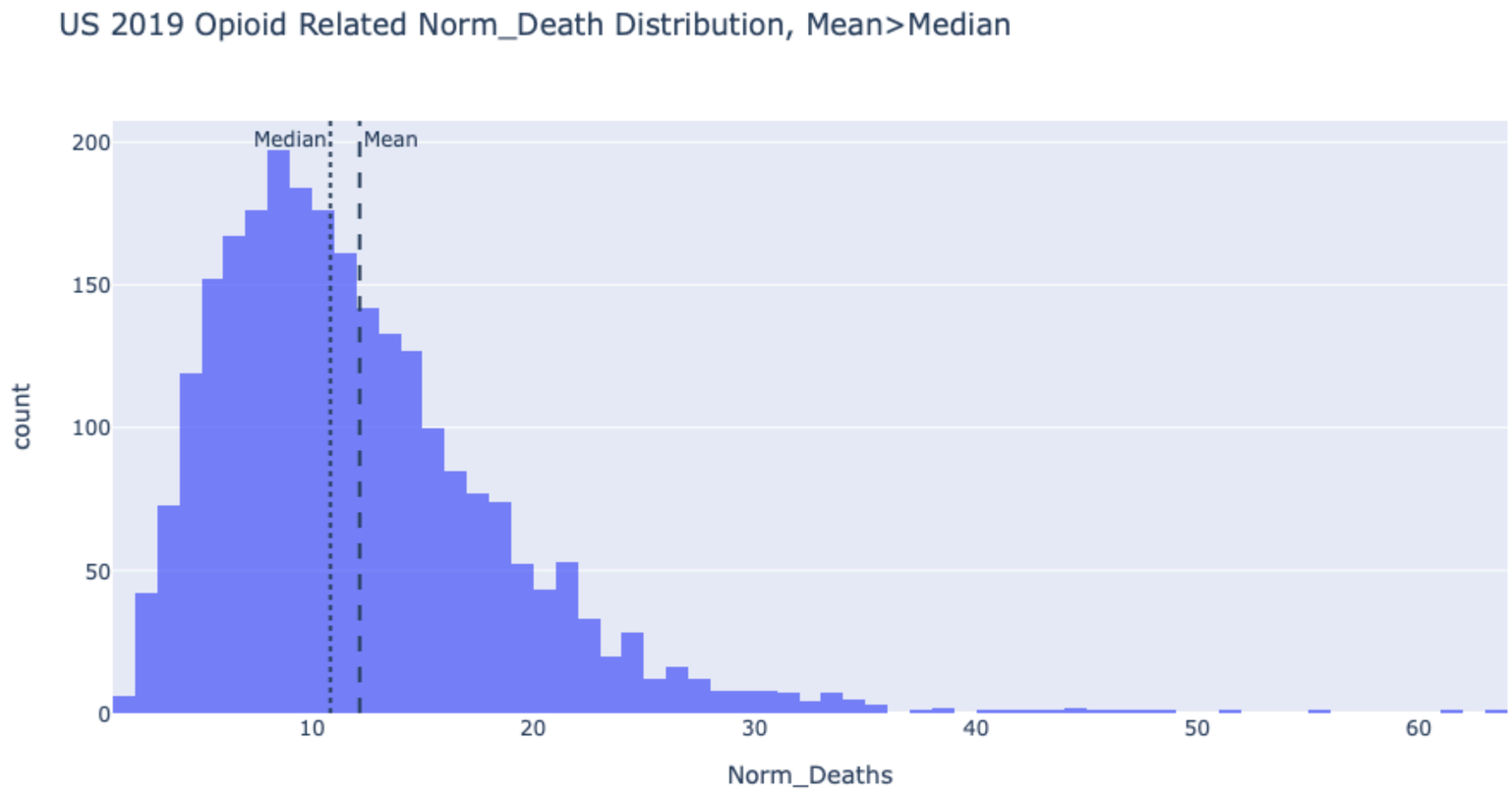
Understand the distribution and measures of center for US opioid mortality for 2019

- For this task we need to import the data and then create a histogram.
- Adding the vertical lines of mean and median.
- If the histogram is skewed right, the mean is greater than the median (A few large values drive the mean upward).
- But they are really close to each other. I think it is better to consider it symmetric.



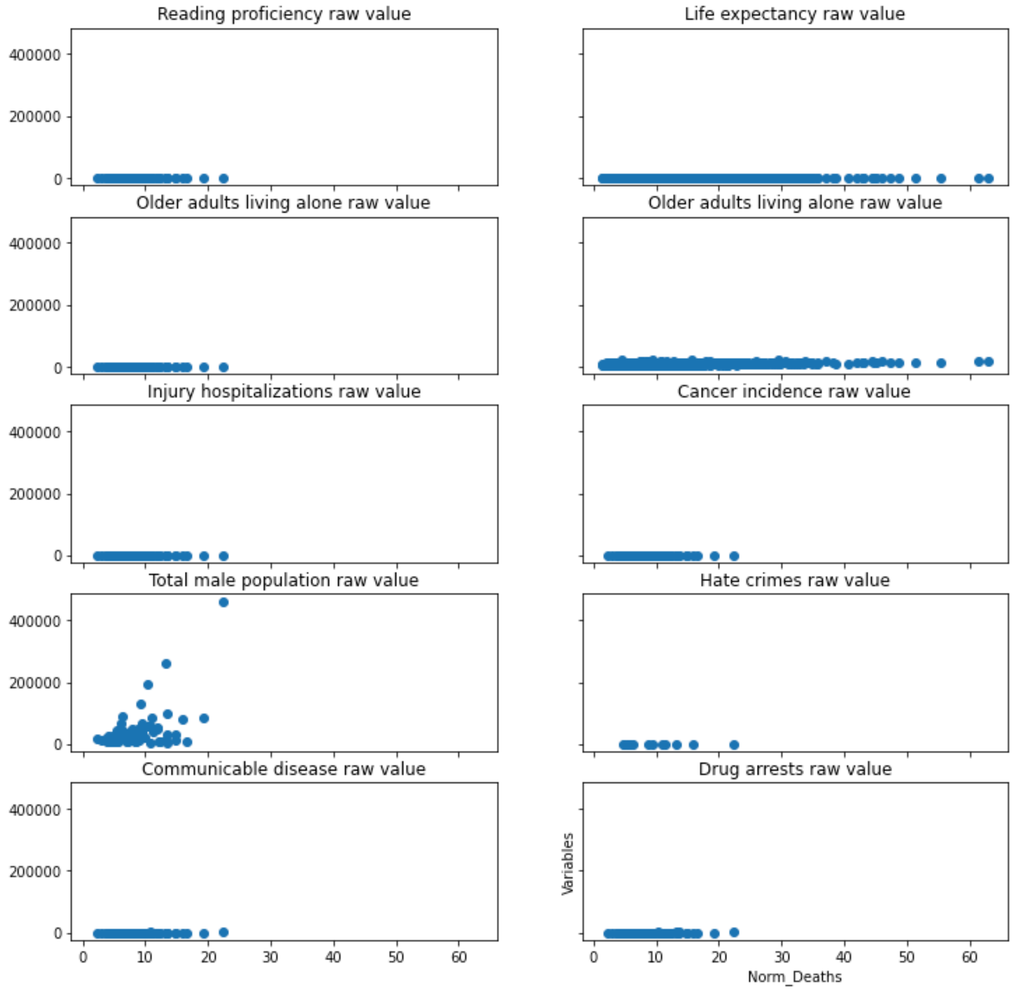
Codify the column Normalized Deaths and store them as a “label’ column

- Minimum Norm_Deaths in my data frame is less than 1.999. So, I'll change the 1.99 to 0.
- I use pandas cut() method to categorize values in Norm_Deaths columns into four categories.
- cut() method takes an input series that need to be grouped into various bins.
- cut() method takes a list of values that begins with the smallest value for the first bin and ends with the maximum value for the last bin. Break points between the bins also exist in the list. Here, I assumed the lower band to be inclusive.
- cut() method also takes a list of bins name. This list have one element less than bin list.

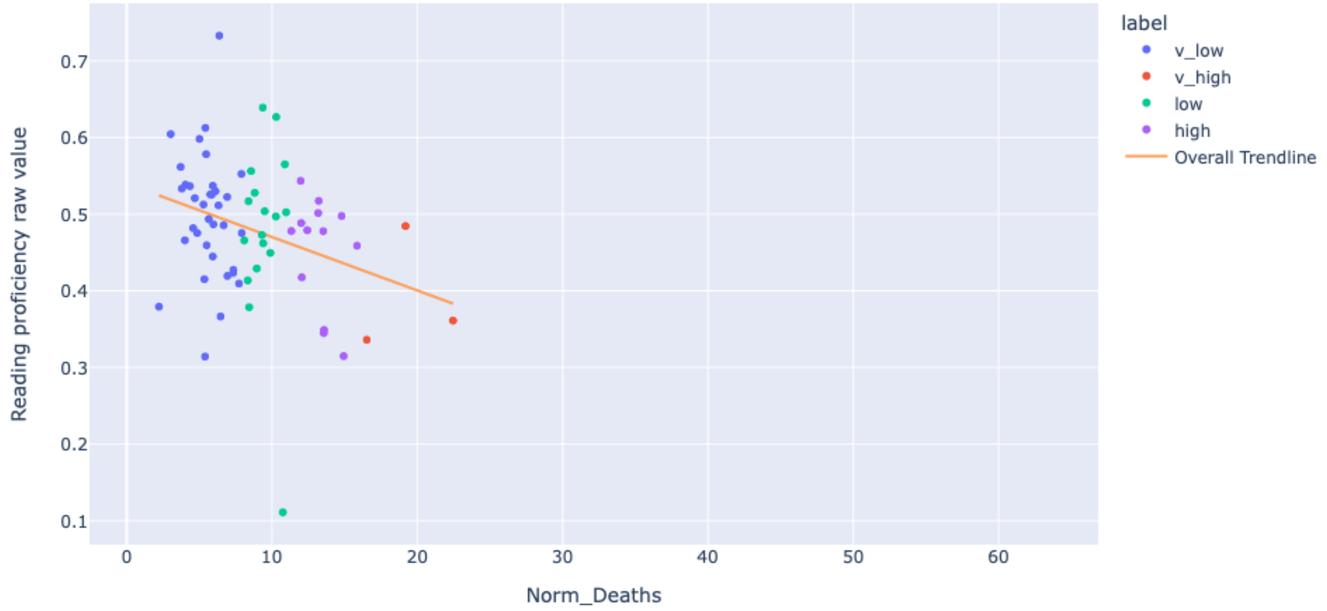
```
In [40]: super_df['label'] = pd.cut(super_df['Norm_Deaths'], bins = [0,8,11,16,64], labels = ['v_low','low','high','v_high']
         super_df['label'].unique()

Out[40]: ['v_low', 'v_high', 'low', 'high']
         Categories (4, object): ['v_low' < 'low' < 'high' < 'v_high']
```

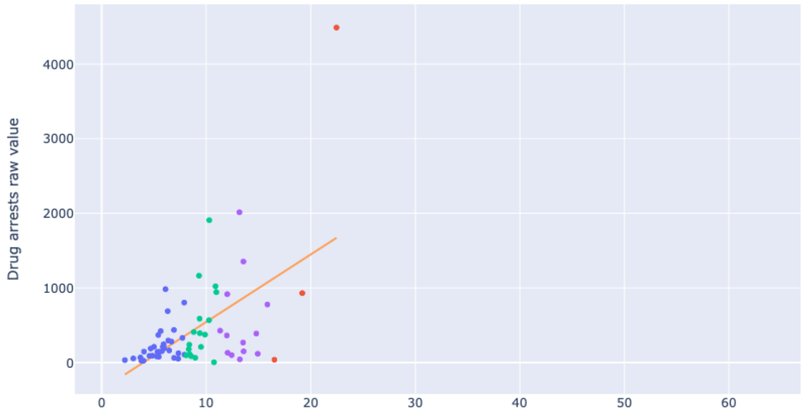
Now try the variables you have identified in Stage I and plot them as a second variable to Normalized Mortality in a scatter plot to observe any trends.

- For this task, first I tries to plot all of them in a subplot. But because their values are not in the same period the result was not satisfying.

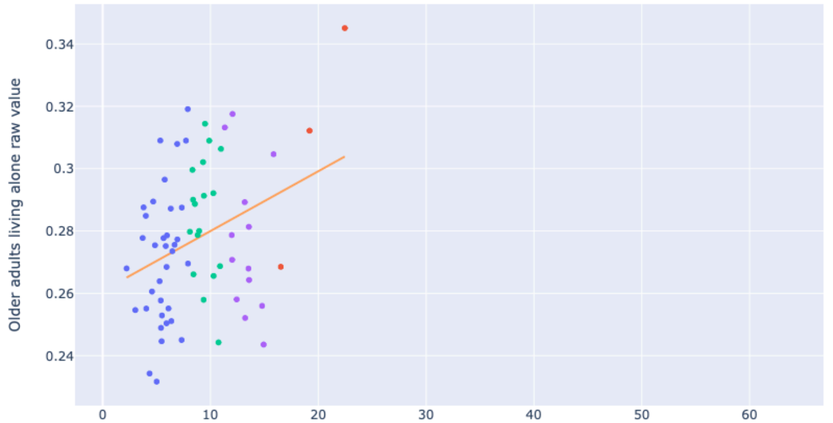


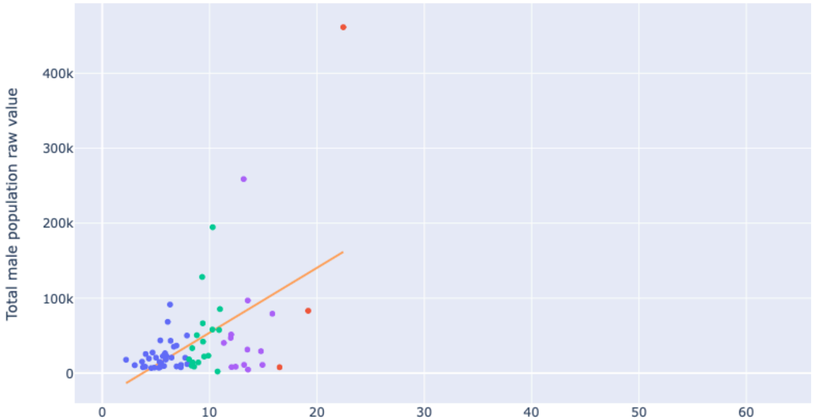
- So, I create separate scatter plots for each of them.
- In my first trial, I selected 10 columns that I believed correlated well with Norm_Death column.
- When creating the scatter plot, I thought it'll be useful to add a trend_line to my visual and color code observation based on the categories defined above. For trend_line I used OLS method or ordinary least square approach. However, I am not sure if this is a right method or if there are other methods available.
- Only five out of ten variables that I have initially selected show strong visual correlation with the Norm_Deaths variable. These are 'Drug arrest raw values', 'Communicable disease raw value', 'Total male population raw value', 'Hate crimes raw value' and 'Cancer incidence raw value' variables. I think I need to take a look at most columns and come up with a better selection!




Drug arrests raw value
4000
3000
2000
1000
0
0
10
20
30
40
50
60

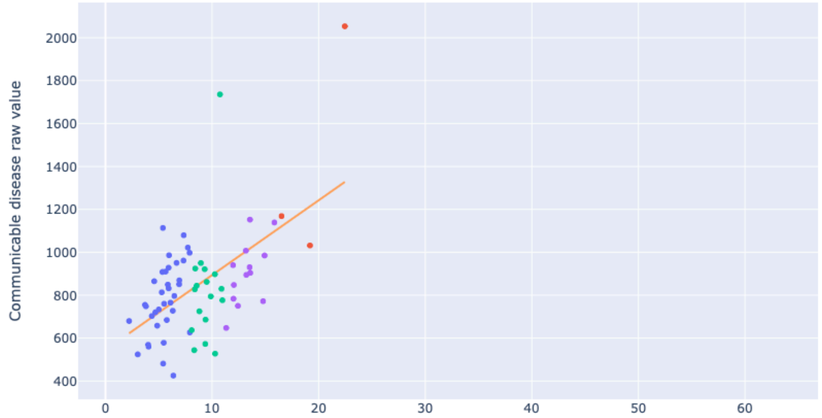

Cancer incidence raw value
600
550
500
450
400
350
0
10
20
30
40
50
60

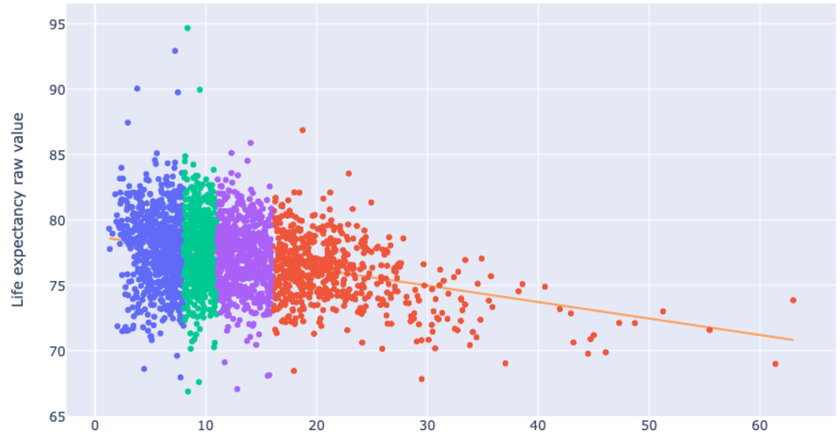

Older adults living alone raw value
0.34
0.32
0.3
0.28
0.26
0.24
0
10
20
30
40
50
60

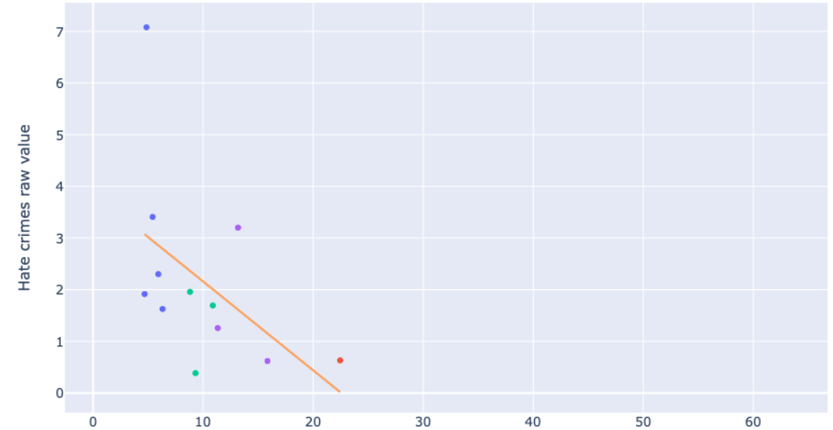

Total male population raw value
400k
300k
200k
100k
0
0
10
20
30
40
50
60


Injury hospitalizations raw value
900
800
700
600
500
400
300
200
0
10
20
30
40
50
60

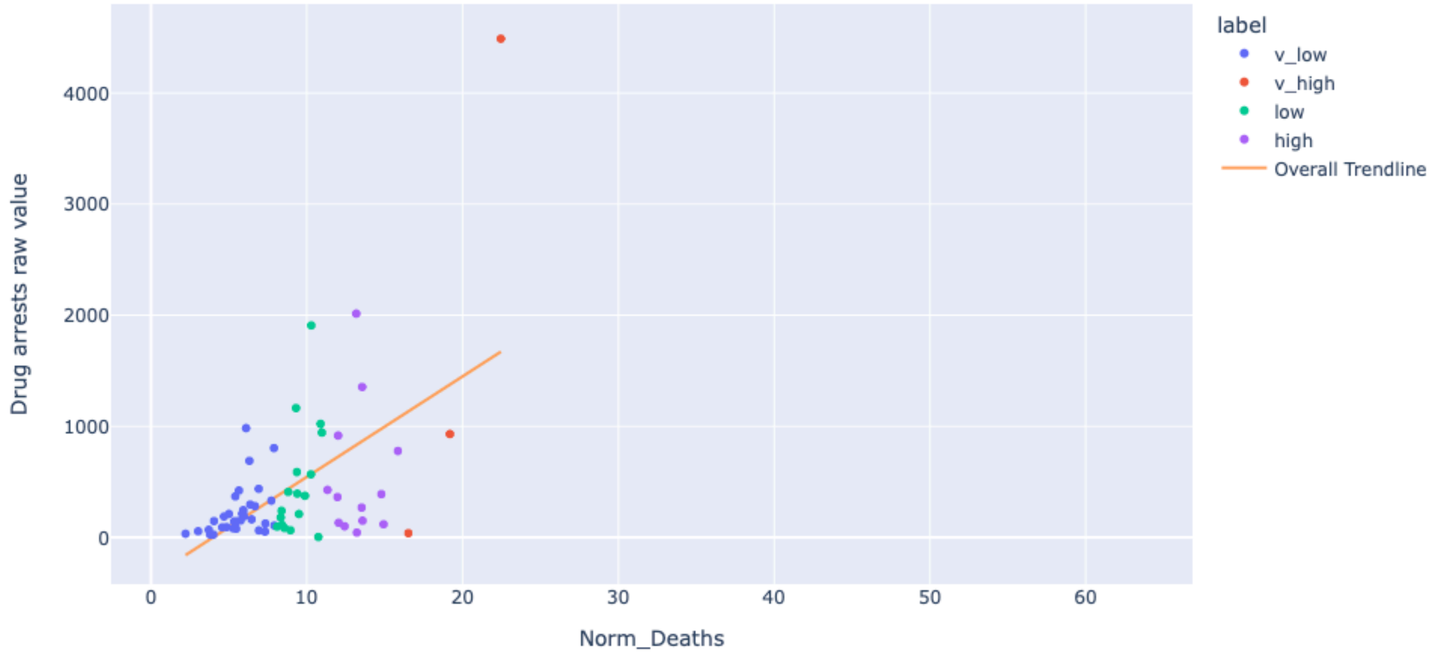

Premature death raw value
20k
15k
10k
5k
0
10
20
30
40
50
60

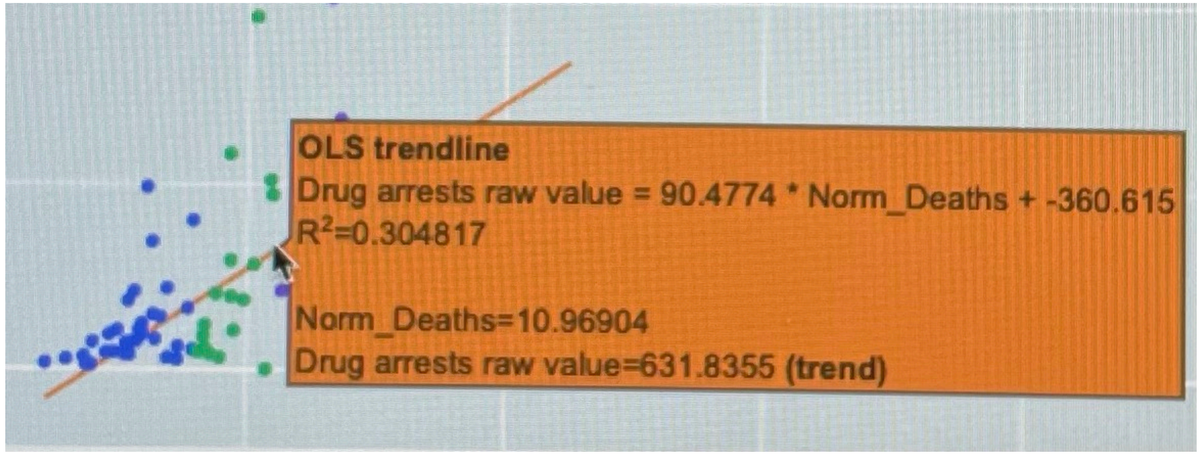

Communicable disease raw value
2000
1800
1600
1400
1200
1000
800
600
400
0
10
20
30
40
50
60


Life expectancy raw value
95
90
85
80
75
70
65
0
10
20
30
40
50
60


Hate crimes raw value
7
6
5
4
3
2
1
0
0
10
20
30
40
50
60

In [108]:
```
px.scatter(super_df, y='Drug arrests raw value', x="Norm_Deaths",trendline="ols",color='label',trendline_scope="over
```

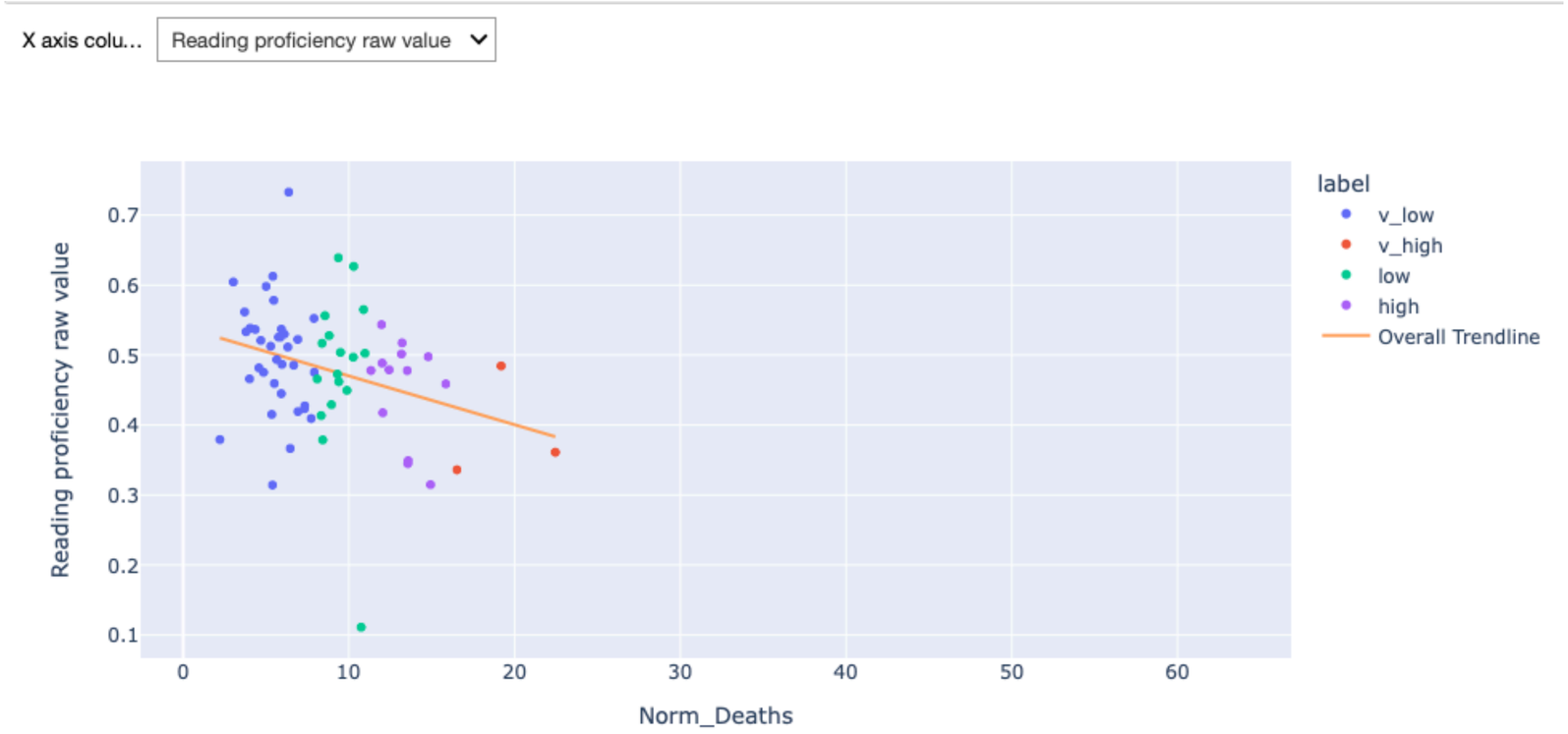




- The R square for the variables are listed in the table.

Variable	R square
Reading Proficiency	0.09
Life Expectancy	0.09
Premature Death	0.11
Older Adults	0.10
Injury Hospitalization	0.11
Cancer Incidence	0.21
Hate Crimes	0.25
Total Male Population	0.26
Communicable disease	0.31
Drug Arrest	0.30

- I used "Plotly widget" to develop a dashboard to visually see how these variables interact with Norm_Death variable.
- So, I first created a list of these ten variables and named it as independent variables. Then, I created the widgets dropdown that can only takes values in this list. Then I followed the guid lines in provided link and added one line of code to show the scatter plot as well as the dropdown.



```
import plotly.graph_objects as go
from ipywidgets import widgets

independent_variables = ['Reading proficiency raw value','Life expectancy raw value','Older adults living alone raw
,'Injury hospitalizations raw value','Cancer incidence raw value','Hate crimes raw value','Total male population raw


X_Columns = widgets.Dropdown(
    options= independent_variables,
    description='X axis column',
)


g = go.FigureWidget(px.scatter(super_df, x=X_Columns.value, y="Norm_Deaths",trendline="ols",color='label',trendline_
                   layout=go.Layout())


def validate():
    if X_Columns.value in independent_variables:
        return True
    else:
        return False


def response(change):
    if validate():
        tmp_df = super_df
        with g.batch_update():
            g.data[0].x = tmp_df[X_Columns.value]
            g.layout.xaxis.title = X_Columns.value


X_Columns.observe(response, names="value")
container = widgets.HBox([X_Columns])
widgets.VBox([container,g])
```