_median	0
histogram_variance	0
histogram_tendency	0
fetal_health	0
dtype: int64	

```
df.info()
```

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

```
RangeIndex: 1700 entries, 0 to 1699
```

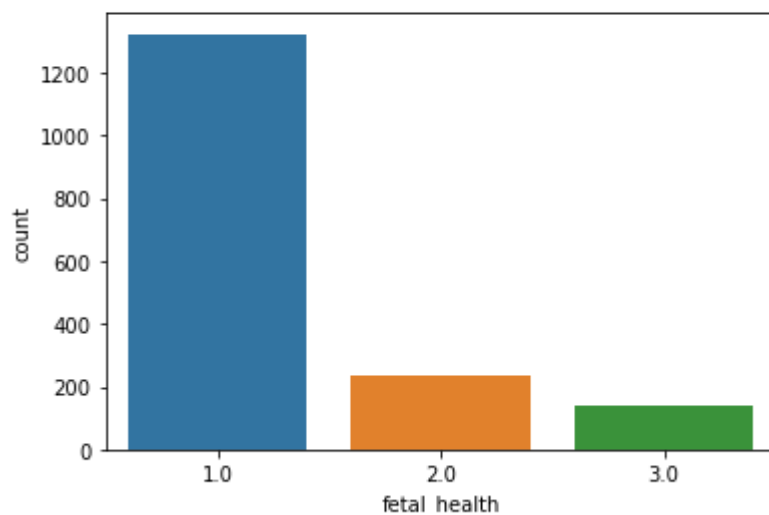
```
Data columns (total 22 columns):
```

#	Column	Non-Null Count	Dtype
0	baseline_value	1700 non-null	float64
1	accelerations	1700 non-null	float64
2	fetal_movement	1700 non-null	float64
3	uterine_contractions	1700 non-null	float64
4	light_decelerations	1700 non-null	float64
5	severe_decelerations	1700 non-null	float64
6	prolongued_decelerations	1700 non-null	float64
7	abnormal_short_term_variability	1700 non-null	float64
8	mean_value_of_short_term_variability	1700 non-null	float64
9	percentage_of_time_with_abnormal_long_term_variability	1700 non-null	float64
10	mean_value_of_long_term_variability	1700 non-null	float64
11	histogram_width	1700 non-null	float64
12	histogram_min	1700 non-null	float64
13	histogram_max	1700 non-null	float64
14	histogram_number_of_peaks	1700 non-null	float64
15	histogram_number_of_zeroes	1700 non-null	float64
16	histogram_mode	1700 non-null	float64
17	histogram_mean	1700 non-null	float64
18	histogram_median	1700 non-null	float64
19	histogram_variance	1700 non-null	float64
20	histogram_tendency	1700 non-null	float64
21	fetal_health	1700 non-null	float64

```
dtypes: float64(22)
```

```
sns.countplot(df.fetal_health)
```

```
<AxesSubplot:xlabel='fetal_health', ylabel='count'>
```



edu - Excel subramanian vijay

File Home Insert Page Layout Formulas Data Review View Help Acrobat Tell me what you want to do

Clipboard Font Alignment Number Conditional Formatting Styles Cell Styles Insert Delete Format AutoSum Sort & Find & Filter Select

SECURITY WARNING External Data Connections have been disabled Enable Content

F8

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Customer name	Duplicate nam	Column1															
2	Akshay Rathod																	
3	Amit kumar	amit kumar																
4	Amit kumar																	
5	Animesh verma																	
6	Arti Ahuja	arti ahuja																
7	Arti Ahuja																	
8	Ashutosh Mahajan																	
9	Eshank sharma	eshank sharma																
10	Eshank sharma																	
11	Harmeet kaur																	
12	Kapil khatri	kapil khatri																
13	Kapil khatri																	
14	Mohit Jain																	
15	Raj Sharma																	
16	Sunil Yadav																	
17	Swati Singh																	
18																		
19																		

New Text Document Sheet1

Ready Accessibility: Investigate

Type here to search

```

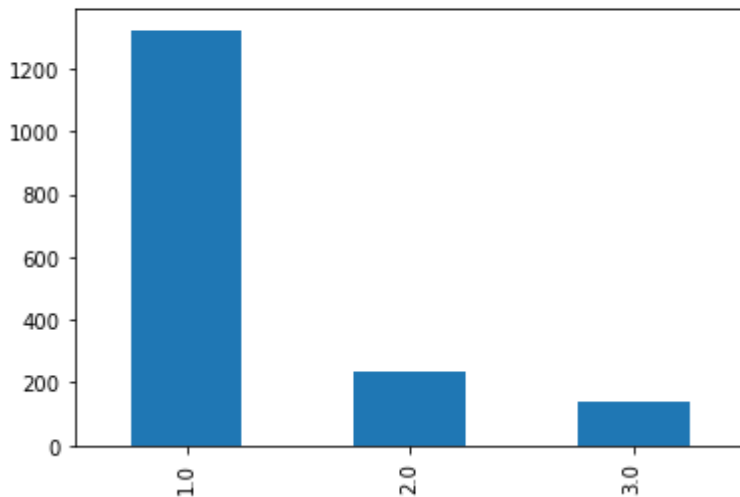
cats = list(df.select_dtypes(include=['object','bool']))
nums = list(df.select_dtypes(include=['int64','float64']))
print(cats)
print(nums)

[]
['baseline_value', 'accelerations', 'fetal_movement', 'uterine_contractions', 'light_decelerations', 'severe_decelerations', 'prolonged_decelerations', 'abnormal_short_term_variability', 'mean_value_of_short_term_variability', 'percentage_of_time_with_a_bnormal_long_term_variability', 'mean_value_of_long_term_variability', 'histogram_width', 'histogram_min', 'histogram_max', 'histogram_number_of_peaks', 'histogram_number_of_zeroes', 'histogram_mode', 'histogram_mean', 'histogram_median', 'histogram_variance', 'histogram_tendency', 'fetal_health']

df['fetal_health'].value_counts().plot(kind='bar')

```

<AxesSubplot:>



```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: df=pd.read_csv("C:\\Users\\Hi\\OneDrive\\Desktop\\train.csv")
df.head()
```

```
Out[2]:
```

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variab
0	142.0	0.000	0.000	0.007	0.000	0.0	0.0	:
1	122.0	0.000	0.000	0.006	0.002	0.0	0.0	:
2	129.0	0.005	0.003	0.001	0.000	0.0	0.0	:
3	136.0	0.006	0.000	0.008	0.000	0.0	0.0	:
4	144.0	0.000	0.000	0.006	0.000	0.0	0.0	:

5 rows x 22 columns

```
from sklearn.neighbors import KNeighborsClassifier
classifier=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
classifier.fit(X_train,y_train)
KNeighborsClassifier()
y_pred=classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1),y_test.reshape(len
(y_test),1)),1))
```

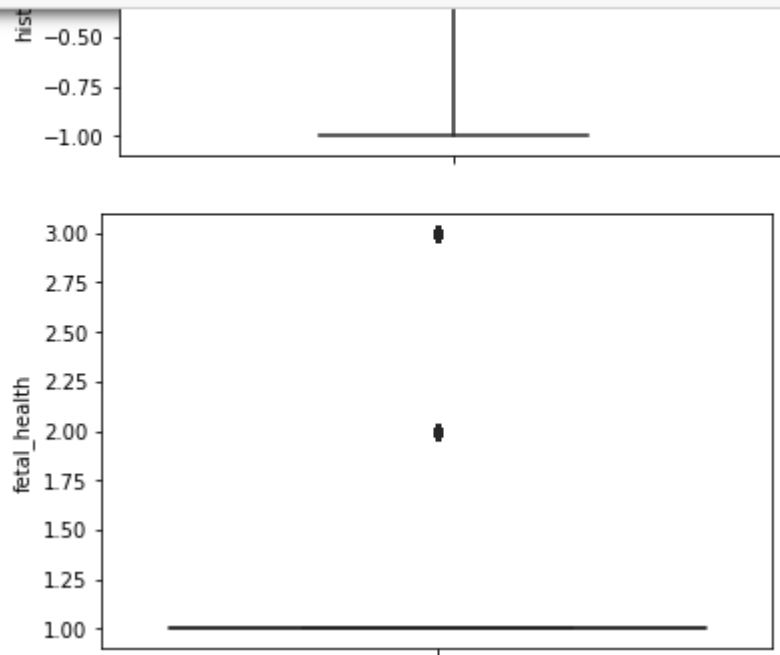
```
[[1. 1.]
 [3. 3.]
 [1. 1.]
 ...
 [1. 1.]
 [1. 1.]
 [2. 1.]]
```

```
from sklearn.preprocessing import StandardScaler
SC=StandardScaler()
X_train=SC.fit_transform(X_train)
X_test=SC.fit_transform(X_test)
print(X_train)
```

```
[[ -1.46513509 -0.83485626 -0.20820521 ... -1.18281773 -0.60623421
  -2.14821249]
 [ 0.16393359 -0.83485626 -0.0746156 ... -0.08025974 -0.53911612
   1.13129086]
 [-1.2615015   2.2251405  -0.00782079 ... -0.42480911 -0.37132092
  -2.14821249]
 ...
 [ 1.28391831  1.71514104 -0.20820521 ...  1.160118  -0.50555708
  -0.50846081]
 [ 1.08028472 -0.83485626  0.6823922 ...  0.74665876 -0.06928955
   1.13129086]
 [ 0.97846793 -0.3248568   0.94957142 ...  0.74665876  1.00459976
  -0.50846081]]
```



```
for i in range(0, len(nums )):
    sns.boxplot(y=df[nums[i]],color='green',orient='v')
plt.show()
```



```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: df=pd.read_csv("C:\\Users\\Hi\\OneDrive\\Desktop\\train.csv")
df.head()
```

Out[2]:

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variab
0	142.0	0.000	0.000	0.007	0.000	0.0	0.0	
1	122.0	0.000	0.000	0.006	0.002	0.0	0.0	
2	129.0	0.005	0.003	0.001	0.000	0.0	0.0	
3	136.0	0.006	0.000	0.008	0.000	0.0	0.0	
4	144.0	0.000	0.000	0.006	0.000	0.0	0.0	

5 rows x 22 columns