Introduction

Group Functions

```
%matplotlib inline

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tabulate import tabulate
import scipy.stats as stats
from scipy import optimize
import seaborn as sns
sns.set(style="whitegrid")
import sqlite3
import statsmodels.api as sm

chartColor = "DimGray"
chartAlpha=0.5
```

```
def restyle_boxplot(patch):
    ## change color and linewidth of the whiskers
    for whisker in patch['whiskers']:
        whisker.set(color='#000000', linewidth=1)
    ## change color and linewidth of the caps
    for cap in patch['caps']:
        cap.set(color='#000000', linewidth=1)
    ## change color and linewidth of the medians
    for median in patch['medians']:
        median.set(color='#000000', linewidth=2)
    ## change the style of fliers and their fill
    for flier in patch['fliers']:
        flier.set(marker='o', color='#000000', alpha=0.0)
    for box in patch["boxes"]:
        box.set( facecolor='#FFFFFF', alpha=0.5)
```

Sloan Cinkle
Aug 8, 2022

set flier opacity to zero

```
def freedman_diaconis(data):
    q = stats.mstats.mquantiles(data, [0.25, 0.75])
    h = 2.0 * (q[1] - q[0]) / len(data) ** (1/3)
    bins = np.arange(data.min(), data.max() + h, h).tolist()
    return bins
```

Sloan Cinkle
Aug 8, 2022

used np.arange to create bins

```
def histChart(axes,data,isDensity=True,in_bins=None,in_xlim=None):
    if in_bins is None:
        in_bins = freedman_diaconis(data)
    axes.hist(data, bins=in_bins,color=chartColor, density=isDensity, alpha=chartAlpha)
    axes.set_ylabel("Density" if isDensity else "Frequency" )
    axes.set_xlabel(f"{data.name} Intervals")
    axes.set_title(f"Histogram of {data.name}")
    axes.xaxis.grid(False)
    axes.set_xlim(in_xlim)
```

```
def boxPlot(axes,data):
    patch = axes.boxplot(data, labels=[''], showfliers=True, patch_artist=True, zorder=1)
    restyle_boxplot(patch)
    axes.set_title(f"Distribution of {data.name}")
    axes.set_ylabel(f"{data.name}")
    x = np.random.normal(1, 0.001, size=len(data))
    axes.plot(x, data, 'o', alpha=0.4, color=chartColor, zorder=2)
```

```
def histAndBox(data,in_bins=None):
    fig = plt.figure(figsize=(20,6))
    axes = fig.add_subplot(1,2,1)
    histChart(axes,data,in_bins=in_bins)
    axes = fig.add_subplot(1,2,2)
    boxPlot(axes,data)
    plt.show()
    plt.close()
```

```
def cdfCompare(data,distData,distType):
    figure = plt.figure(figsize=(20, 8))
    ticket_mn = np.min(data)
    ticket_mx = np.max(data)

    axes = figure.add_subplot(1, 2, 1)
    values, base = np.histogram(data, bins=20, density=True)
    cumulative = np.cumsum(values)
    axes.plot(base[:-1], cumulative, color="steelblue")
    axes.set_xlim((ticket_mn, ticket_mx))

    values2, base = np.histogram(distData, bins=base, density=True)

    cumulative2 = np.cumsum(values2)
    axes.plot( base[:-1], cumulative2, color="firebrick")
    axes.set_xlim((ticket_mn, ticket_mx))
    axes.set_xlabel(f"Empirical v. Theoretical: {distType}")

    axes = figure.add_subplot(1, 2, 2)

    differences = cumulative2 - cumulative
    axes.plot(base[:-1], differences, color='firebrick')
    axes.set_xlim((ticket_mn, ticket_mx))
    axes.hlines(0, 0, 14000, linestyles="dotted")
    axes.set_xlabel(f"Empirical v. Theoretical: {distType} Distribution, Difference")

    plt.show()
    plt.close()
```

```
def correlation(data, x, y):
    print("Correlation coefficients:")
    print( "r =", stats.pearsonr(data[x], data[y])[0])
    print( "rho =", stats.spearmanr(data[x], data[y])[0])
```

```
def lowess_scatter(data, x, y, jitter=0.0, skip_lowess=False):
    if skip_lowess:
        fit = np.polyfit(data[x], data[y], 1)
        line_x = np.linspace(data[x].min(), data[x].max(), 10)
        line = np.poly1d(fit)
        line_y = list(map(line, line_x))
    else:
        lowess = sm.nonparametric.lowess(data[y], data[x], frac=.3)
        line_x = list(zip(*lowess))[0]
        line_y = list(zip(*lowess))[1]

    figure = plt.figure(figsize=(10, 6))
    axes = figure.add_subplot(1, 1, 1)
    xs = data[x]
    if jitter > 0.0:
        xs = data[x] + stats.norm.rvs( 0, 0.5, data[x].size)
    axes.scatter(xs, data[y], marker="o", color="DimGray", alpha=0.5)
    axes.plot(line_x, line_y, color="DarkRed")

    title = "Plot of {0} v. {1}".format(x, y)
    if not skip_lowess:
        title += " with LOWESS"
    axes.set_title(title)
    axes.set_xlabel(x)
    axes.set_ylabel(y)
    plt.show()
    plt.close()
```

Sloan Cinkle
Aug 8, 2022

changed to always show linear relationship

Reading the Data

```
#@title
# import PS6_Group4_ETL as etl
# etl.createCOVIDDB()
conn = sqlite3.connect('./data/PS6_Group4.db')
covid = pd.read_sql_query("SELECT * from COVID_DATA", conn)
conn.close()
covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1961 entries, 0 to 1960
Data columns (total 40 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   county_fips                             1961 non-null   int64
 1   covid_19_deaths                         1961 non-null   float64
 2   percent_of_smokers                      1961 non-null   float64
 3   percent_of_diabetes                     1961 non-null   float64
 4   median_household_income                 1961 non-null   int64
 5   less_than_high_school_diploma           1961 non-null   float64
 6   high_school_diploma_only                1961 non-null   float64
 7   some_college_or_higher                  1961 non-null   float64
 8   population_density                      1961 non-null   float64
 9   social_distancing_total_grade           1961 non-null   object
 10  social_distancing_visitation_grade      1961 non-null   object
 11  social_distancing_encounters_grade      1961 non-null   object
 12  social_distancing_travel_distance_grade 1961 non-null   object
 13  social_distance_gpa                     1961 non-null   float64
 14  social_distance_gpa_visitation          1961 non-null   float64
 15  social_distance_gpa_encounters          1961 non-null   float64
 16  social_distance_gpa_travel              1961 non-null   float64
 17  percent_of_vaccinated_residents         1961 non-null   float64
 18  age_0_4                                 1961 non-null   int64
 19  age_5_9                                 1961 non-null   int64
 20  age_10_14                               1961 non-null   int64
 21  age_15_19                               1961 non-null   int64
 22  age_20_24                               1961 non-null   int64
 23  age_25_29                               1961 non-null   int64
 24  age_30_34                               1961 non-null   int64
 25  age_35_39                               1961 non-null   int64
 26  age_40_44                               1961 non-null   int64
 27  age_45_49                               1961 non-null   int64
 28  age_50_54                               1961 non-null   int64
 29  age_55_59                               1961 non-null   int64
 30  age_60_64                               1961 non-null   int64
 31  age_65_69                               1961 non-null   int64
 32  age_70_74                               1961 non-null   int64
 33  age_75_79                               1961 non-null   int64
 34  age_80_84                               1961 non-null   int64
 35  age_85_or_higher                        1961 non-null   int64
 36  hospital_beds_ratio                     1961 non-null   float64
 37  ventilator_capacity_ratio               1961 non-null   float64
 38  intensive_care_unit_ICU_bed_ratio       1961 non-null   float64
 39  total_population                        1961 non-null   int64
dtypes: float64(15), int64(21), object(4)
memory usage: 612.9+ KB
```

```
covid["covid_19_deaths_per_100k"] = covid.covid_19_deaths / \
    covid.total_population * 100000
covid.covid_19_deaths_per_100k.describe()
```

```
count    1961.000000
mean      192.161417
std        96.581790
min         0.000000
25%       123.053876
50%       183.113969
75%       247.360260
max       820.152314
Name: covid_19_deaths_per_100k, dtype: float64
```

Single Variable EDA

[ ] ↳ *31 cells hidden*

Pairwise EDA

Median Household Income v. COVID-19 Deaths per Capita

[ ] ↳ *9 cells hidden*

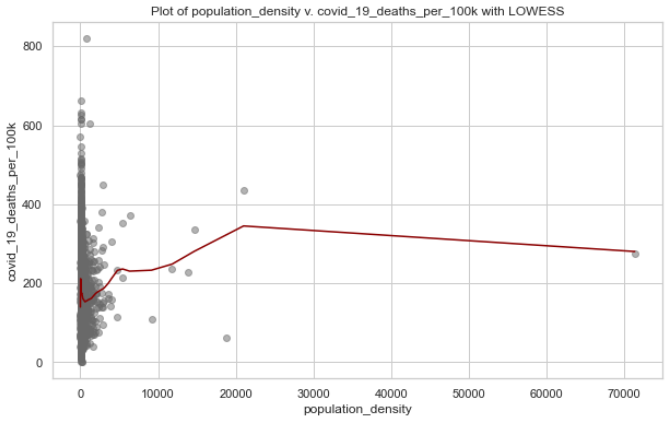
Education Level Distribution v. COVID-19 Deaths per Capita

[ ] ↳ *4 cells hidden*

Population Density v. COVID-19 Deaths per Capita

Let's see a scatterplot without using the log transformation.

```
lowess_scatter(covid, "population_density", "covid_19_deaths_per_100k")
```



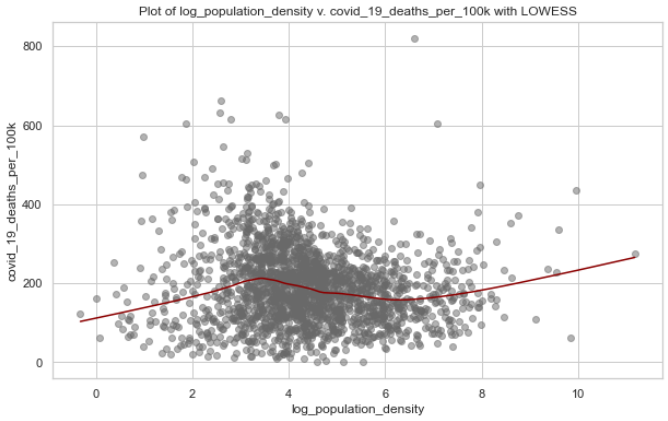
All the points are jumbled on the left side of the graph, so we should probably apply a transformation to the data. Let's see what the correlation is.

```
correlation(covid, "population_density", "covid_19_deaths_per_100k")
```

```
    Correlation coefficients:
    r = 0.01950303281564219
    rho = -0.13327406127861594
```

This is a very low correlation, and we have differing signs for linear and non-linear relationships. Let's apply the log transformation.

```
lowess_scatter(covid, "log_population_density", "covid_19_deaths_per_100k")
```



This looks much better than before. Now there is a more even spread of observations and more data in the center; however, I don't see a definite positive or negative slope to either line.

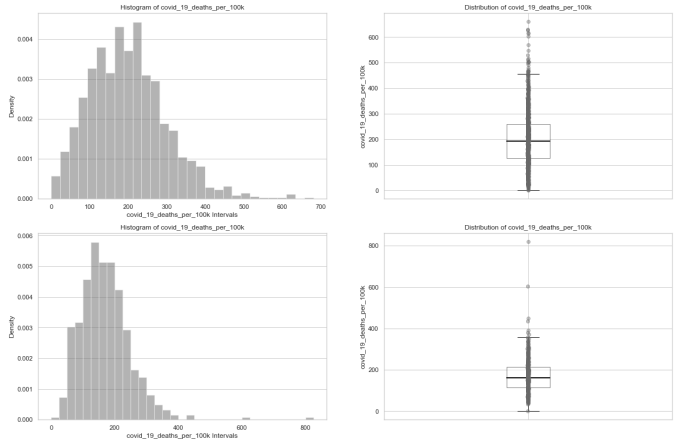
```
correlation(covid, "log_population_density", "covid_19_deaths_per_100k")
```

```
Correlation coefficients:
r = -0.11749861151038053
rho = -0.13327406127861594
```

This is a stronger correlation than before and now definitely negative, but it is still a very weak correlation. I am not sure if this variable will provide much predictive power to the model.

We can visualize COVID-19 deaths per capita for high-density and low-density counties separately to see if an indicator variable might be more useful.

```
histAndBox(covid[covid.high_density == 0].covid_19_deaths_per_100k)
histAndBox(covid[covid.high_density == 1].covid_19_deaths_per_100k)
```



On top is the distribution of COVID-19 deaths per capita for low-density counties, and on bottom is that for high-density counties. The distribution of COVID-19 deaths per capita looks skewed to the right for both types of counties; however, the maximum deaths per capita is higher for high-density counties.

```
correlation(covid, "high_density", "covid_19_deaths_per_100k")
```

```
Correlation coefficients:
r = -0.1286790117424592
rho = -0.13143550918972374
```

The correlation between our indicator variable and the target variable is weaker than what we had between our log transformation and the target.

I would not recommend including *population_density* in the initial model. The relationship between this variable and COVID-19 deaths per capita is not clear, even after applying transformations. If we find that

there is some unexplained trend in the residual plots after our initial model, we might want to consider including interaction effects between other variables and $log(population\_density)$ or the high-density indicator.

Interactions between Quantitative Variables

```
from matplotlib import cm
```

I'm not sure if the professor has a function like this for plotting interaction effects between 2 quantitative variables. I'm plotting a scatterplot of `x` vs. `y` and coloring them by `c`. The data is partitioned into `g` groups and I'm plotting lines with a matching color to represent the partition. I'm using the "coolwarm" color map from matplotlib since it was the most visible one I could find for a white background.

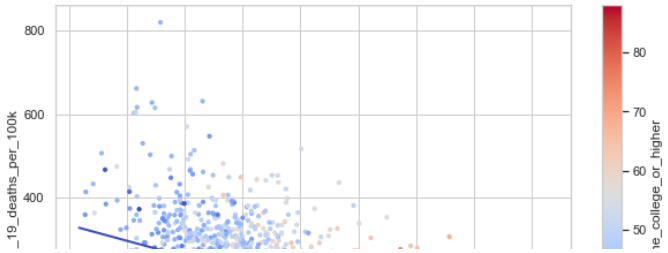
```
def interaction_plot(data, x, y, c, g=2, size=10, width=2, cmap="coolwarm"):
    figure = plt.figure(figsize=(10, 6))
    axes = figure.add_subplot(1, 1, 1)

    p = axes.scatter(data[x], data[y], c=data[c], s=size, cmap=cmap)
    figure.colorbar(p, label=c)

    intervals = np.linspace(0, 1, g+1)
    quantiles = stats.mstats.mquantiles(data[c], intervals)
    grouped_data = []
    for i in range(len(quantiles)-1):
      group = data[quantiles[i] <= data[c]]
      group = group[group[c] <= quantiles[i+1]]
      grouped_data.append(group)

    line_colors = cm.get_cmap(cmap, g)(np.linspace(0, 1, g))
    for i in range(g):
      fit = np.polyfit(grouped_data[i][x], grouped_data[i][y], 1)
      line_x = np.linspace(data[x].min(), data[x].max(), 10)
      line = np.poly1d(fit)
      line_y = list(map(line, line_x))

      label = "[{0}, {1}]:".format(round(quantiles[i], 3),
                                   round(quantiles[i+1], 3))
      print(label, "m = {0}".format(fit[0]))
      axes.plot(line_x, line_y, linewidth=width, color=line_colors[i])

    axes.plot()

    axes.set_xlabel(x)
    axes.set_ylabel(y)

    plt.show()
    plt.close()
```

Median Household Income and Education Level on COVID-19 Deaths per Capita

The more income a household has, the more money they can spend on COVID-19 preventative measures and treatments. I expect that people with a higher level of education have better research on the virus and can make each dollar worth more in terms of prevention and treatment.

```
interaction_plot(covid, "log_mhi", "covid_19_deaths_per_100k",
                 "some_college_or_higher")
```

```
[24.4, 53.7]: m = -189.25548167708033
[53.7, 88.0]: m = -91.5735850143364
```



It looks like median household income has a different effect on COVID-19 deaths per capita depending on the proportion of educated people in the county.

There is a more positive effect for counties with lots of college attendees and a more negative effect for counties with fewer college attendees, which is the opposite of what I expected.

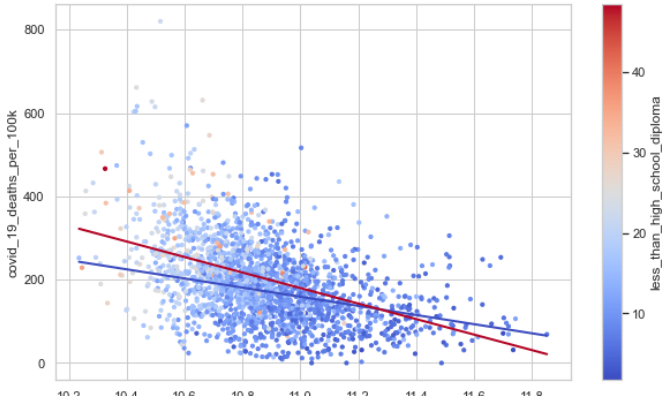
```
correlation(covid, "log_mhi", "some_college_or_higher")
```

```
Correlation coefficients:
r = 0.6970484062946982
rho = 0.6771208901617789
```

Median household income and the proportion of people with a college education have a moderate positive correlation. We can see that the points on the scatterplot gradually change color as median household income increases.

```
interaction_plot(covid, "log_mhi", "covid_19_deaths_per_100k",
                 "less_than_high_school_diploma")
```

```
[1.7, 11.8]: m = -109.61985774802164
[11.8, 48.5]: m = -186.09201177634256
```



There is an opposite interaction between median household income and the proportion of the county that did not graduate high school than attended college.

```
correlation(covid, "median_household_income", "less_than_high_school_diploma")
```

```
Correlation coefficients:
r = -0.5370252993236934
rho = -0.6469715087189807
```

Median household income is less correlated with the percent of people without a high school diploma than with the percent of people who have attended college.

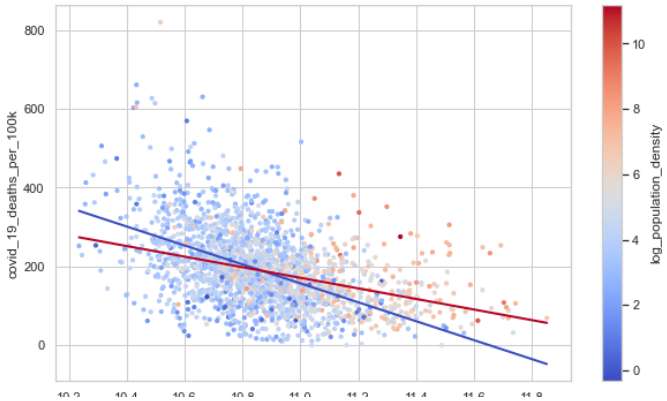
We should start with the interaction effect between median household income and less than high school diploma before checking if the interaction with some college or higher adds any predictive power.

Median Household Income and Population Density on COVID-19 Deaths per Capita

I chose not to include population density in the pairwise EDA, but I want to see if it might be useful in interactions. I think that median household income might have an interaction effect with population density on COVID-19 deaths per capita, since total income for the community increases with median income and with population density.

```
interaction_plot(covid, "log_mhi", "covid_19_deaths_per_100k",
                 "log_population_density")
```

```
[-0.324, 4.239]: m = -240.52853163036355
[4.239, 11.175]: m = -134.3580855619706
```

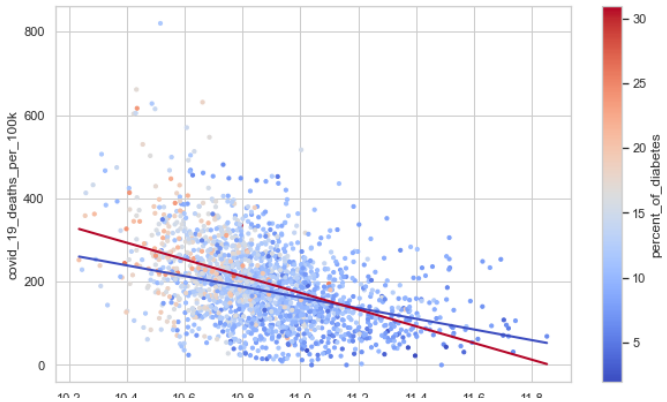


It looks like median household income has a more negative effect on COVID-19 deaths per capita for counties with low population density.

Median Household Income and Percent of Diabetes on COVID-19 Deaths per Capita

```
interaction_plot(covid, "log_mhi", "covid_19_deaths_per_100k",
                 "percent_of_diabetes")
```

```
[1.9, 11.5]: m = -128.15719465204475
[11.5, 31.0]: m = -200.48085587185668
```

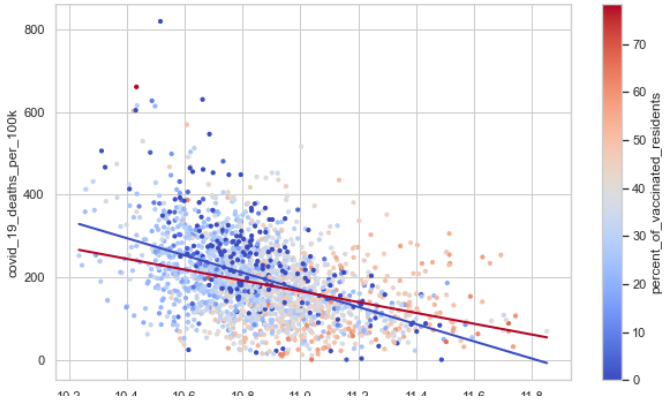


Median household income has a more negative effect on COVID-19 deaths per capita for counties with a higher percent of diabetes.

Median Household Income and Percent of Vaccinated Residents on COVID-19 Deaths per Capita

```
interaction_plot(covid, "log_mhi", "covid_19_deaths_per_100k",
                 "percent_of_vaccinated_residents")
```

```
[0.0, 31.7]: m = -208.40323182219905
[31.7, 78.3]: m = -131.0972879258017
```



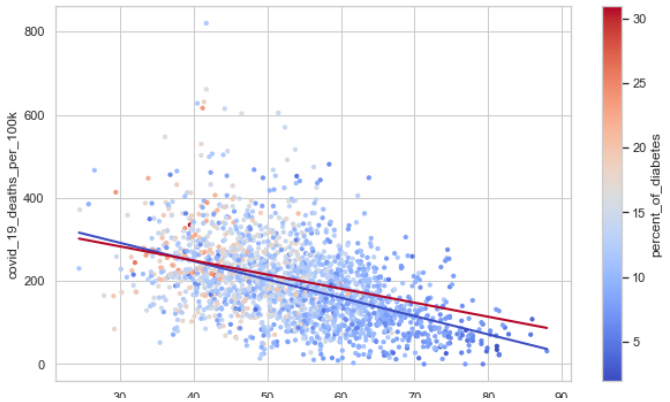
Median household income has a more negative effect on COVID-19 deaths per capita for counties with a lower percent of vaccinated residents.

Education Level and Percent of Diabetes on COVID-19 Deaths per Capita

People with a higher education might have more knowledge of the fact that COVID-19 is more deadly when there are underlying health conditions.

```
interaction_plot(covid, "some_college_or_higher", "covid_19_deaths_per_100k",
                 "percent_of_diabetes")
```

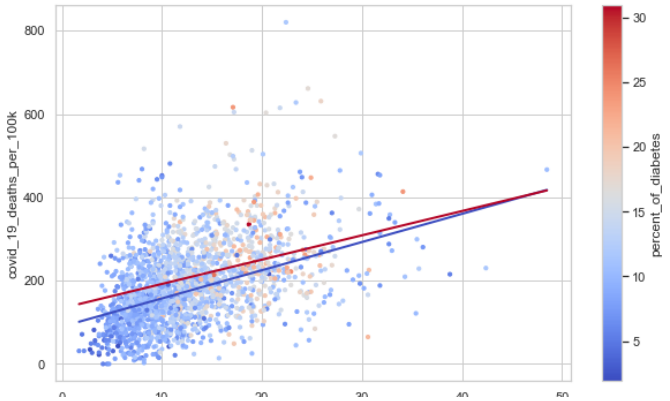
```
[1.9, 11.5]: m = -4.402248592490818
[11.5, 31.0]: m = -3.3791917212410763
```



The proportion of the county that has attended college has a more negative effect on COVID-19 deaths per capita for counties with low percent of diabetes.

```
interaction_plot(covid, "less_than_high_school_diploma", "covid_19_deaths_per_100k",
                 "percent_of_diabetes")
```

```
[1.9, 11.5]: m = 6.7626710530831895
[11.5, 31.0]: m = 5.830050749991564
```



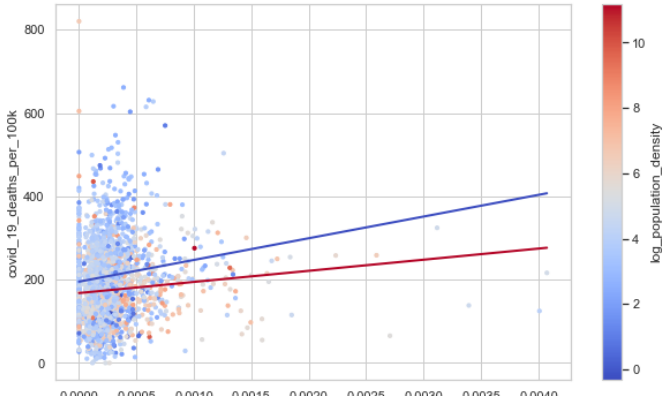
The proportion of the county that has not graduated high school has a more positive effect on COVID-19 deaths per capita for counties with low percent of diabetes.

Population Density and Ventilator Capacity Ratio on COVID-19 Deaths per Capita

Higher population density means that there are less medical supplies per person.

```
interaction_plot(covid, "ventilator_capacity_ratio", "covid_19_deaths_per_100k",
                 "log_population_density")
```

```
[-0.324, 4.239]: m = 52068.71971395643
[4.239, 11.175]: m = 26692.772159040516
```



Population density has a more positive effect on COVID-19 deaths for counties with lower population density.

Interactions with a Binary Feature

```
from matplotlib.patches import Patch
```

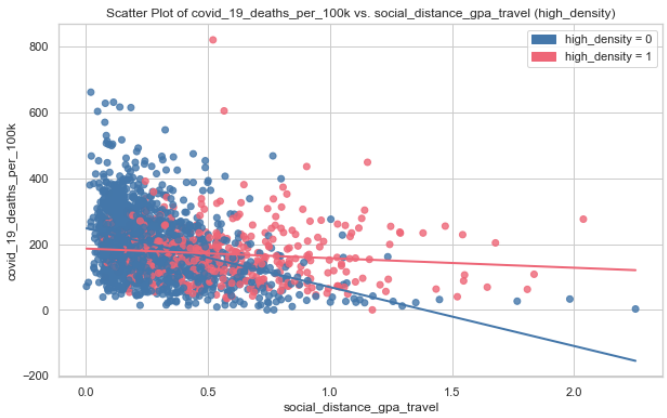
```
scheme = ["#47a", "#e67", "#283", "#cb4", "#6ce", "#a37"]
colorAlpha = .8

def color_scatter(data, x, y, c, show_lines=True):
  colors = data[c].apply(lambda x: scheme[x])
```

```
  figure = plt.figure(figsize=(10, 6))

  axes = figure.add_subplot(1, 1, 1)
  axes.scatter(data[x], data[y], marker="o", color=colors, alpha=colorAlpha)

  axes.set_ylabel(y)
  axes.set_xlabel(x)
  axes.set_title("Scatter Plot of " + y + " vs. " + x + " (" + c + ")")

  unique = data[c].unique()
  patches = []

  for i in range(len(unique)):
    patches.append(Patch(color=scheme[i], label=c + " = " + str(unique[i])))

    if show_lines:
        grouped_data = data[data[c] == unique[i]]
        fit = np.polyfit(grouped_data[x], grouped_data[y], 1)
        line_x = np.linspace(data[x].min(), data[x].max(), 10)
        line_y = list(map(np.poly1d(fit), line_x))
        axes.plot(line_x, line_y, linewidth=2, color=scheme[i])

  plt.legend(handles=patches)

  plt.show()
  plt.close()
```

Population Density

I expect social distancing GPA for travel and COVID-19 deaths per capita to have a more positive relationship for high-density counties than for low-density counties. It makes sense that COVID-19 spreads more easily in a high-density county, and the more people in a high-density county are travelling, the more COVID-19 exists in that county to spread.

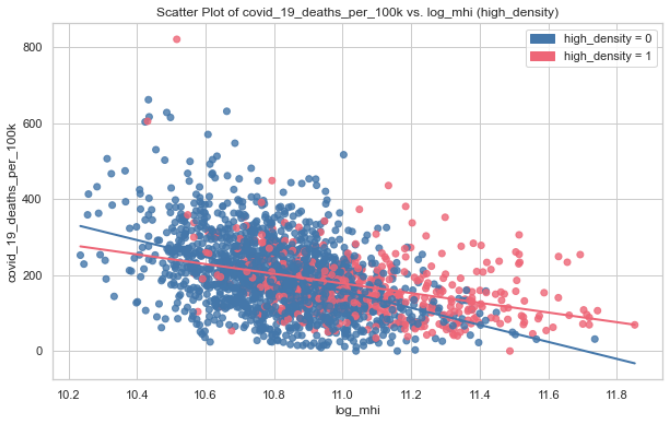
```
color_scatter(covid, "social_distance_gpa_travel", "covid_19_deaths_per_100k",
              "high_density")
```



As expected, the slope of the relationship between social distance GPA travel and COVID-19 deaths per capita is more positive for high-density counties than for low-density counties; however, both relationships are negative. This is to say that a higher social distance GPA travel generally results in lower COVID-19 deaths per capita, and this effect is stronger for low-density counties than for high-density counties.

We should explore the interaction effect between population density and median household income. I think income will have a more negative effect on COVID-19 deaths per capita in low-density counties than in high-density counties. In a low-density county, more wealth might make it easier to distance yourself from the rest of the community than in a high-density county.

```
color_scatter(covid, "log_mhi", "covid_19_deaths_per_100k",
              "high_density")
```



Median household income has a negative effect on COVID-19 deaths per capita, and this effect is more negative for low-density counties than it is for high-density counties.

I want to see the interaction between population density and percent of smokers on COVID-19 deaths per capita. In a high-density county with a high proportion of smokers, I expect that more people are affected by second-hand smoke than in other counties. It is possible that second-hand smoke could cause an underlying health condition which puts people more at risk for COVID-19 death.

```
color_scatter(covid, "percent_of_smokers", "covid_19_deaths_per_100k",
              "high_density")
```

The slope of the lines for each group are not very different; however, it looks like there is more variance in COVID-19 deaths per capita as the percent of smokers increases for low-density counties. High-density counties generally seem to have a lower percent of smokers and less variance in COVID-19 deaths per capita.

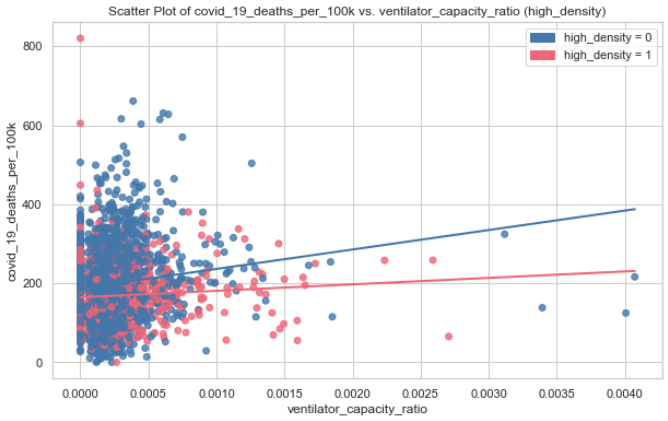
I want to see the interaction effect between population density and ventilator capacity ratio. The equation for this interaction effect is:

$$population\_density * ventilator\_capacity\_ratio = (\frac{population}{area})(\frac{ventilators}{population}) = \frac{ventilators}{area}.$$

We are measuring the amount of ventilators proportional to the size of the county. This is a measure of the accessibility of ventilators in the county, where a lower number means that people generally need to travel a greater distance to access a ventilator.

I expect that the ventilator capacity ratio will have a more negative effect on COVID-19 deaths per capita for high-density counties, since the ratio is dependent on population and a higher population that means there are more ventilators.

```
color_scatter(covid, "ventilator_capacity_ratio", "covid_19_deaths_per_100k",
              "high_density")
```



Ventilator capacity ratio has a positive effect on COVID-19 deaths per capita, and the effect is more positive for low-density counties than it is for high-density counties.

I would try including all four of these interaction effects and checking if they increase predictive power of the model compared to a base model with no interaction terms.

Initial Model

Linear Regression Functions

[ ] *2 cells hidden*

Construct Age Variables

[ ] *3 cells hidden*

Exploring Interactions

I'm going to start with the log transformation on median household income to see if that increases predictive power.

```
formula = 'covid_19_deaths_per_100k ~ percent_of_diabetes + '\
          'percent_of_smokers + log_mhi + '\
          'some_college_or_higher + less_than_high_school_diploma + adult + '\
          'old + hospital_beds_ratio + ventilator_capacity_ratio + '\
          'social_distance_gpa_travel + social_distance_gpa_visitation + '\
          'percent_of_vaccinated_residents'
result = bootstrap_linear_regression(formula, data=covid)
adjusted_r_squared(result)
```

```
0.32244055673590966
```

Adjusted R-squared is higher than before the log transformation on median household income.

I'm going to add each interaction effect to the base model one at a time to see which have the highest increase in adjusted R-squared.

```
from itertools import combinations
```

WARNING: The next chunk took me about 7 minutes to run.

```
# single_terms = ["percent_of_diabetes", "percent_of_smokers", "log_mhi",
#                 "some_college_or_higher", "less_than_high_school_diploma",
#                 "adult", "old", "hospital_beds_ratio",
#                 "ventilator_capacity_ratio", "log_population_density",
#                 "social_distance_gpa_travel", "social_distance_gpa_visitation",
#                 "percent_of_vaccinated_residents"]

# scores = pd.DataFrame()

# for pair in combinations(single_terms, 2):
#   f = formula + " + " + pair[0] + ":" + pair[1]
#   result = bootstrap_linear_regression(f, data=covid)
#   adj_r_sq = adjusted_r_squared(result)
#   scores = scores.append([[*pair, adj_r_sq]])

# scores.columns = ["var_1", "var_2", "adj_r_sq"]
# print(scores.sort_values(by="adj_r_sq", ascending=False).head(10))
```

```
formula_1 = formula + " + log_population_density:social_distance_gpa_travel \
                      + log_population_density:log_mhi \
                      + log_population_density:percent_of_smokers \
                      + log_population_density:ventilator_capacity_ratio"
result_1 = bootstrap_linear_regression(formula_1, data=covid)
adjusted_r_squared(result_1)
```

```
0.35590900636099
```

```
describe_bootstrap_lr(result_1)
```

Model: covid_19_deaths_per_100k ~ percent_of_diabetes + log_mhi + some_college_or_higher + less_than_high_school_diploma + adult + hospital_beds_ratio + ventilator_capacity_ratio + log_population_density + social_distance_gpa_travel + social_distance_gpa_visitation + percent_of_vaccinated_residents + percent_of_smokers + old + log_population_density:social_distance_gpa_travel + log_population_density:log_mhi + log_population_density:percent_of_smokers + log_population_density:ventilator_capacity_ratio

Coefficients		Mean	95% Lo
	β_{0}	1806.40	948.
percent_of_diabetes	β_{1}	0.91	-0.0
log_mhi	β_{2}	-114.16	-177
some_college_or_higher	β_{3}	-2.07	-2.8
less_than_high_school_diploma	β_{4}	1.11	-0.3
adult	β_{5}	-5.89	-7.4
hospital_beds_ratio	β_{6}	3536.44	119
ventilator_capacity_ratio	β_{7}	118886.99	668
log_population_density	β_{8}	-61.30	-235
social_distance_gpa_travel	β_{9}	-124.49	-194
social_distance_gpa_visitation	β_{10}	55.82	3.14

Model Improvement

Let's start with a log transformation on median household income to see if that increases predictive power of the model.

```
formula_m2 = 'covid_19_deaths_per_100k ~ percent_of_diabetes + \
              percent_of_smokers + log_mhi + some_college_or_higher + \
```

```
              less_than_high_school_diploma + adult + old + \
              hospital_beds_ratio + ventilator_capacity_ratio + \
              social_distance_gpa_travel + social_distance_gpa_visitation + \
              percent_of_vaccinated_residents'
result_m2 = bootstrap_linear_regression(formula_m2, data=covid,  style='linear')
adjusted_r_squared(result_m2)
```

```
0.32244055673590966
```

Adjusted R-squared increases a tiny bit when we apply the log transformation to median household income. Let's see coefficients.

```
describe_bootstrap_lr(result_m2)
```

Model: covid_19_deaths_per_100k ~ percent_of_diabetes + percent_of_smokers + log_mhi + some_college_or_higher + less_than_high_school_diploma + adult + old + hospital_beds_ratio + ventilator_capacity_ratio + social_distance_gpa_travel + social_distance_gpa_visitation + percent_of_vaccinated_residents

			95% BCI	
Coefficients		**Mean**	**Lo**	**Hi**
	β_{0}	969.55	589.57	1398.28
percent_of_diabetes	β_{1}	2.09	0.98	3.22
percent_of_smokers	β_{2}	0.96	-0.94	2.95
log_mhi	β_{3}	-32.68	-60.02	-3.88
some_college_or_higher	β_{4}	-1.83	-2.76	-1.22
less_than_high_school_diploma	β_{5}	2.34	0.73	3.36
adult	β_{6}	-6.13	-8.13	-4.19
old	β_{7}	0.23	-1.71	2.00
hospital_beds_ratio	β_{8}	3740.01	1326.61	6419.97
ventilator_capacity_ratio	β_{9}	22584.71	12105.85	37234.10

Right off the bat, some of these variables look like they could be removed. Percent of smokers, old, and social distance GPA travel do not pass the bounds test. However, some of these might be misrepresented since we are not including their interaction terms yet. Let's add all the interactions.

```
formula_m3 = 'covid_19_deaths_per_100k ~ percent_of_diabetes + \
              percent_of_smokers + log_mhi + \
              some_college_or_higher + less_than_high_school_diploma + \
              adult + old + hospital_beds_ratio + ventilator_capacity_ratio + \
              social_distance_gpa_travel + social_distance_gpa_visitation + \
              percent_of_vaccinated_residents + log_population_density + \
              log_population_density:high_density + \
              high_density:social_distance_gpa_travel + \
              high_density:log_mhi + high_density:percent_of_smokers + \
              high_density:ventilator_capacity_ratio'
result_m3 = bootstrap_linear_regression(formula_m3, data=covid)
adjusted_r_squared(result_m3)
```

```
0.3579395498838529
```

Adjusted R-squared increased by about 3.5% when we include interaction effects with high-density counties.

```
describe_bootstrap_lr(result_m3)
```

Model: covid_19_deaths_per_100k ~ percent_of_diabetes + percent_of_smokers + log_mhi + some_college_or_higher + less_than_high_school_diploma + adult + old + hospital_beds_ratio + ventilator_capacity_ratio + social_distance_gpa_travel + social_distance_gpa_visitation + percent_of_vaccinated_residents + log_population_density + log_population_density:high_density + high_density:social_distance_gpa_travel + high_density:log_mhi + high_density:percent_of_smokers + high_density:ventilator_capacity_ratio

			95% BCI	
Coefficients		**Mean**	**Lo**	**Hi**
	β_{0}	1386.60	1063.72	175
percent_of_diabetes	β_{1}	1.30	0.09	2.51
percent_of_smokers	β_{2}	-0.35	-2.24	1.85
log_mhi	β_{3}	-66.81	-92.73	-42.9
some_college_or_higher	β_{4}	-2.09	-2.81	-1.1(
less_than_high_school_diploma	β_{5}	1.42	0.19	2.76
adult	β_{6}	-5.83	-7.74	-4.2
old	β_{7}	0.12	-1.25	1.44
hospital_beds_ratio	β_{8}	3815.52	1292.69	576

Some of these terms still do not pass the bounds test, so I will try to find out which ones we can remove without losing too much predictive power from the model.

```
formula_m4 = 'covid_19_deaths_per_100k ~ percent_of_diabetes + \
              log_mhi + some_college_or_higher + less_than_high_school_diploma + \
              adult + hospital_beds_ratio + ventilator_capacity_ratio + \
              social_distance_gpa_travel + social_distance_gpa_visitation + \
              percent_of_vaccinated_residents + \
              log_population_density:high_density + \
              high_density:social_distance_gpa_travel + \
              high_density:log_mhi + high_density:ventilator_capacity_ratio'
result_m4 = bootstrap_linear_regression(formula_m4, data=covid)
adjusted_r_squared(result_m4)
```

```
0.35787013994047834
```

After removing old, log population density, and percent of smokers from the model, adjusted R-squared barely dropped at all. Let's see if this changed any other coefficients.

```
describe_bootstrap_lr(result_m4)
```

Model: covid_19_deaths_per_100k ~ percent_of_diabetes + log_mhi + some_college_or_higher + less_than_high_school_diploma + adult + hospital_beds_ratio + ventilator_capacity_ratio + social_distance_gpa_travel + social_distance_gpa_visitation + percent_of_vaccinated_residents + log_population_density:high_density + high_density:social_distance_gpa_travel + high_density:log_mhi + high_density:ventilator_capacity_ratio

			95% BCI	
Coefficients		**Mean**	**Lo**	**Hi**
	β_{0}	1455.85	1142.05	176(
percent_of_diabetes	β_{1}	1.31	0.07	2.61
log_mhi	β_{2}	-72.49	-97.80	-45.9
some_college_or_higher	β_{3}	-2.15	-2.85	-1.5(
less_than_high_school_diploma	β_{4}	1.21	-0.02	2.39
adult	β_{5}	-5.89	-7.11	-4.6
hospital_beds_ratio	β_{6}	3857.48	1388.20	604(
ventilator_capacity_ratio	β_{7}	34173.72	17447.86	647
social_distance_gpa_travel	β_{8}	-65.39	-88.68	-41.
social_distance_gpa_visitation	β_{9}	49.97	8.36	110.

Now, some college or higher does not pass the bounds test, and percent of diabetes looks dangerously close to zero. Let's see what happens when we remove them.

```
formula_m5 = 'covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + \
              adult + hospital_beds_ratio + ventilator_capacity_ratio + \
              social_distance_gpa_travel + social_distance_gpa_visitation + \
              percent_of_vaccinated_residents + \
              log_population_density:high_density + \
              high_density:social_distance_gpa_travel + \
              high_density:log_mhi + high_density:ventilator_capacity_ratio'
result_m5 = bootstrap_linear_regression(formula_m5, data=covid)
adjusted_r_squared(result_m5)
```

```
0.35531570132125134
```

Adjusted R-squared only drops by about 0.25%, but we are more confident that our coefficients have the stated effect on COVID-19 deaths per capita. Let's see the final model.

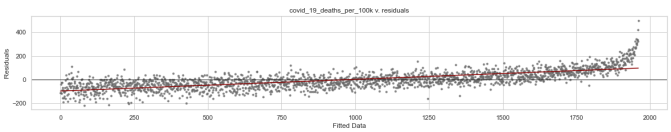
```
describe_bootstrap_lr(result_m5)
```

Model: covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + adult + hospital_beds_ratio + ventilator_capacity_ratio + social_distance_gpa_travel + social_distance_gpa_visitation + percent_of_vaccinated_residents + log_population_density:high_density + high_density:social_distance_gpa_travel + high_density:log_mhi + high_density:ventilator_capacity_ratio

			95% BCI	
Coefficients		**Mean**	**Lo**	**Hi**
	β_{0}	1646.87	1357.69	197
log_mhi	β_{1}	-84.03	-114.01	-61.
some_college_or_higher	β_{2}	-2.75	-3.14	-2.2
adult	β_{3}	-5.87	-7.33	-4.5
hospital_beds_ratio	β_{4}	3620.68	1436.17	591
ventilator_capacity_ratio	β_{5}	35333.82	21763.17	709
social_distance_gpa_travel	β_{6}	-67.29	-93.31	-44.
social_distance_gpa_visitation	β_{7}	50.06	9.30	103.
percent_of_vaccinated_residents	β_{8}	-0.62	-0.96	-0.3
log_population_density:high_density	β_{9}	18.47	8.71	28.8

The results look good in terms of our coefficients, but our predictive power is still pretty low. Let's check residual plots to see if there is any trend.

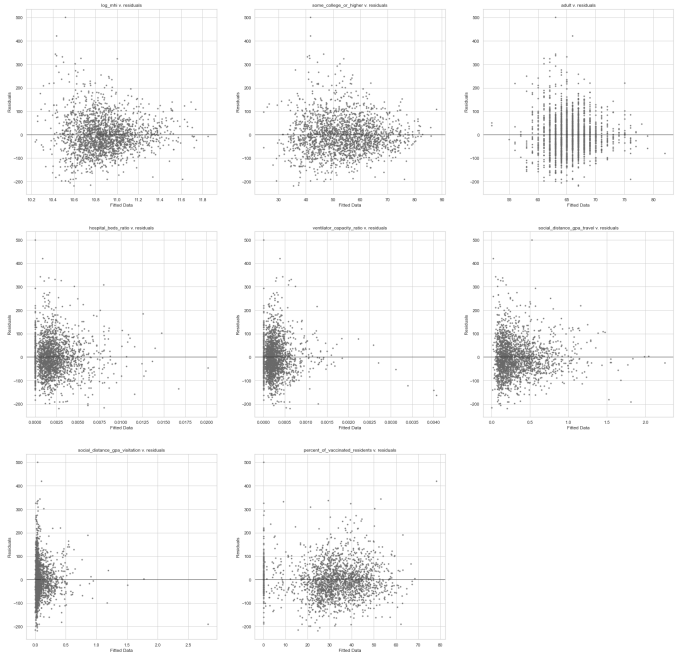
```
figure = plt.figure(figsize=(20, 3))
axes = figure.add_subplot(1, 1, 1)
residualsChart(axes,covid.covid_19_deaths_per_100k,result_m5["residuals"],add_trend=True)
plt.show()
plt.close()
```



Our residuals increase as COVID-19 deaths per capita increases, and the right side looks like we are overestimating at an exponential rate. Let's check residual plots by each feature.

```
figure = plt.figure(figsize=(30,30))
variables = list(filter(lambda x: not (":" in x) and not (x in ['intercept']),
                        result_m5["variables"]))
plots = len( variables)
rows = (plots // 3) + 1
```

```
for i, variable in enumerate(variables):
    axes = figure.add_subplot(rows, 3, i + 1)
    residualsChart(axes,covid[variable],result_m5["residuals"], sort_x=False)
plt.show()
plt.close()
```



A few of these variables, like hospital beds ratio, ventilator capacity ratio, and percent of vaccinated residents have many observations at zero. If we turn these into indicator variables, then we can explore more interaction effects with easier interpretations of their coefficients.

```
covid["bed_indicator"] = [1 if b > 0 else 0 for b in covid.hospital_beds_ratio]
covid["ventilator_indicator"] = [1 if v > 0 else 0
                                 for v in covid.ventilator_capacity_ratio]
covid["vaccinated_indicator"] = [1 if v > 0 else 0
                                 for v in covid.percent_of_vaccinated_residents]

formula_m6 = 'covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + \
              adult + bed_indicator + ventilator_indicator + \
              social_distance_gpa_travel + social_distance_gpa_visitation + \
              vaccinated_indicator + \
              log_population_density:high_density + \
              high_density:social_distance_gpa_travel + \
              high_density:ventilator_indicator'
result_m6 = bootstrap_linear_regression(formula_m6, data=covid)
adjusted_r_squared(result_m6)
```

```
0.3525933030822821
```

When we use indicator variables instead of raw values, adjusted R-squared only drops by 0.3%.

```
describe_bootstrap_lr(result_m6)
```

Model: covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + adult + bed_indicator + ventilator_indicator + social_distance_gpa_travel + social_distance_gpa_visitation + vaccinated_indicator + log_population_density:high_density + high_density:social_distance_gpa_travel + high_density:ventilator_indicator

			95% BCI	
Coefficients		**Mean**	**Lo**	**Hi**
	β_{0}	1840.45	1607.20	2126.17
log_mhi	β_{1}	-100.86	-129.62	-80.50
some_college_or_higher	β_{2}	-2.70	-3.22	-2.20
adult	β_{3}	-5.86	-6.94	-4.86
bed_indicator	β_{4}	-16.23	-53.78	29.22
ventilator_indicator	β_{5}	42.02	-3.73	84.75
social_distance_gpa_travel	β_{6}	-79.07	-99.65	-54.39

The hospital bed indicator is no longer significant, so I will remove it. The ventilator indicator has a significant interaction effect with high density, so I will leave it in. Let's try some more interaction effects to see if we can make up for the loss in predictive power.

I think vaccinated residents might have an interaction with median household income, since the vaccinated resident indicator is a good representation of the population's willingness to prevent COVID-19. Rich counties that are willing to prevent COVID-19 likely spend more of their income on other preventative measures too.

I think the vaccinated residents might also have an interaction with age, since older people are generally more at-risk and on priority for the vaccine. I will add the old variable back into the model to see if it has any effect.

```
formula_m7 = 'covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + \
              adult + old + ventilator_indicator + \
              social_distance_gpa_travel + social_distance_gpa_visitation + \
              vaccinated_indicator + \
              log_population_density:high_density + \
              high_density:social_distance_gpa_travel + \
              high_density:ventilator_indicator + \
              vaccinated_indicator:log_mhi + \
```

```
              vaccinated_indicator:adult + \
              vaccinated_indicator:old'
result_m7 = bootstrap_linear_regression(formula_m7, data=covid)
adjusted_r_squared(result_m7)
```

```
0.3615839422393775
```

This is the highest predictive power that we have gotten from any model so far. Let's check coefficients.

```
describe_bootstrap_lr(result_m7)
```

Model: covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + adult + old + ventilator_indicator + social_distance_gpa_travel + social_distance_gpa_visitation + vaccinated_indicator + log_population_density:high_density + high_density:social_distance_gpa_travel + high_density:ventilator_indicator + vaccinated_indicator:log_mhi + vaccinated_indicator:adult + vaccinated_indicator:old

Coefficients		M
	β_{0}	36
log_mhi	β_{1}	-2
some_college_or_higher	β_{2}	-2
adult	β_{3}	-9
old	β_{4}	-3
ventilator_indicator	β_{5}	26
social_distance_gpa_travel	β_{6}	-7
social_distance_gpa_visitation	β_{7}	53
vaccinated_indicator	β_{8}	-1
log_population_density:high_density	β_{9}	11
high_density:social_distance_gpa_travel	β_{10}	10

It looks like the vaccinated residents indicator did not have much of an interaction with either age level, so I will remove these and old from the model. The interaction between vaccinated residents and log median household income is definitely positive. Let's make the final model.

```
formula_m8 = 'covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + \
              adult + ventilator_indicator + social_distance_gpa_travel + \
              social_distance_gpa_visitation + vaccinated_indicator + \
              log_population_density:high_density + \
              high_density:social_distance_gpa_travel + \
              high_density:ventilator_indicator + \
              vaccinated_indicator:log_mhi'
result_m8 = bootstrap_linear_regression(formula_m8, data=covid)
adjusted_r_squared(result_m8)
```

```
0.36107444780749876
```

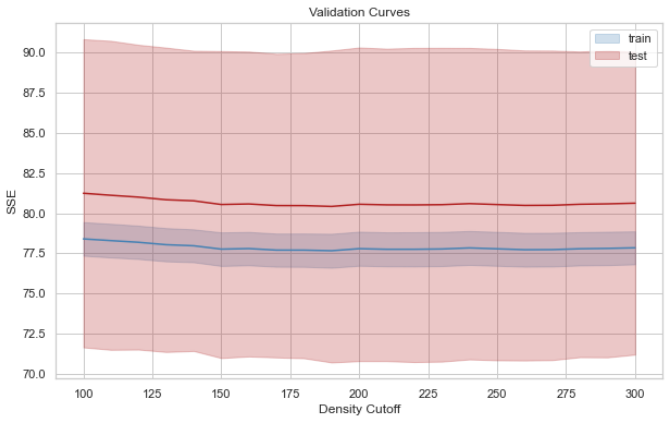
```
describe_bootstrap_lr(result_m8)
```

Model: covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + adult + ventilator_indicator + social_distance_gpa_travel + social_distance_gpa_visitation + vaccinated_indicator +

This looks like a solid, interpretable model. I am a little concerned about the method of distinguishing high-density and low-density counties, so I want to make validation curves for the population density cutoff.

```
density_values = list(range(100, 310, 10))

formulas = []
formula = 'covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + \
           adult + ventilator_indicator + \
           social_distance_gpa_travel + social_distance_gpa_visitation + \
           vaccinated_indicator + vaccinated_indicator:log_mhi'

data = covid.copy()

for v in density_values:
    label = "density_over_" + str(v)
    data[label] = [1 if d >= v else 0 for d in data.population_density]

    f = formula + f" + {label}:log_population_density \
                     + {label}:social_distance_gpa_travel \
                     + {label}:ventilator_capacity_ratio"
    formulas.append(f)

def f(formula, data, v):
    return linear_regression(formula, data, style="linear")

def r(result):
    return result["sigma"]

density_validation = validation_curves(f, formulas, data, density_values, r)
```

```
plot_validation_curves(density_validation, "SSE", "Density Cutoff", density_values)
```




```
test_validation = pd.DataFrame(results_to_curves("test", density_validation)[1:]).T
test_validation.columns = ["lower", "mean", "upper"]
test_validation["cutoff"] = density_values
test_validation.sort_values("mean").head()
```

	lower	mean	upper	cutoff
9	70.699678	80.420291	90.140903	190
8	70.963125	80.471362	89.979599	180
7	71.012343	80.474357	89.936371	170
16	70.822828	80.483801	90.144774	260

```
covid["density_over_190"] = [1 if d >= 190 else 0 for d in covid.population_density]

formula_m8 = 'covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + \
              adult + ventilator_indicator + social_distance_gpa_travel + \
              social_distance_gpa_visitation + vaccinated_indicator + \
              density_over_190:log_population_density + \
              density_over_190:social_distance_gpa_travel + \
              density_over_190:ventilator_indicator + \
              vaccinated_indicator:log_mhi'

result_m8 = bootstrap_linear_regression(formula_m8, data=covid)
adjusted_r_squared(result_m8)
```

```
0.361629322793254
```

```
describe_bootstrap_lr(result_m8)
```

Model: covid_19_deaths_per_100k ~ log_mhi + some_college_or_higher + adult + ventilator_indicator + social_distance_gpa_travel + social_distance_gpa_visitation + vaccinated_indicator + density_over_190:log_population_density + density_over_190:social_distance_gpa_travel + density_over_190:ventilator_indicator + vaccinated_indicator:log_mhi

			95% BCI	
Coefficients		**Mean**	**Lo**	**H**
	β_{0}	3340.52	2696.61	3
log_mhi	β_{1}	-239.60	-293.78	-1
some_college_or_higher	β_{2}	-2.69	-3.25	-2
adult	β_{3}	-5.79	-7.32	-4
ventilator_indicator	β_{4}	27.00	17.78	3
social_distance_gpa_travel	β_{5}	-77.91	-101.32	-5
social_distance_gpa_visitation	β_{6}	54.75	14.66	1