```
# libraries
import numpy as np
import pandas as pd
import altair as alt
# load tidied data and print rows
data=pd.read_csv('covidprisondata.csv')
data
```

```
#creating new variables for analysis
data['Inmate Positivity Rate']= data['Total Inmate Cases']/data['Inmate Tests']
data['Inmate Death Rate']=data['Total Inmate Deaths']/ data['Total Inmate Cases']
data['Officer Death Rate']=data['Total Officer Deaths']/data['Total Officer Cases']
data['Prison Density']=data['Latest Inmate Population']/data['Max Inmate Population 2020']
data['Total Death Rate']=data['Total Deaths']/data['Total Infections']
data['Differences Between (Total - Inmate) Death Rates']= data['Total Death Rate']- data['Inmate Death Rate']
```

```
#selecting columns for Part 1 Analysis
Part1data=data.iloc[:,[0,9,10,21,22,23,24,25]]
Part1data
```

```
#deleting columns with missing/0 for Prison Density, inmate death rate vs prison density chart
data1=data.drop(data.index[[27]])
plot1=alt.Chart(data1).mark_circle().encode(
   x = alt.X('Prison Density', scale = alt.Scale(zero = False)),
    y = alt.Y('Inmate Death Rate', scale = alt.Scale(type = 'sqrt')),
)
plot1
```

```
#deleting columns with missing/0 for Prison Density, inmate death rate vs prison density chart
data1=data.drop(data.index[[27]])
plot1=alt.Chart(data1).mark_circle().encode(
   x = alt.X('Prison Density', scale = alt.Scale(zero = False)),
    y = alt.Y('Inmate Death Rate', scale = alt.Scale(type = 'sqrt')),
)
```

```
#LOESS
loess = plot1.transform_loess(
    on = 'Prison Density', # x variable
    loess = 'Inmate Death Rate', # y variable
    bandwidth = 0.2 # how smooth?
).mark_line(color = 'black')

#regression
reg = plot1.transform_regression(
    on = 'Prison Density', # x variable
    regression = 'Inmate Death Rate', # y variable
).mark_line(color = 'red')


plot1 + reg + loess #combine plots
```

```
#deleting columns with missing/0 for Prison Density, inmate death rate vs prison density chart
data1=data.drop(data.index[[27]])
plot1=alt.Chart(data1).mark_circle().encode(
   x = alt.X('Officer Death Rate', scale = alt.Scale(zero = False)),
    y = alt.Y('Inmate Death Rate', scale = alt.Scale(type = 'log')),
)



#LOESS
loess = plot1.transform_loess(
    on = 'Officer Death Rate', # x variable
    loess = 'Inmate Death Rate', # y variable
    bandwidth = 0.2 # how smooth?
).mark_line(color = 'black')

#regression
reg = plot1.transform_regression(
    on = 'Officer Death Rate', # x variable
    regression = 'Inmate Death Rate', # y variable
).mark_line(color = 'red')


plot1 + reg + loess #combine plots
```

```
data1.boxplot(column="Differences Between (Total - Inmate) Death Rates")
```

#PART 2

```
#deleting columns with missing/0 for Prison Density, inmate death rate vs prison density
chart
data1=data.drop(data.index[[27]])
plot2=alt.Chart(data1).mark_circle().encode(
   x = 'Minority',
    y = 'Inmate Positivity Rate',
).properties(
    width = 250, height = 250)

#LOESS
loess2 = plot2.transform_loess(
    on = 'Minority', # x variable
    loess = 'Inmate Positivity Rate', # y variable
    bandwidth = 0.2 # how smooth?
).mark_line(color = 'black')

#regression
reg2 = plot2.transform_regression(
    on = 'Minority', # x variable
    regression = 'Inmate Positivity Rate', # y variable
).mark_line(color = 'red')

data2=data.drop(data.index[[54]])
plot3=alt.Chart(data1).mark_circle().encode(
   x = alt.X('Poverty Rate', scale = alt.Scale(zero = False)),
```

```
    y = alt.Y('Inmate Positivity Rate'),
).properties(
    width = 250, height = 250)

#LOESS
loess3 = plot3.transform_loess(
    on = 'Poverty Rate', # x variable
    loess = 'Inmate Positivity Rate', # y variable
    bandwidth = 0.2 # how smooth?
).mark_line(color = 'black')

#regression
reg3 =plot3.transform_regression(
    on = 'Poverty Rate', # x variable
    regression = 'Inmate Positivity Rate', # y variable
).mark_line(color = 'red')
plot2 + loess2 + reg2 |plot3 + reg3 + loess3 #combine plots
```

```
data['Crude Infection Rate'] = data['Male crude rate (cases)'] + data['Female crude rate
(cases)']
data['Crude Death Rate'] = data['Male crude rate (deaths)'] + data['Female crude rate
(deaths)\n']

data3=data.drop(data.index[[20,23]])
plot4=alt.Chart(data3).mark_circle().encode(
   x = alt.X('Poverty Rate', scale = alt.Scale(zero = False)),
    y = alt.Y('Crude Infection Rate'),
).properties(
    width = 250, height = 250)



#LOESS
loess4 = plot4.transform_loess(
    on = 'Poverty Rate', # x variable
    loess = 'Crude Infection Rate', # y variable
    bandwidth = 0.2 # how smooth?
).mark_line(color = 'black')
```

```
#regression
reg4 =plot4.transform_regression(
    on = 'Poverty Rate', # x variable
    regression = 'Crude Infection Rate', # y variable
).mark_line(color = 'red')



data4=data.drop(data.index[[3,8,23,30,33,47]])

plot5=alt.Chart(data4).mark_circle().encode(
   x = alt.X('Poverty Rate', scale = alt.Scale(zero = False)),
    y = alt.Y('Crude Death Rate'),
).properties(
    width = 250, height = 250)



#LOESS
loess5 = plot5.transform_loess(
    on = 'Poverty Rate', # x variable
    loess = 'Crude Death Rate', # y variable
    bandwidth = 0.2 # how smooth?
).mark_line(color = 'black')

#regression
reg5=plot5.transform_regression(
    on = 'Poverty Rate', # x variable
    regression = 'Crude Death Rate', # y variable
).mark_line(color = 'red')



plot4 + reg4 + loess4 |  plot5 + reg5 + loess5
```

```
data3=data.drop(data.index[[20,23]])
plot6=alt.Chart(data3).mark_circle().encode(
   x = alt.X('Minority', scale = alt.Scale(zero = False)),
```

```
    y = alt.Y('Crude Infection Rate'),
).properties(
    width = 300, height = 200)



#LOESS
loess6 = plot6.transform_loess(
    on = 'Minority', # x variable
    loess = 'Crude Infection Rate', # y variable
    bandwidth = 0.2 # how smooth?
).mark_line(color = 'black')

#regression
reg6=plot6.transform_regression(
    on = 'Minority', # x variable
    regression = 'Crude Infection Rate', # y variable
).mark_line(color = 'red')



plot7=alt.Chart(data4).mark_circle().encode(
   x = alt.X('Minority', scale = alt.Scale(zero = False)),
    y = alt.Y('Crude Death Rate'),
).properties(
    width = 300, height = 200)



#LOESS
loess7 = plot7.transform_loess(
    on = 'Minority', # x variable
    loess = 'Crude Death Rate', # y variable
    bandwidth = 0.2 # how smooth?
).mark_line(color = 'black')

#regression
reg7=plot7.transform_regression(
    on = 'Minority', # x variable
    regression = 'Crude Death Rate', # y variable
```

```
).mark_line(color = 'red')



plot6 + reg6 + loess6 |  plot7 + reg7 + loess7
```