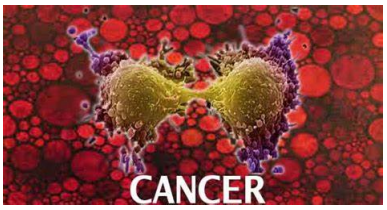
Double-click (or enter) to edit

```
from google.colab import drive
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as ex
```

source: 'cancer reg'




```
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
data = pd.read_csv('/content/drive/MyDrive/dataset/cancer_reg.csv')
```

```
data.head()
```

	avganncount	avgdeathsperyear	target_deathrate	incidencerate	medincome	popest2015	povertypercent	studypercap	binnedinc	medianage	...	pc
0	1397.0	469	164.9	489.8	61898	260131	11.2	499.748204	(61494.5, 125635]	39.3	...	
1	173.0	70	161.3	411.6	48127	43269	18.6	23.111234	(48021.6, 51046.4]	33.0	...	
2	102.0	50	174.7	349.7	49348	21026	14.6	47.560164	(48021.6, 51046.4]	45.0	...	
3	427.0	202	194.8	430.4	44243	75882	17.1	342.637253	(42724.4, 45201]	42.8	...	
4	57.0	26	144.4	350.1	49955	10321	12.5	0.000000	(48021.6, 51046.4]	48.3	...	

5 rows × 33 columns

TABLE COLUMNS

1.avganncount - Mean number of reported cases of cancer diagnosed annually (a)

2.avgdeathsperyear - Mean number of reported mortalities due to cancer (a)

3.target_deathrate - Dependent variable. Mean per capita (100,000) cancer mortalities (a)

4.incidencerate - Mean per capita (100,000) cancer diagnoses (a)

5.medincome - Median income per county (b)

6.popest2015 - Population of county (b)

7.povertypercent - Percent of populace in poverty (b)

8.studypercap - Per capita number of cancer-related clinical trials per county (a)

9.binnedinc - Median income per capita binned by decile (b)

10.medianage - Median age of county residents (b)

11.medianagemale - Median age of male county residents (b)

12.medianagefemale - Median age of female county residents (b)

13.geography - County name (b)

14.percentmarried - Percent of county residents who are married (b)

15.pctnohs18_24 - Percent of county residents ages 18-24 highest education attained: less than high school (b)

16.pcths18_24 - Percent of county residents ages 18-24 highest education attained: high school diploma (b)

17.pctsomecol18_24 - Percent of county residents ages 18-24 highest education attained: some college (b)

18.pctbachdeg18_24 - Percent of county residents ages 18-24 highest education attained: bachelor's degree (b)

19.pcths25_over - Percent of county residents ages 25 and over highest education attained: high school diploma (b)

20.pctbachdeg25_over - Percent of county residents ages 25 and over highest education attained: bachelor's degree (b)

21.pctemployed16_over - Percent of county residents ages 16 and over employed (b)

22.pctunemployed16_over - Percent of county residents ages 16 and over unemployed (b)

23.pctprivatecoverage - Percent of county residents with private health coverage (b)

24.pctprivatecoveragealone - Percent of county residents with private health coverage alone (no public assistance) (b)

25.pctemprivcoverage - Percent of county residents with employee-provided private health coverage (b)

26.pctpubliccoverage - Percent of county residents with government-provided health coverage (b)

27.pctpubliccoveragealone - Percent of county residents with government-provided health coverage alone (b)

28.pctwhite - Percent of county residents who identify as White (b)

29.pctblack - Percent of county residents who identify as Black (b)

30.pctasian - Percent of county residents who identify as Asian (b)

31.pctotherrace - Percent of county residents who identify in a category which is not White, Black, or Asian (b)

32.pctmarriedhouseholds - Percent of householdsmarried (b)

33.birthrate - Number of live births relative to number of women in county (b)

Double-click (or enter) to edit

```
data.shape
```

```
(3047, 33)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3047 entries, 0 to 3046
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   avganncount              3047 non-null   float64
 1   avgdeathsperyear         3047 non-null   int64
 2   target_deathrate         3047 non-null   float64
 3   incidencerate            3047 non-null   float64
```

```
 4   medincome               3047 non-null   int64
 5   popest2015              3047 non-null   int64
 6   povertypercent          3047 non-null   float64
 7   studypercap             3047 non-null   float64
 8   binnedinc               3047 non-null   object
 9   medianage               3047 non-null   float64
 10  medianagemale           3047 non-null   float64
 11  medianagefemale         3047 non-null   float64
 12  geography               3047 non-null   object
 13  percentmarried          3047 non-null   float64
 14  pctnohs18_24            3047 non-null   float64
 15  pcths18_24              3047 non-null   float64
 16  pctsomecol18_24         762 non-null    float64
 17  pctbachdeg18_24         3047 non-null   float64
 18  pcths25_over            3047 non-null   float64
 19  pctbachdeg25_over       3047 non-null   float64
 20  pctemployed16_over      2895 non-null   float64
 21  pctunemployed16_over    3047 non-null   float64
 22  pctprivatecoverage      3047 non-null   float64
 23  pctprivatecoveragealone 2438 non-null   float64
 24  pctemppprivcoverage     3047 non-null   float64
 25  pctpubliccoverage       3047 non-null   float64
 26  pctpubliccoveragealone  3047 non-null   float64
 27  pctwhite                3047 non-null   float64
 28  pctblack                3047 non-null   float64
 29  pctasian                3047 non-null   float64
 30  pctotherrace            3047 non-null   float64
 31  pctmarriedhouseholds    3047 non-null   float64
 32  birthrate               3047 non-null   float64
dtypes: float64(28), int64(3), object(2)
memory usage: 785.7+ KB
```

Exploratory Data Analysis

```
data.describe().T
```

	count	mean	std	min	25%	50
avganncount	3047.0	606.338544	1416.356223	6.000000	76.000000	171.00000
avgdeathsperyear	3047.0	185.965868	504.134286	3.000000	28.000000	61.00000
target_deathrate	3047.0	178.664063	27.751511	59.700000	161.200000	178.10000
incidencerate	3047.0	448.268586	54.560733	201.300000	420.300000	453.54942
medincome	3047.0	47063.281917	12040.090836	22640.000000	38882.500000	45207.00000
popest2015	3047.0	102637.370528	329059.220504	827.000000	11684.000000	26643.00000
povertypercent	3047.0	16.878175	6.409087	3.200000	12.150000	15.90000
studypercap	3047.0	155.399415	529.628366	0.000000	0.000000	0.00000
medianage	3047.0	45.272333	45.304480	22.300000	37.700000	41.00000
medianagemale	3047.0	39.570725	5.226017	22.400000	36.350000	39.60000
medianagefemale	3047.0	42.145323	5.292849	22.300000	39.100000	42.40000
percentmarried	3047.0	51.773679	6.896928	23.100000	47.750000	52.40000
pctnohs18_24	3047.0	18.224450	8.093064	0.000000	12.800000	17.10000
pcths18_24	3047.0	35.002068	9.069722	0.000000	29.200000	34.70000
pctsomecol18_24	762.0	40.977034	11.115805	7.100000	34.000000	40.40000
pctbachdeg18_24	3047.0	6.158287	4.529059	0.000000	3.100000	5.40000
pcths25_over	3047.0	34.804660	7.034924	7.500000	30.400000	35.30000
pctbachdeg25_over	3047.0	13.282015	5.394756	2.500000	9.400000	12.30000
pctemployed16_over	2895.0	54.152642	8.315064	17.600000	48.600000	54.50000
pctunemployed16_over	3047.0	7.852412	3.452371	0.400000	5.500000	7.60000
pctprivatecoverage	3047.0	64.354939	10.647057	22.300000	57.200000	65.10000
pctprivatecoveragealone	2438.0	48.453774	10.083006	15.700000	41.000000	48.70000
pctemppprivcoverage	3047.0	41.196324	9.447687	13.500000	34.500000	41.10000
pctpubliccoverage	3047.0	36.252642	7.841741	11.200000	30.900000	36.30000
pctpubliccoveragealone	3047.0	19.240072	6.113041	2.600000	14.850000	18.80000
pctwhite	3047.0	83.645286	16.380025	10.199155	77.296180	90.05977
pctblack	3047.0	9.107978	14.534538	0.000000	0.620675	2.24757
pctasian	3047.0	1.253965	2.610276	0.000000	0.254199	0.54981
pctotherrace	3047.0	1.983523	3.517710	0.000000	0.295172	0.82618
pctmarriedhouseholds	3047.0	51.243872	6.572814	22.992490	47.763063	51.66994

treat: Target variable

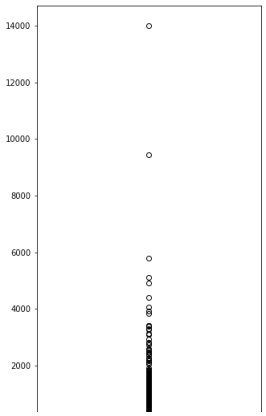
```
data['avgdeath']=data['avgdeathsperyear'].astype('int64')
data2=data.drop(['avgdeathsperyear'],axis='columns')
data2
```

	avganncount	target_deathrate	incidencerate	medincome	popest2015	povertypercent	studyperca
0	1397.000000	164.9	489.800000	61898	260131	11.2	499.74820
1	173.000000	161.3	411.600000	48127	43269	18.6	23.11123
2	102.000000	174.7	349.700000	49348	21026	14.6	47.56016
3	427.000000	194.8	430.400000	44243	75882	17.1	342.63725
4	57.000000	144.4	350.100000	49955	10321	12.5	0.00000
...	...	...	...	...	...	...	.
3042	1962.667684	149.6	453.549422	46961	6343	12.4	0.00000
3043	1962.667684	150.1	453.549422	48609	37118	18.8	377.17549
3044	1962.667684	153.9	453.549422	51144	34536	15.0	1968.95992
3045	1962.667684	175.0	453.549422	50745	25609	13.3	0.00000
3046	1962.667684	213.6	453.549422	41193	37030	13.9	0.00000

3047 rows × 34 columns

```
data2['avgdeath'].plot.box(figsize=(5,10))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf6313d650>
```



```
dum1=data2['avgdeath'].loc[data2.avgdeath>8000]
dum1
```

```
999     14010
2373     9445
Name: avgdeath, dtype: int64
```

```
data2.loc[999]
```

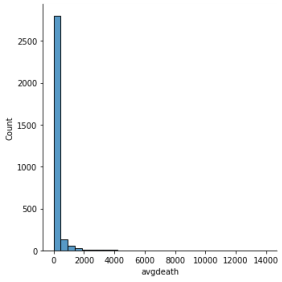
```
avganncount                                    38150.0
target_deathrate                                 148.4
incidencerate                                    405.5
medincome                                        55686
popest2015                                    10170292
povertypercent                                    18.7
studypercap                                  255.941521
binnedinc                          (54545.6, 61494.5]
medianage                                         35.6
medianagemale                                     34.4
medianagefemale                                   36.8
geography                 Los Angeles County, California
percentmarried                                    42.4
pctnohs18_24                                      15.3
pcths18_24                                        27.0
pctsomecol18_24                                   47.9
pctbachdeg18_24                                    9.9
pcths25_over                                      20.7
pctbachdeg25_over                                 19.8
pctemployed16_over                                58.0
pctunemployed16_over                              10.0
pctprivatecoverage                                55.0
pctprivatecoveragealone                           47.4
pctemprivcoverage                                 39.7
pctpubliccoverage                                 32.9
pctpubliccoveragealone                            23.0
pctwhite                                       53.25871
pctblack                                       8.27614
pctasian                                       14.12938
pctotherrace                                   19.591522
pctmarriedhouseholds                           44.58165
birthrate                                      4.705281
deathrate                                           148
avgdeath                                          14010
Name: 999, dtype: object
```

```
data2.loc[2373]
```

```
avganncount                              24965.0
target_deathrate                           177.0
incidencerate                              470.8
medincome                                  55058
popest2015                               5238216
povertypercent                              17.1
studypercap                            371.118717
binnedinc                    (54545.6, 61494.5]
medianage                                  35.9
medianagemale                              34.7
medianagefemale                            37.1
geography                 Cook County, Illinois
percentmarried                             41.9
pctnohs18_24                               14.9
pcths18_24                                 26.7
pctsomecol18_24                             NaN
pctbachdeg18_24                            15.5
pcths25_over                               24.0
pctbachdeg25_over                          21.5
pctemployed16_over                         59.0
pctunemployed16_over                       10.7
pctprivatecoverage                         61.5
pctprivatecoveragealone                     NaN
pctemprivcoverage                          46.3
pctpubliccoverage                          32.6
pctpubliccoveragealone                     22.0
pctwhite                                56.842582
pctblack                                23.982596
pctasian                                 6.77283
pctotherrace                            9.847733
pctmarriedhouseholds                    41.008791
birthrate                               4.994881
deathrate                                    177
avgdeath                                    9445
Name: 2373, dtype: object
```

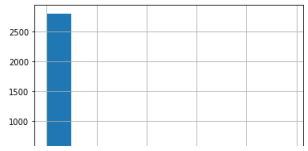
```
sns.displot(data2, x="avgdeath",bins=30)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fcf62ed98d0>
```



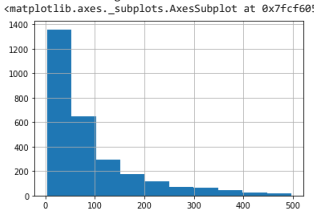

```
data3=data2.loc[data2.avgdeath<5000]
data3['avgdeath'].hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf605dbb50>
```



```
data3=data2.loc[data2.avgdeath<500]
print("less than 500 avarge count :",len(data3))
data3['avgdeath'].hist()
```

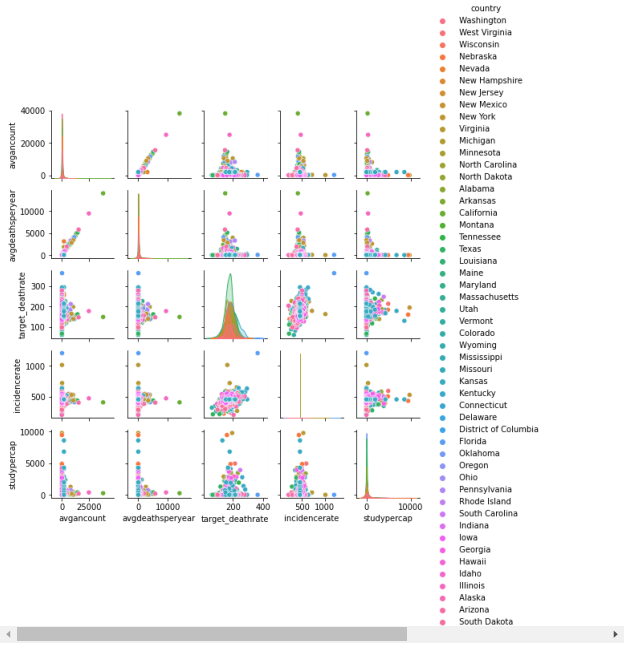
```
less than 500 avarge count : 2811
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf6051b390>
```



```
a_col = data2[['avganncount','avgdeath','target_deathrate','incidencerate','studypercap','country']]
```

```
sns.pairplot(a_col,hue='country',size=1.5);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:2076: UserWarning: The `size` parameter ha
  warnings.warn(msg, UserWarning)
```




```
data.geography
```

```
0          Kitsap County, Washington
1        Kittitas County, Washington
2       Klickitat County, Washington
3           Lewis County, Washington
4         Lincoln County, Washington
                   ...
3042        Ellsworth County, Kansas
3043           Finney County, Kansas
3044             Ford County, Kansas
3045         Franklin County, Kansas
3046            Geary County, Kansas
Name: geography, Length: 3047, dtype: object
```

```
data_copy = data2.copy
data2[['state','country']]=data2['geography'].str.split(',',expand=True)
data2.head()
```

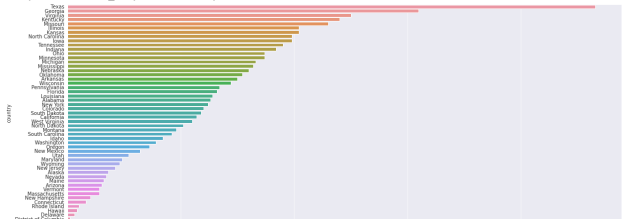
	avganncount	target_deathrate	incidencerate	medincome	popest2015	povertypercent	studypercap	l
0	1397.0	164.9	489.8	61898	260131	11.2	499.748204	
1	173.0	161.3	411.6	48127	43269	18.6	23.111234	
2	102.0	174.7	349.7	49348	21026	14.6	47.560164	
3	427.0	194.8	430.4	44243	75882	17.1	342.637253	
4	57.0	144.4	350.1	49955	10321	12.5	0.000000	

5 rows × 36 columns

```
data_contry = data2['country'].value_counts(ascending = False).index
```

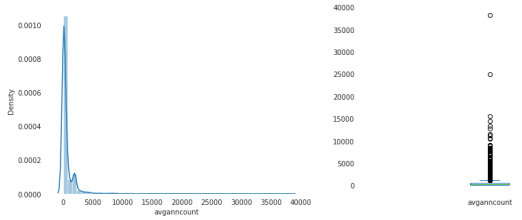
```
plt.figure(figsize=(20,8))
sns.set_style("darkgrid")
sns.countplot(data=data2,y='country',order=data_contry)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf5cbd61d0>
```



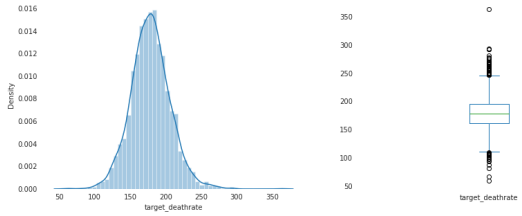
Independent Variable (Numerical)

```
plt.figure(1)
plt.subplot(121)
sns.distplot(data["avganncount"]);

plt.subplot(122)
data["avganncount"].plot.box(figsize=(15,5))
plt.show()
```

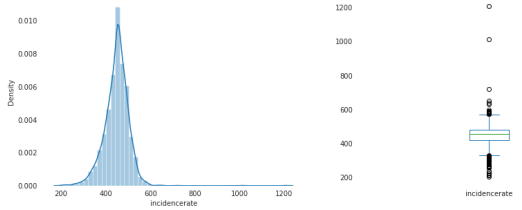


```
plt.figure(1)
plt.subplot(121)
sns.distplot(data["avganncount"]);

plt.subplot(122)
data["avganncount"].plot.box(figsize=(15,5))
plt.show()
```

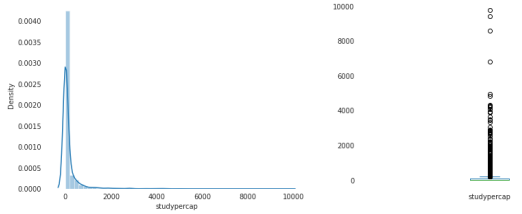
```
plt.figure(1)
plt.subplot(121)
sns.distplot(data["target_deathrate"]);

plt.subplot(122)
data["target_deathrate"].plot.box(figsize=(15,5))
plt.show()
```



```
plt.figure(1)
plt.subplot(121)
sns.distplot(data["incidencerate"]);

plt.subplot(122)
data["incidencerate"].plot.box(figsize=(15,5))
plt.show()
```



```
plt.figure(1)
plt.subplot(121)
sns.distplot(data["studypercap"]);

plt.subplot(122)
data["studypercap"].plot.box(figsize=(15,5))
plt.show()
```



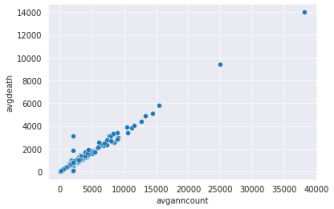
```
data2.corr()
```

	avganncount	target_deathrate	incidencerate	medincome	popest2015	povert
avganncount	1.000000	-0.143532	0.073553	0.269145	0.926894	
target_deathrate	-0.143532	1.000000	0.449432	-0.428615	-0.120073	
incidencerate	0.073553	0.449432	1.000000	-0.001036	0.026912	
medincome	0.269145	-0.428615	-0.001036	1.000000	0.235523	
popest2015	0.926894	-0.120073	0.026912	0.235523	1.000000	
povertypercent	-0.135694	0.429389	0.009046	-0.788965	-0.065299	
studypercap	0.082071	-0.022285	0.077283	0.044003	0.055722	
medianage	-0.024098	0.004375	0.018089	-0.013288	-0.025219	
medianagemale	-0.124969	-0.021929	-0.014733	-0.091663	-0.176608	
medianagefemale	-0.122844	0.012048	-0.009106	-0.153278	-0.177932	
percentmarried	-0.106108	-0.266820	-0.119524	0.355123	-0.160463	
pctnohs18_24	-0.143327	0.088463	-0.170762	-0.289383	-0.126582	
pcths18_24	-0.182054	0.261976	0.022644	-0.190006	-0.151821	
pctsomecol18_24	0.109455	-0.188688	0.077666	0.212953	0.093202	
pctbachdeg18_24	0.284176	-0.287817	0.046835	0.492810	0.248375	
pcths25_over	-0.311375	0.404589	0.121725	-0.471348	-0.311849	
pctbachdeg25_over	0.321021	-0.485477	-0.038177	0.704928	0.297463	
pctemployed16_over	0.199459	-0.412046	0.004906	0.693432	0.140146	
pctunemployed16_over	-0.009016	0.378412	0.099979	-0.453108	0.050768	
pctprivatecoverage	0.132244	-0.386066	0.105174	0.724175	0.052677	
pctprivatecoveragealone	0.186045	-0.363704	0.109278	0.788048	0.132660	
pctemprivcoverage	0.202349	-0.267399	0.149825	0.747294	0.158650	
pctpubliccoverage	-0.173548	0.404572	0.046109	-0.754822	-0.160066	
pctpubliccoveragealone	-0.093699	0.449358	0.040812	-0.719756	-0.041469	
pctwhite	-0.136501	-0.177400	-0.014510	0.167225	-0.190095	
pctblack	0.031376	0.257024	0.113489	-0.270232	0.073044	
pctasian	0.435071	-0.186331	-0.008123	0.425844	0.464168	
pctotherrace	0.209184	-0.189894	-0.208748	0.083635	0.241468	
pctmarriedhouseholds	-0.106221	-0.293325	-0.152176	0.446083	-0.127979	
birthrate	-0.034508	-0.087407	-0.118181	-0.010195	-0.057740	

```
final = data2[['avganncount','avgdeath','target_deathrate','incidencerate','studypercap']]
```

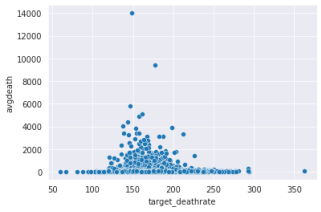
```
sns.scatterplot(data=final, x="avganncount", y="avgdeath")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf57fa70d0>
```



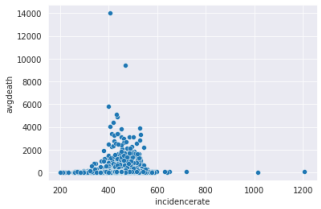

```
sns.scatterplot(data=final, x="target_deathrate", y="avgdeath")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf57f767d0>
```



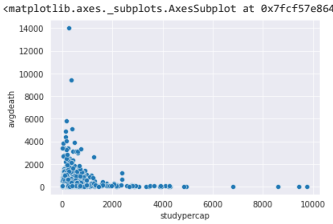

```
sns.scatterplot(data=final, x="incidencerate", y="avgdeath")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf57f0b090>
```




```
sns.scatterplot(data=final, x="studypercap", y="avgdeath")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf57e864d0>
```



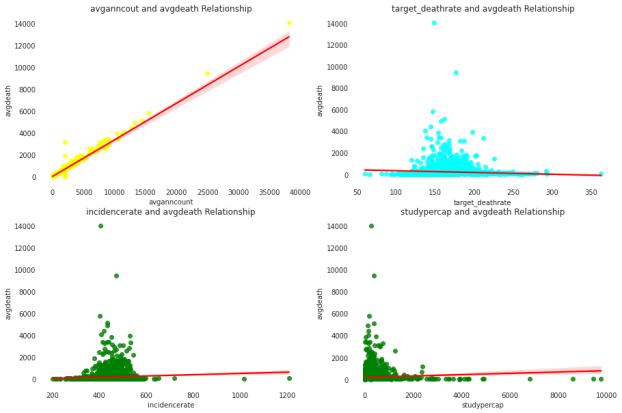
Assumptions for Linear Regression

```
import statsmodels.api as sm

X = data2[['avganncount','target_deathrate','incidencerate','studypercap']]
Y = data2['avgdeath']

# Defining the modell
X = sm.add_constant(X)
model = sm.OLS(Y, X)

# Fitting the model
result = model.fit()

# Printing the model summary
print(result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               avgdeath   R-squared:                       0.886
Model:                            OLS   Adj. R-squared:                  0.885
Method:                 Least Squares   F-statistic:                     5883.
Date:                Mon, 19 Sep 2022   Prob (F-statistic):               0.00
Time:                        16:15:11   Log-Likelihood:                -19982.
No. Observations:                3047   AIC:                         3.997e+04
Df Residuals:                    3042   BIC:                         4.000e+04
Df Model:                           4
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const           -75.1640     27.342     -2.749      0.006    -128.776     -21.552
avganncount       0.3387      0.002    151.295      0.000       0.334       0.343
target_deathrate  1.1047      0.127      8.669      0.000       0.855       1.355
incidencerate    -0.3123      0.064     -4.844      0.000      -0.439      -0.186
studypercap      -0.0101      0.006     -1.721      0.085      -0.022       0.001
==============================================================================
Omnibus:                     1249.435   Durbin-Watson:                   0.592
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            70380.791
Skew:                          -1.146   Prob(JB):                         0.00
Kurtosis:                      26.433   Cond. No.                     1.38e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.38e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
plt.rcParams["figure.figsize"] = (15,10)

plt.subplot(2, 2, 1)
plt.title("avganncout and avgdeath Relationship")
sns.regplot(X['avganncount'], Y, scatter_kws={"color": "yellow"}, line_kws={"color": "red"})

plt.subplot(2, 2, 2)
plt.title("target_deathrate and avgdeath Relationship")
sns.regplot(X['target_deathrate'], Y, scatter_kws={"color": "cyan"}, line_kws={"color": "red"})

plt.subplot(2, 2, 3)
plt.title("incidencerate and avgdeath Relationship")
sns.regplot(X['incidencerate'], Y, scatter_kws={"color": "green"}, line_kws={"color": "red"})

plt.subplot(2, 2, 4)
plt.title("studypercap and avgdeath Relationship")
sns.regplot(X['studypercap'], Y, scatter_kws={"color": "green"}, line_kws={"color": "red"})
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf4ca67490>
```



Assumption 2: No or Less Multicollinearity

Formula to calculate variance inflation factor

$$VIF = 1/(1-r^2) = 1/Tolerance$$

```
# Library for checking multicollinearity
from statsmodels.stats.outliers_influence import variance_inflation_factor

import warnings
warnings.filterwarnings('ignore')

def calculate_vif(dataset):
    vif = pd.DataFrame()
    vif['features']=dataset.columns
    vif["vif_value"]=[variance_inflation_factor(dataset.values, i) for i in range(dataset.shape[1])]
    return vif

features = X
calculate_vif(features)
```

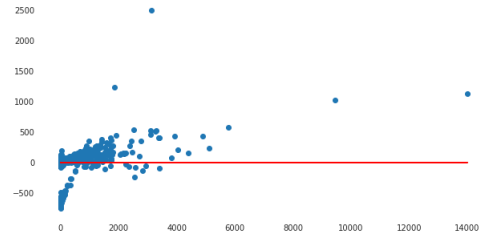
	features	vif_value
0	const	78.198323
1	avganncount	1.051077
2	target_deathrate	1.307622
3	incidencerate	1.294153
4	studypercap	1.014510

Assumption 3: Homoskedasticity - Constant Variance

- List item
- List item

```
residuals = result.resid
plt.rcParams['figure.figsize']=(10, 5)
plt.scatter(Y, residuals)
plt.plot(Y, [0]*len(Y), color='red')
```

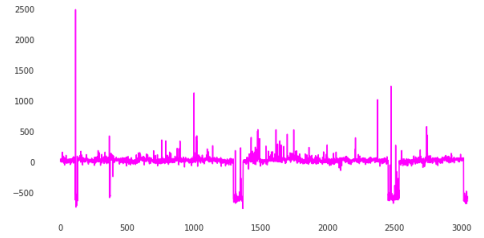
```
[<matplotlib.lines.Line2D at 0x7fcf4d03ddd0>]
```



Assumption 4 : No Autocorrelation of errors

```
plt.rcParams['figure.figsize']=(10,5)
plt.rcParams['axes.facecolor'] = 'white'
plt.plot(residuals.index, residuals.values, color='magenta')
```

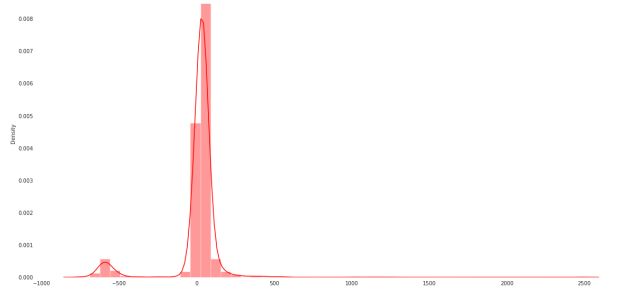
```
[<matplotlib.lines.Line2D at 0x7fcf4cb858d0>]
```



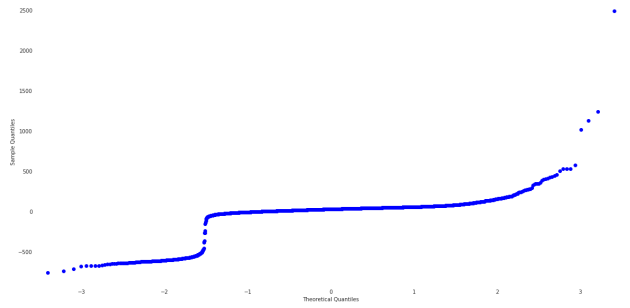
Assumption 5: Residual Normality

```
sns.distplot(residuals, color='red')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code t

<matplotlib.axes._subplots.AxesSubplot at 0x7fcf4d3d0a50>
```



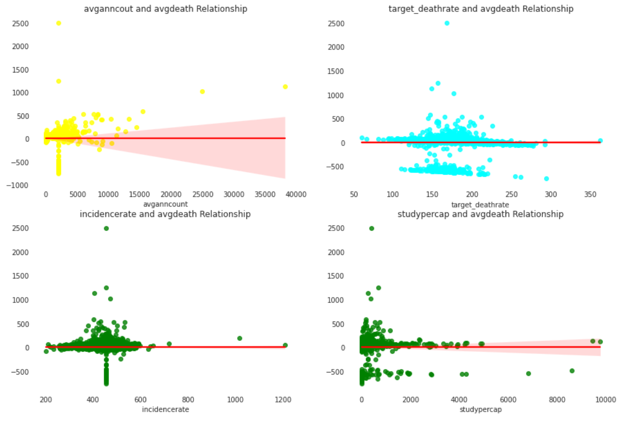
```
sm.qqplot(residuals)
plt.show()
```



Assumption 6 : Residual relation with independent variables

```
plt.rcParams["figure.figsize"] = (15,10)

plt.subplot(2, 2, 1)
plt.title("avganncout and avgdeath Relationship")
sns.regplot(X['avganncount'], residuals, scatter_kws={"color": "yellow"}, line_kws={"color": "red"})

plt.subplot(2, 2, 2)
plt.title("target_deathrate and avgdeath Relationship")
sns.regplot(X['target_deathrate'], residuals, scatter_kws={"color": "cyan"}, line_kws={"color": "red"})

plt.subplot(2, 2, 3)
plt.title("incidencerate and avgdeath Relationship")
sns.regplot(X['incidencerate'], residuals, scatter_kws={"color": "green"}, line_kws={"color": "red"})

plt.subplot(2, 2, 4)
plt.title("studypercap and avgdeath Relationship")
sns.regplot(X['studypercap'], residuals, scatter_kws={"color": "green"}, line_kws={"color": "red"})
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argu

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argu

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argu

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argu

<matplotlib.axes._subplots.AxesSubplot at 0x7fcf4d165b90>
```




```
X.head()
```

	const	avganncount	target_deathrate	incidencerate	studypercap
0	1.0	1397.0	164.9	489.8	499.748204
1	1.0	173.0	161.3	411.6	23.111234
2	1.0	102.0	174.7	349.7	47.560164
3	1.0	427.0	194.8	430.4	342.637253
4	1.0	57.0	144.4	350.1	0.000000

```
Y.head()
```

```
0    469
1     70
2     50
3    202
4     26
Name: avgdeath, dtype: int64
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=10)
```

```
from sklearn.linear_model import LinearRegression
lr_clf = LinearRegression()
lr_clf.fit(X_train,y_train)
lr_clf.score(X_test,y_test)
```

```
0.9038004299620067
```

Using K Fold Cross Validation to measure accuracy of our LinearRegression Model

```
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score
cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
cross_val_score(LinearRegression(),X,y,cv=cv)
```

```
array([0.92931238, 0.77437374, 0.71255449, 0.65469105, 0.75541814])
```