COVID-19 Dataset Predictive Analysis

Submitted to:

Dr Thi - Quynh Nguyen

Predictive Analytics - Quantitative Data

Report Prepared by:

Kyra Nicole Melenciano Feliz

April 09th, 2022

Table of Contents

Abstract

This report centers around the predictive analysis for the data set of COVID-19 outbreak and potential predictive features in the USA. The data set provides information related to the outbreak of COVID-19 disease in the United States, including data from its counties and states. The data set includes confirmed cases, deaths, and different features that may prove relevant to the pandemic dynamics. This is the second part of a two-part project including descriptive and predictive analysis for the data set. Go to Descriptive Analysis to read the first part.

The report is organized as follows. Section I contains the predictive analysis, including a description of the variable selection methods, the regression model used to predict COVID-19 confirmed cases and deaths, bootstrap coefficients for the regression model and model performance evaluation. Section II concludes the report with a summary of the findings and recommendations for next steps.

I. Predictive Analysis

Variable Selection

Since we're trying to predict COVID-19 confirmed cases and deaths, we want to see which variables have an effect on them. As mentioned in the abstract, this project consists of two parts: descriptive and predictive analysis. In the Descriptive Analysis we performed correlation analysis for the numerical variables and mean comparison for the categorical variables. We decided which variables to include in our regression model based on the results from that analysis.

Selected Variables

From the correlation analysis, apart from confirmed cases being linearly related to deaths, we did not find any strong relationships between the predictors and the response variables. The variables that show the higher correlation to the responses will be included in the models. The mean comparison, however, did show us that the political party and the majority of the population from a county being in a specific age group have an effect on confirmed cases and deaths. Given this, we decided to include some other variables based on domain knowledge.

Below the final list of variables to be included in each model:

Predictors for Confirmed Cases	**Predictors for Deaths**
total_state_test	total_confirmed_cases
precipitation	total_state_test
temperature	virus_pressure
virus_pressure	population_density
population_density	hospital_beds_ratio
less_than_high_school_diploma	ventilator_capacity_ratio

high_school_diploma_only	icu_beds_ratio
some_college_or_higher	percent_smokers
religious_congregation_ratio	percent_diabetes
meat_plants	political_party
age_cat	meat_plants
political_party	age_cat
	percent_insured
	median_household_income

COVID-19 Confirmed Cases Prediction

Since we know from the Descriptive Analysis that our response variables, total confirmed cases and total deaths, do not follow a normal distribution and we have count data for a specific period of time, we had to look for an alternative to the linear regression model.

Poisson regression is used to model count data.[1] The output follows the Poisson distribution assuming the logarithm of expected values. The general equation of the Poisson Regression model is:

$$log(y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ \ldots \ + \beta_n Xn$$

Before fitting the poisson regression model for each of the response variables, all numerical predictors were transformed by calculating the log of the observations to have all predictors included in the models within a similar scale (See Appendix for comparative histograms).

The data set was partitioned into training and testing sets following a 70:30 ratio. The model was fitted using the training set. After that, the predictions were made using the testing set. The

[1](DataQuest, 2019)

training set contains 1,622 rows and 28 columns. The testing set contains 730 rows and 28 columns.

Poisson Regression Model

After transforming the numerical variables, the Poisson regression model was run on the training set. The table below summarizes the results (See Appendix for R output).

Predictor	Coefficient (~)	Statistically Significant
Intercept	-13.68	Yes
Total State Test	0.02	Yes
Precipitation	-0.34	Yes
Temperature	3.09	Yes
Virus Pressure	0.49	Yes
Population Density	0.70	Yes
Less than High School	1.93	Yes
High School Diploma	0.74	Yes
Some College or Higher	6.13	Yes
Religious Congregation Ratio	0.37	Yes
Political Party	-0.26	Yes
Meat Plants	1.35	Yes
Age Category (Less than 50)	0.55	Yes
Age Category (Same Proportion)	-1.87	Yes

The variables with negative coefficients have a negative effect on total confirmed cases. For example, a county from the democratic political party (encoded as 1) with otherwise the same characteristics as one from the republican party, will have less confirmed cases.

On the contrary, the variables with positive coefficient have a positive effect on total confirmed cases. A county with the majority of the population in the age group “less than 50 years old” will have more confirmed cases than one with the majority of the population in the age group “50 years or older” despite having otherwise the same characteristics.

Multicollinearity

From the correlation analysis, we identified three pairs of predictors that could represent an issue with multicollinearity for our models. To see if that is the case, let’s check the generalized variance inflation factor (GVIF) for each predictor. The GVIF consists of the VIF adjusted by the number of degrees of freedom of the predictor.

Predictor	GVIF^(1/(2*Df))
Total State Test	1.50
Precipitation	1.52
Temperature	1.40
Virus Pressure	1.63
Population Density	1.96
Less than High School	2.82
High School Diploma	2.90
Some College or Higher	4.00
Religious Congregation Ratio	1.04
Political Party	1.28

Meat Plants	1.64
Age Category	1.00

As can be seen above, since none of the GVIFs is larger than 5, none of the predictors pose a multicollinearity issue for our model.

Bootstrap Coefficients

Bootstrapping a regression model is an alternative to the statistical approach that is based on building a sampling distribution for the regression coefficients by resampling repeatedly from the data set.[2] The process consisted of the following. We drew a sample from the data set (in our case, we used the training set which was already a sample of 70% of our data set). From the sample, we proceeded to take 1000 samples with replacement (this is necessary because, if we used samples without replacement we would get the same results with each sample). We applied the poisson regression to each of the 1000 samples and calculated the mean of each coefficient vector, which we used as our bootstrap coefficient. Below is a table comparing the bootstrap coefficients to the coefficients that resulted from the training set.

Predictor	Bootstrap	Training
Intercept	-13.14	-13.68
Total State Test	0.02	0.02
Precipitation	-0.32	-0.34
Temperature	3.03	3.09
Virus Pressure	0.49	0.49
Population Density	0.74	0.70
Less than High School	1.95	1.93

[2](John Fox & Sanford Weisberg, 2018)

High School Diploma	0.53	0.74
Some College or Higher	5.95	6.13
Religious Congregation Ratio	0.37	0.37
Political Party	-0.25	-0.26
Meat Plants	1.29	1.35
Age Category (Less than 50)	0.62	0.55
Age Category (Same Proportion)	-1.75	-1.87

All the bootstrap coefficients are pretty similar to the coefficients from the poisson regression with one sample. However, in the next section we're going to evaluate both model's performance to see which one is better for the prediction of COVID-19 confirmed cases.

Model Performance Evaluation

We used our training model to make predictions on the test data. Let's visualize how the model performed for the counties with the highest and least number of cases in the testing set.

We can see that the model performs poorly, especially for counties with a low number of total cases in the period under analysis. Let's check performance metrics for the training model, along with the bootstrap model.

Model	MAE	RMSE
Training	1,355	5,478
Bootstrap	1,275	5,055

The Mean Absolute Error (MAE) indicates the average distance between the observed and predicted values. For model comparison, the lower MAE indicates a better model. In our case, this means that the coefficients using the bootstrapping approach result in a better model. The Root Mean Square Error (RMSE) is also lower for the Bootstrap model, which confirms that this model would do a better job in predicting confirmed cases than the training model. However, the mean of total confirmed cases in the test set is 2,907, which means that both of these models are not very good in estimating COVID-19 confirmed cases, because the average distance between the observed and predicted values is almost half of the mean of observed values, 1,355 for the training model, and 1,275 for the bootstrap model.

COVID-19 Deaths Prediction

Since COVID-19 related deaths are also considered count data, we also modeled total deaths using the Poisson regression. Similar to the model for confirmed cases, all numerical predictors were transformers by calculating the log of the observations to have all predictors included in the model within a similar scale (See Appendix for comparative histograms).

Poisson Regression Model

After transforming the numerical variables, the Poisson regression model was fitted using the training set on the selected variables previously mentioned. The table below summarizes the results (See Appendix for R output).

Predictor	Coefficient (~)	Statistically Significant
Intercept	9.49	Yes
Total Confirmed Cases	0.0000033	Yes

Total State Test	-0.02	Yes
Virus Pressure	0.83	Yes
Population Density	1.17	Yes
Hospital Beds Ratio	-276.72	Yes
Ventilator Capacity Ratio	-12915.73	Yes
ICU Beds Ratio	16221.74	Yes
Percent Smokers	-0.99	Yes
Percent Diabetes	1.27	Yes
Political Party	-0.04	Yes
Meat Plants	0.97	Yes
Age Category (Less than 50)	0.32	Yes
Age Category (Same Proportion)	-3.23	Yes
Percent Insured	-4.11	Yes
Median Household Income	-0.41	Yes

Let’s check for multicollinearity. After the multicollinearity assessment is done, we’ll fit the final model.

Multicollinearity

Since we had identified pairs of predictors that could represent an issue with multicollinearity, we checked the GVIF for each predictor.

Predictor	GVIF^(1/(2*Df))
Total Confirmed Cases	1.81
Total State Test	1.49
Virus Pressure	1.91
Population Density	2.04
Hospital Beds Ratio	2.01
Ventilator Capacity Ratio	10.73
ICU Beds Ratio	11.65
Percent Smokers	2.17
Percent Diabetes	1.54
Political Party	1.31
Meat Plants	2.08
Age Category	1.00
Percent Insured	1.39
Median Household Income	2.25

The highlighted predictors have GVIFs greater than 10, which means these variables are highly correlated to other predictors and represent a strong multicollinearity issue for our model. Since we know from our Descriptive Analysis that Ventilator Capacity Ratio, Hospital Beds Ratio and ICU Beds Ratio are correlated to each other, we will remove ICU Beds Ratio from the model.

After removing the multicollinearity and the statistically insignificant predictors, the table below summarizes the final model results.

Predictor	Coefficient (~)	Statistically Significant
Intercept	7.76	Yes
Total Confirmed Cases	0.0000030	Yes
Total State Test	-0.04	Yes
Virus Pressure	0.83	Yes
Population Density	1.17	Yes
Hospital Beds Ratio	29.33	Yes
Ventilator Capacity Ratio	389.54	Yes
Percent Smokers	-0.49	Yes
Percent Diabetes	1.53	Yes
Political Party	-0.02	Yes
Meat Plants	1.07	Yes
Age Category (Less than 50)	0.32	Yes
Age Category (Same Proportion)	-3.11	Yes
Percent Insured	-3.71	Yes
Median Household Income	-0.39	Yes

As before, the variables with negative coefficients have a negative effect on total deaths. For example, a county from the democratic political party (encoded as 1) with otherwise the same characteristics as one from the republican party, will have less total deaths.

On the other side, the variables with positive coefficients have a positive effect on total deaths. A county with a higher percentage of diabetic individuals, or the majority of the population in the

age group “less than 50 years old” will have more total deaths than one with otherwise the same characteristics.

This final model was also checked for multicollinearity with the GVIFs. The table below shows that the multicollinearity issue was fixed by removing the ICU Beds Ratio Variable.

Predictor	GVIF^(1/(2*Df))
Total Confirmed Cases	1.83
Total State Test	1.49
Virus Pressure	1.89
Population Density	2.04
Hospital Beds Ratio	1.36
Ventilator Capacity Ratio	1.40
Percent Smokers	2.16
Percent Diabetes	1.57
Political Party	1.31
Meat Plants	2.04
Age Category	1.00
Percent Insured	1.38
Median Household Income	2.27

Bootstrap Coefficients

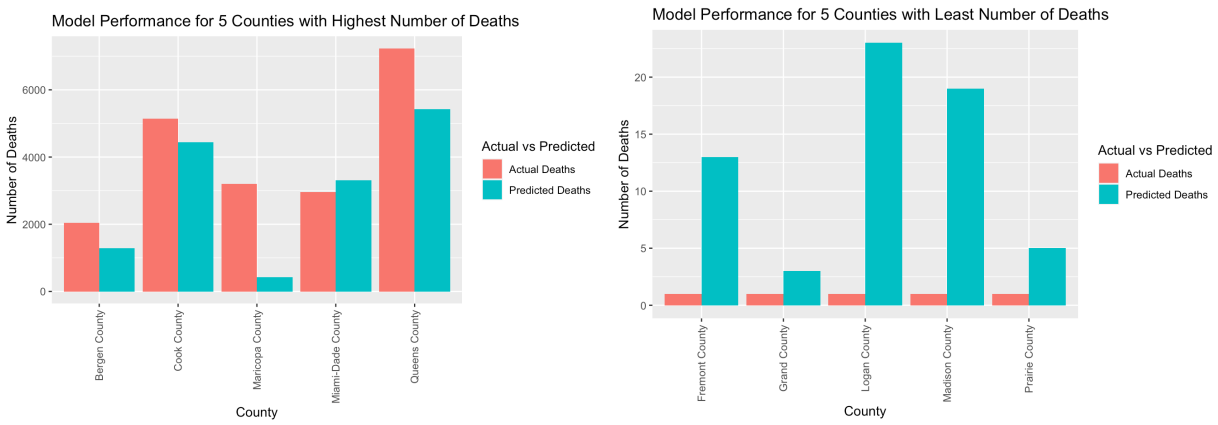
The bootstrap coefficients were also calculated for the total deaths regression model. Below is a table comparing the bootstrap coefficients to the coefficients from the training set.

Predictor	Bootstrap	Training
Intercept	5.99	7.76
Total Confirmed Cases	0.0000062	0.0000030
Total State Test	-0.024	-0.04
Virus Pressure	0.83	0.83
Population Density	1.19	1.17
Hospital Beds Ratio	28.49	29.33
Ventilator Capacity Ratio	470.56	389.54
Percent Smokers	-0.54	-0.49
Percent Diabetes	1.47	1.53
Political Party	-0.02	-0.02
Meat Plants	0.99	1.07
Age Category (Less than 50)	0.44	0.32
Age Category (Same Proportion)	-5.91	-3.11
Percent Insured	-2.51	-3.71
Median Household Income	-0.54	-0.39

As happened with confirmed cases, the coefficients for bootstrap and training are similar. In the next section we'll evaluate model performance of both cases.

Model Performance Evaluation

As before, we used the training model to make predictions of total deaths on the test data. Let's visualize how the model performed for the counties with the highest and least[3] number of cases in the testing set.



Again, the model performs poorly, especially for counties with a low number of total deaths. Let's check performance metrics.

Model	MAE	RMSE
Training	47	174
Bootstrap	484	10,736

Based on these metrics, the training model is better at predicting total deaths. The MAE for training model indicates that the average distance between observed and predicted values is 47 deaths. The RMSE is also lower for the training model. However, the mean of total deaths in the test set is 85.8, which means that none of these models is good for estimating COVID-19 related deaths, because the average distance between observed and predicted values is more than half of the mean of observed values.

[3] We had a high number of counties with 0 total deaths. For the purpose of showing the bar graph, we chose 5 counties that had 1 in total deaths.

II. Summary and Recommendations

To conclude this report, let's summarize the steps taken through both parts of this analysis to predict COVID-19 confirmed cases and deaths. After aggregating our data set based on states and counties, we analyzed the impact of several indicators on the development of confirmed cases and deaths. We found that confirmed cases are strongly correlated to deaths and the number of meat processing plants a county has. Additionally, confirmed cases have weak correlation with virus pressure. On the other side, total deaths are weakly correlated to virus pressure and the population density of the county. We also discovered that the political party of the governor of each state and whether or not the majority of the population for a county is in a specific age group had an impact on the number of cases and deaths a county had.

Even though we had a huge data set with a lot of potential predictors, we found no strong relationships between the indicators and the number of confirmed cases and deaths a county had. Therefore, we understand that there is a need to introduce other predictors to make more accurate predictions since current available indicators do not suffice this requirement. Even the CDC has stated that their forecasts have not reliably predicted the rapid changes in the trends of reported cases, hospitalizations, and deaths and these should not be relied upon for making decisions about the possibility or timing of rapid changes in trends[4].

There are many external factors that play a major role in the number of cases and deaths related to COVID-19, such as which variant of the virus a person got infected with, underlying factors such as immunity and unmeasurable behavioral indicators of how a person mentally responds to the disease, which varies from individual to individual.

When we performed our regression models, we found that the predictive power was very low and the predicted values were off from the actual ones. This led us to the conclusion that regression is not the best technique to apply to this kind of data set, machine learning algorithms could have given us better performance, but they are out of the scope of this study.

[4] (CDC, 2022)

III. References

Centers for Disease Control and Prevention (2022). COVID-19 Forecasts: Cases. Available at: https://www.cdc.gov/coronavirus/2019-ncov/science/forecasting/forecasts-cases.html

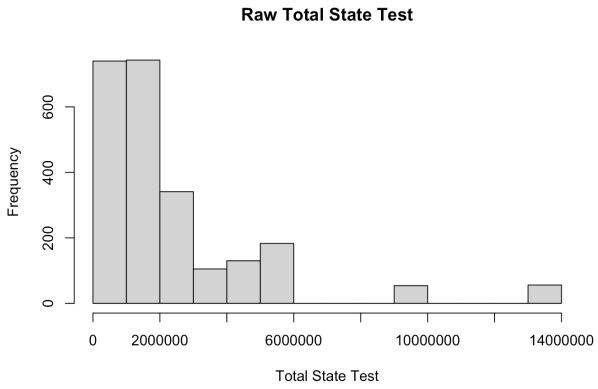
DataQuest (2019). *Tutorial: Poisson Regression in R.* [online] Available at: https://www.dataquest.io/blog/tutorial-poisson-regression-in-r/

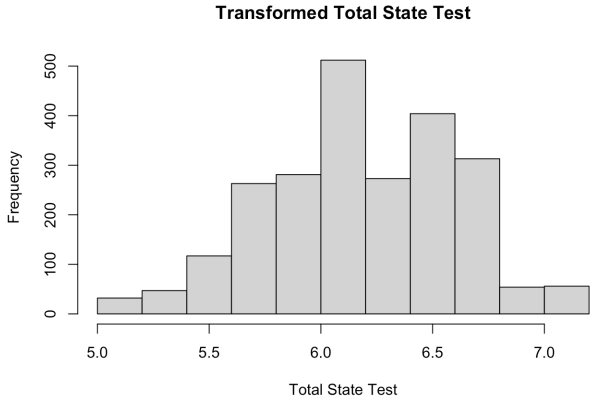
Fox John & Weisberg Sanford (2018). *Bootstrapping Regression Models in R | An R Companion to Applied Regression.* [online] Available at: https://socialsciences.mcmaster.ca/jfox/Books/Companion/appendices/Appendix-Bootstrapping.pdf

IV. Appendix

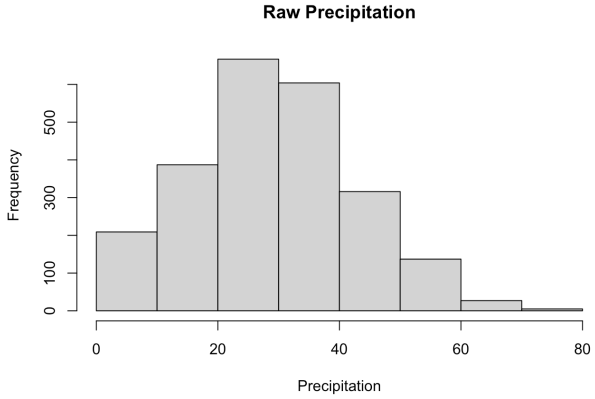
Histograms for Predictors

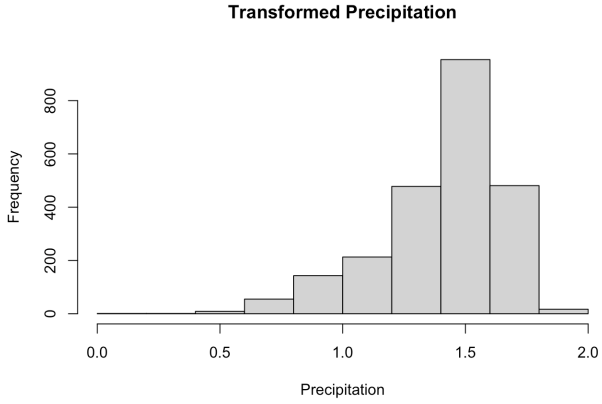
1. Total State Test



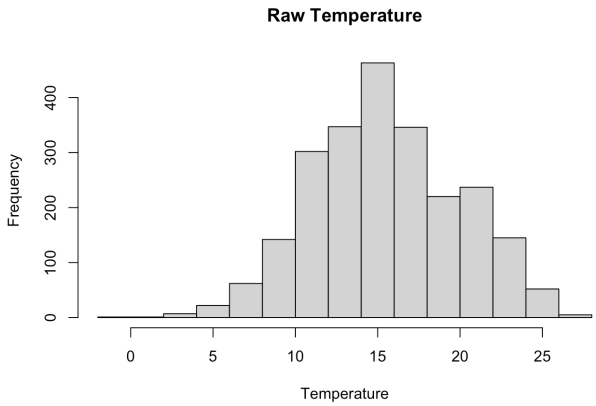


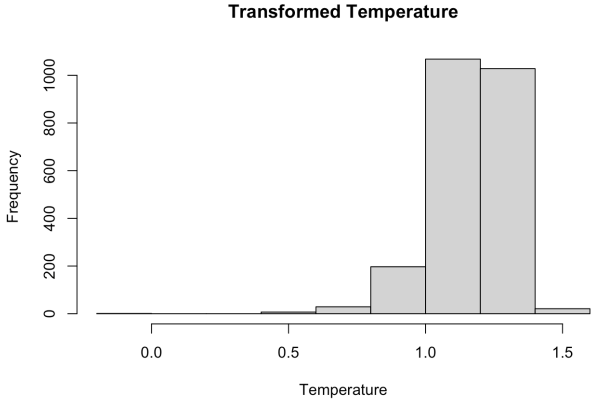
2. Precipitation



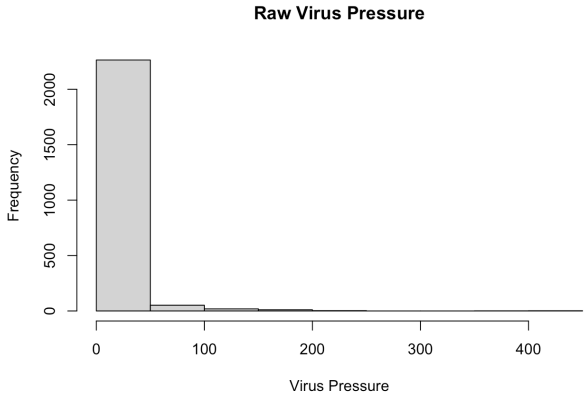


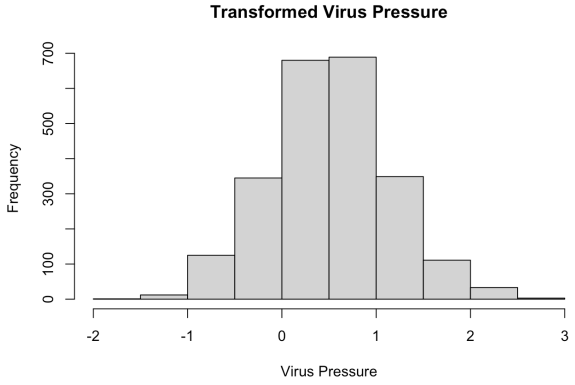
3. Temperature



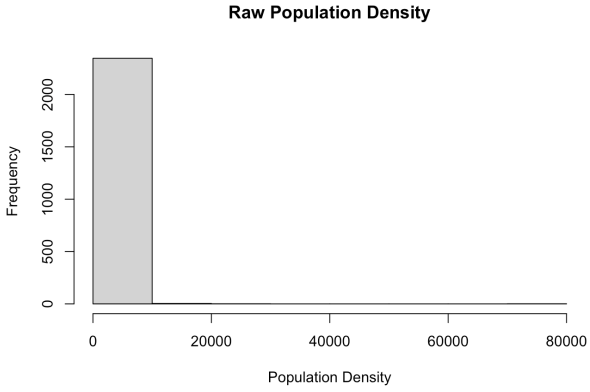


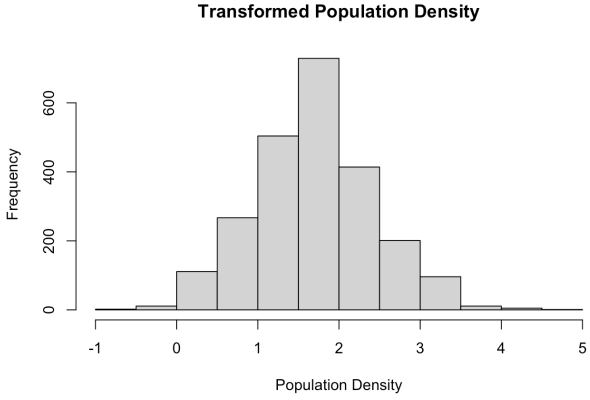
4. Virus Pressure



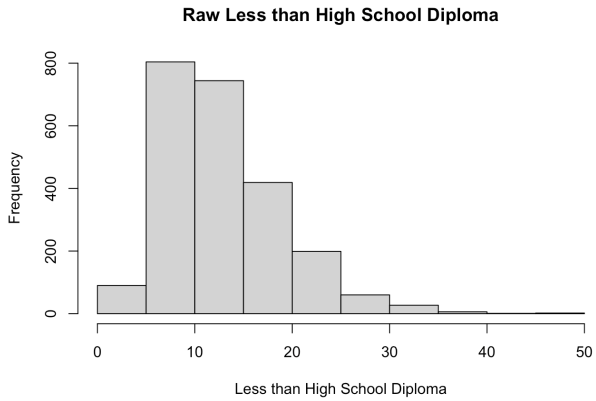


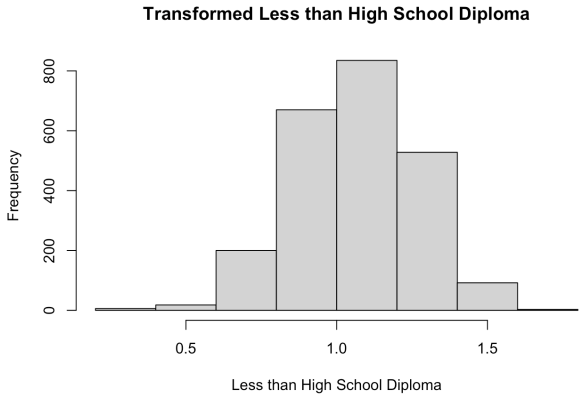
5. Population Density



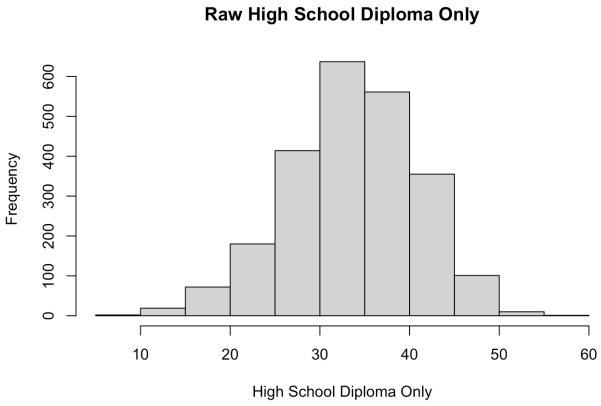


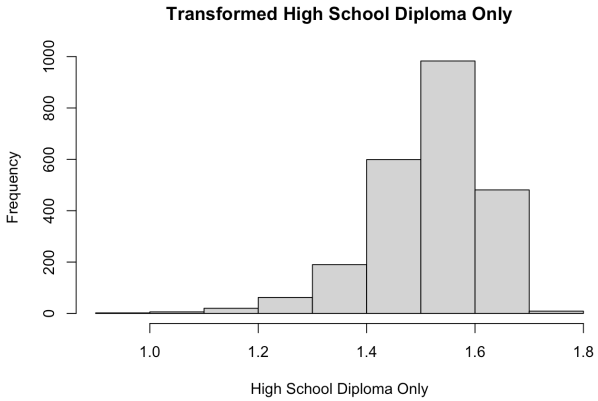
6. Less Than High School Diploma



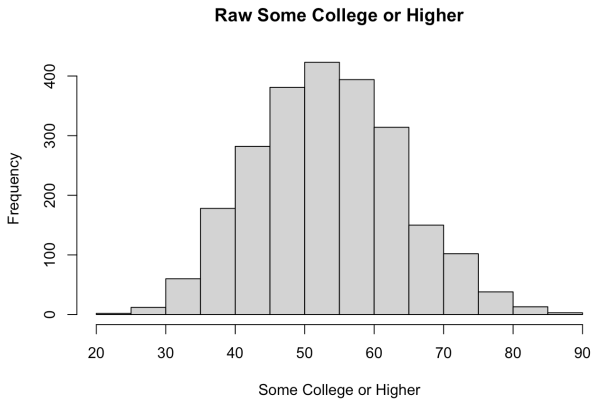


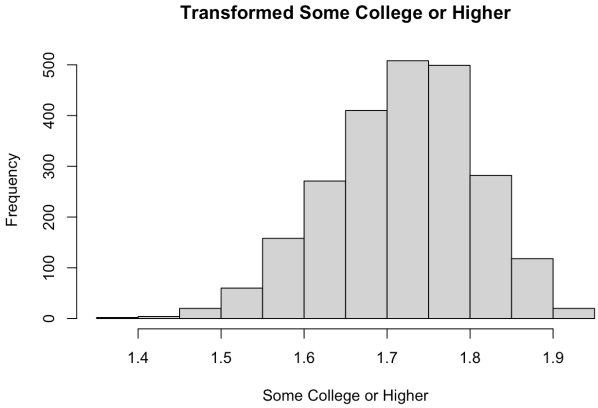
7. High School Diploma Only



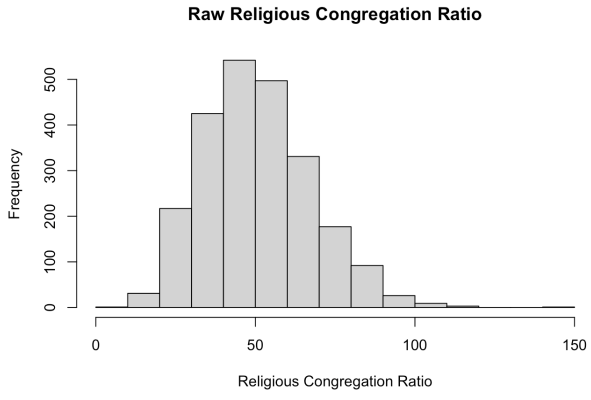


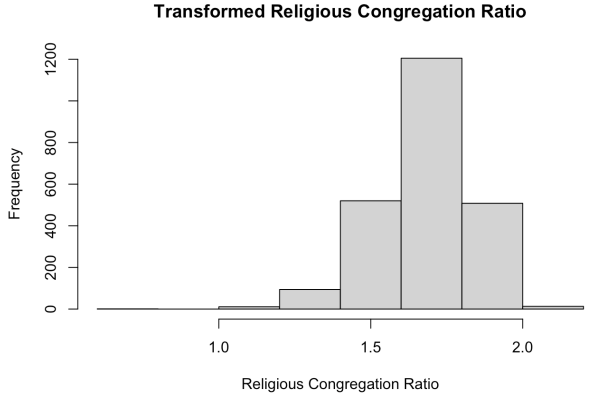
8. Some College or Higher



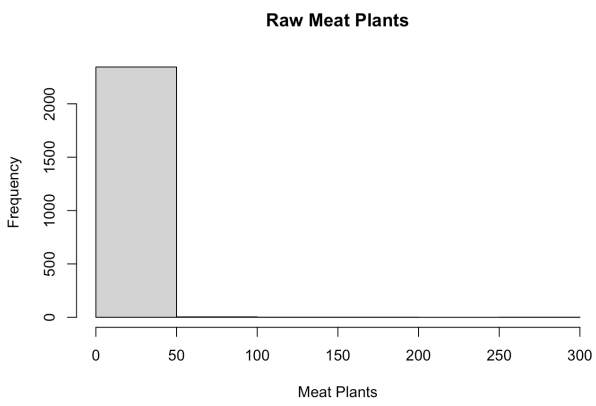


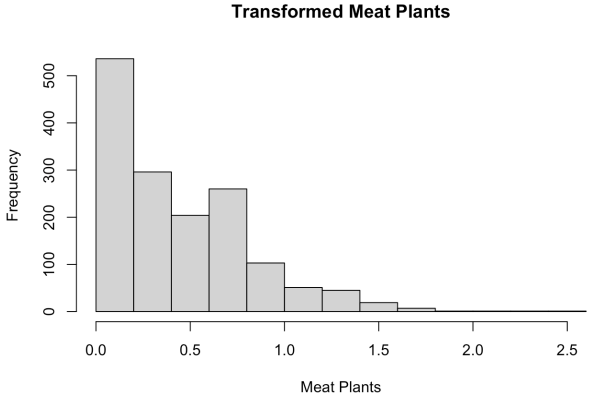
9. Religious Congregation Ratio



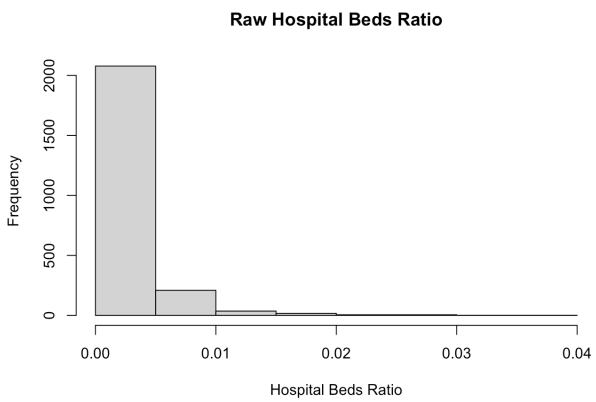


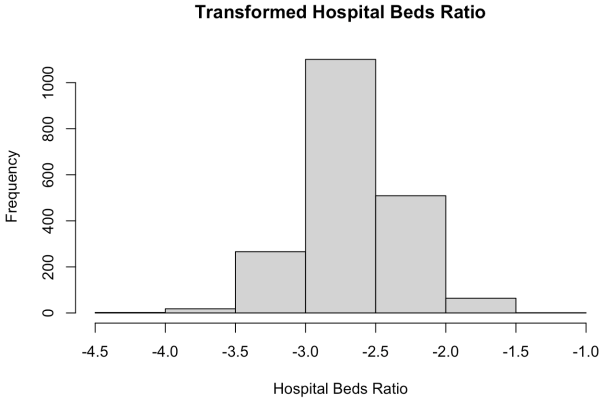
10. Meat Plants



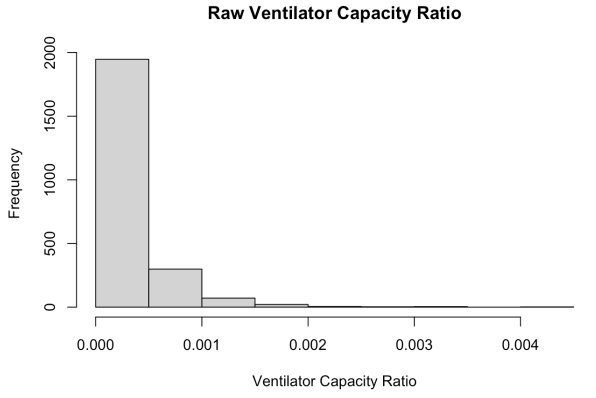


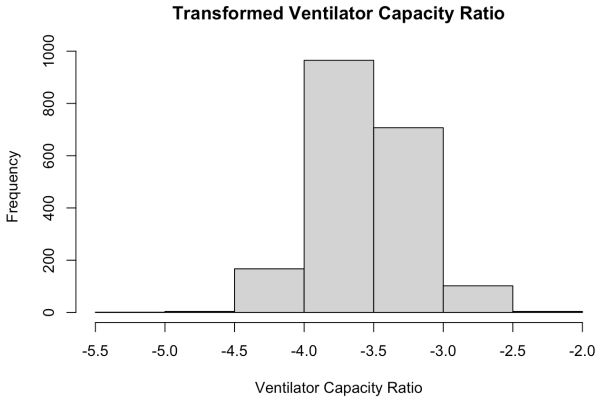
11. Hospital Beds Ratio



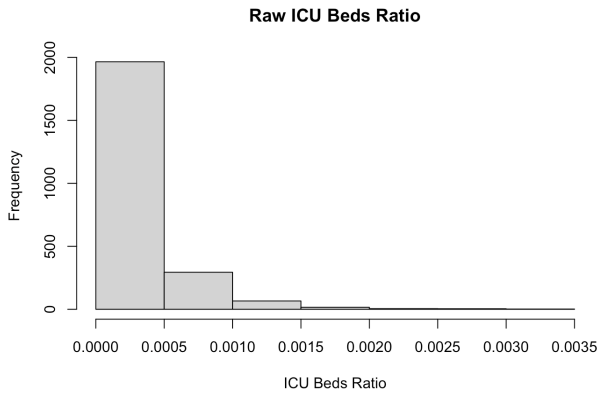


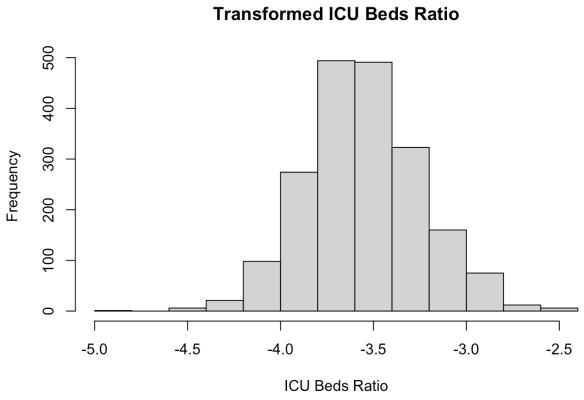
12. Ventilator Capacity Ratio



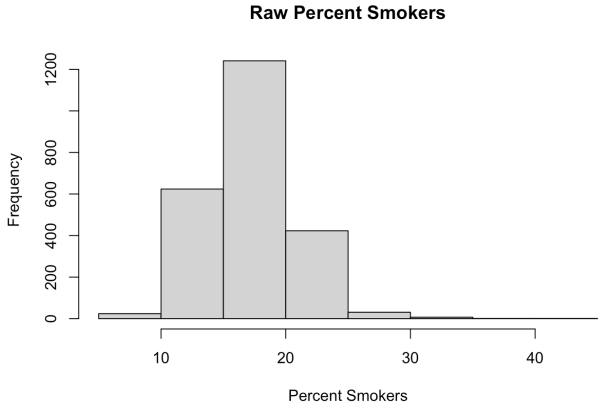


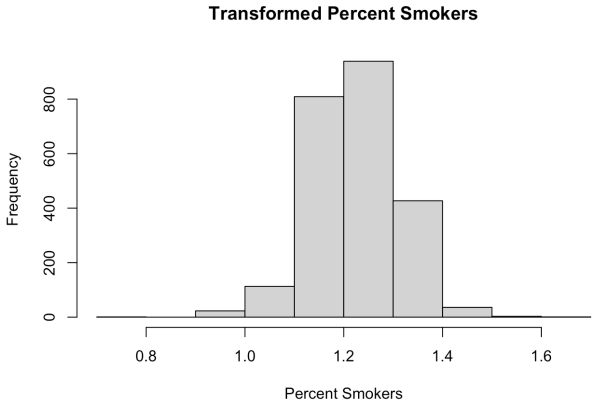
13. ICU Beds Ratio



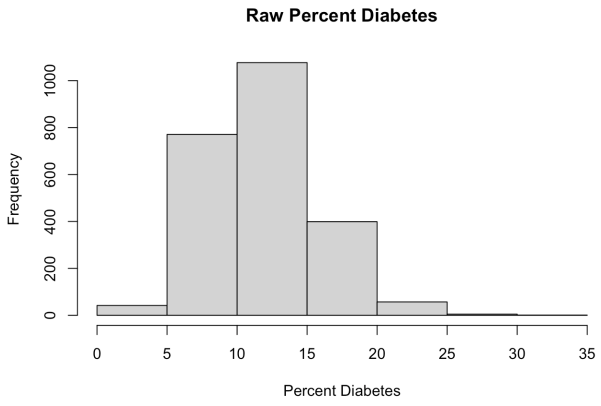


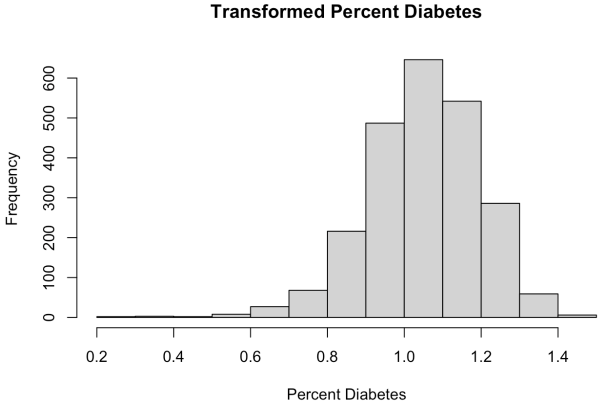
14. Percent Smokers



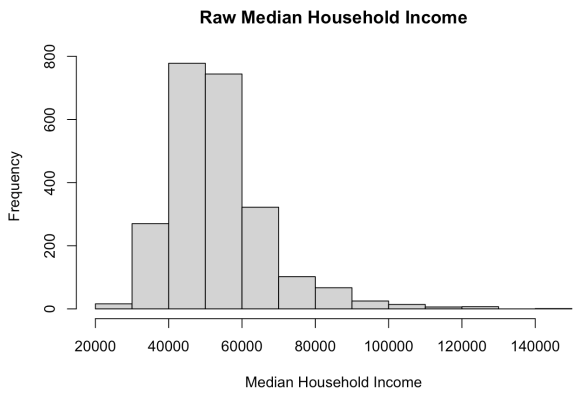


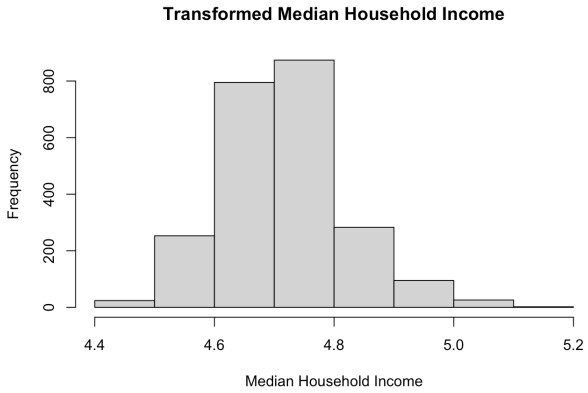
15. Percent Diabetes



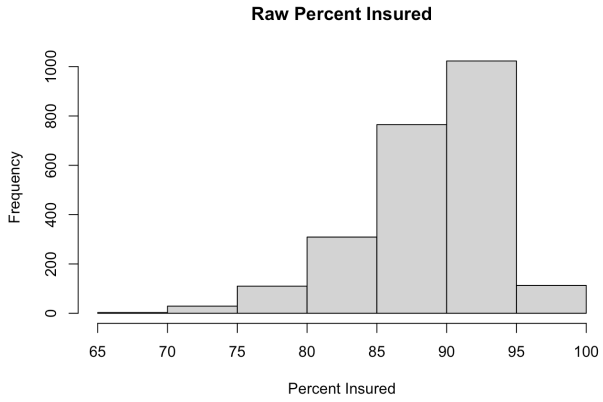


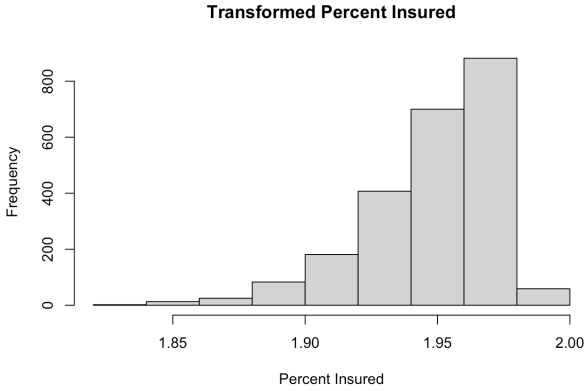
16. Median Household Income





17. Percent Insured





R Output Confirmed Cases

```
Call:
glm(formula = total_confirmed_cases ~ total_state_test + precipitation +
    temperature + virus_pressure + population_density + less_than_high_school_diploma +
    high_school_diploma_only + some_college_or_higher + Religious_congregation_ratio +
    political_party + meat_plants + age_cat, family = poisson(link = "log"),
    data = training_set)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-189.6   -18.3    -9.7     2.3   391.8

Coefficients:
                              Estimate Std. Error z value            Pr(>|z|)
(Intercept)                  -13.67807    0.07532  -181.6 <0.0000000000000002 ***
total_state_test               0.02146    0.00181    11.8 <0.0000000000000002 ***
precipitation                 -0.34428    0.00242  -142.2 <0.0000000000000002 ***
temperature                    3.08885    0.00745   414.5 <0.0000000000000002 ***
virus_pressure                 0.49276    0.00134   366.4 <0.0000000000000002 ***
population_density             0.70141    0.00125   563.1 <0.0000000000000002 ***
less_than_high_school_diploma  1.93422    0.00890   217.3 <0.0000000000000002 ***
high_school_diploma_only       0.73836    0.01263    58.4 <0.0000000000000002 ***
some_college_or_higher         6.12667    0.02836   216.0 <0.0000000000000002 ***
Religious_congregation_ratio   0.37357    0.00521    71.8 <0.0000000000000002 ***
political_party1              -0.25804    0.00132  -194.9 <0.0000000000000002 ***

meat_plants                    1.35187    0.00142   951.9 <0.0000000000000002 ***
age_catLess than 50            0.55384    0.01216    45.6 <0.0000000000000002 ***
age_catSame Proportion        -1.86984    0.07699   -24.3 <0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 11913903  on 1621  degrees of freedom
Residual deviance:  1625728  on 1608  degrees of freedom
AIC: 1638561

Number of Fisher Scoring iterations: 5
```

R Output Deaths

```
Call:
glm(formula = total_deaths ~ total_confirmed_cases + total_state_test +
    virus_pressure + population_density + hospital_beds_ratio +
    ventilator_capacity_ratio + icu_beds_ratio + percent_smokers +
    percent_diabetes + political_party + meat_plants + age_cat +
    percent_insured + median_household_income, family = poisson(link = "log"),
    data = training_set)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-33.74   -3.93   -2.22    0.15   50.22

Coefficients:
                                  Estimate        Std. Error z value             Pr(>|z|)
(Intercept)                    9.4879409409     0.3960373783   23.96 < 0.0000000000000002 ***
total_confirmed_cases          0.0000032891     0.0000000885   37.16 < 0.0000000000000002 ***
total_state_test              -0.0216917050     0.0113978040   -1.90               0.0570 .
virus_pressure                 0.8290803657     0.0096266644   86.12 < 0.0000000000000002 ***
population_density             1.1716515234     0.0075343663  155.51 < 0.0000000000000002 ***
hospital_beds_ratio         -276.7243532255     9.1751718333  -30.16 < 0.0000000000000002 ***
ventilator_capacity_ratio -12915.7277029842   209.3713276469  -61.69 < 0.0000000000000002 ***
icu_beds_ratio             16221.7408298346   255.8537376607   63.40 < 0.0000000000000002 ***
percent_smokers               -0.9868008321     0.0703727515  -14.02 < 0.0000000000000002 ***
percent_diabetes               1.2655224333     0.0453012225   27.94 < 0.0000000000000002 ***
political_party1              -0.0367438579     0.0082866683   -4.43      0.0000092460702 ***
meat_plants                    0.9688567626     0.0106518787   90.96 < 0.0000000000000002 ***
age_catLess than 50            0.3231195286     0.0696433197    4.64      0.0000034902641 ***
age_catSame Proportion        -3.2312360609     1.0024104683   -3.22               0.0013 **
percent_insured               -4.1148839153     0.1672722652  -24.60 < 0.0000000000000002 ***
median_household_income       -0.4141715911     0.0584691851   -7.08      0.0000000000014 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 405696  on 1621  degrees of freedom
Residual deviance:  61177  on 1606  degrees of freedom
AIC: 67195

Number of Fisher Scoring iterations: 6
```

```
Call:
glm(formula = total_deaths ~ total_confirmed_cases + total_state_test +
    virus_pressure + population_density + hospital_beds_ratio +
    ventilator_capacity_ratio + percent_smokers + percent_diabetes +
    political_party + meat_plants + age_cat + percent_insured +
    median_household_income, family = poisson(link = "log"),
    data = training_set)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-39.35   -3.98   -2.30   -0.02   50.42

Coefficients:
                               Estimate     Std. Error z value             Pr(>|z|)
(Intercept)                 7.7622584041   0.3989360412   19.46 < 0.0000000000000002 ***
total_confirmed_cases       0.0000029090   0.0000000883   32.94 < 0.0000000000000002 ***
total_state_test           -0.0367836164   0.0113451048   -3.24               0.0012 **
virus_pressure              0.8303707177   0.0094807678   87.58 < 0.0000000000000002 ***
population_density          1.1711324226   0.0074301646  157.62 < 0.0000000000000002 ***
hospital_beds_ratio        29.3273403138   6.0315298230    4.86      0.0000011600696 ***
ventilator_capacity_ratio 389.5445835617  27.2168050252   14.31 < 0.0000000000000002 ***
percent_smokers            -0.4944685947   0.0703369572   -7.03      0.0000000000021 ***
percent_diabetes            1.5309367393   0.0449133291   34.09 < 0.0000000000000002 ***
political_party1           -0.0209726872   0.0082430908   -2.54               0.0110 *
meat_plants                 1.0690748463   0.0105034787  101.78 < 0.0000000000000002 ***
age_catLess than 50         0.3242320932   0.0696723711    4.65      0.0000032608148 ***
age_catSame Proportion     -3.1130855817   1.0008046748   -3.11               0.0019 **
percent_insured            -3.7058371850   0.1675211581  -22.12 < 0.0000000000000002 ***
median_household_income    -0.3873061681   0.0595074940   -6.51      0.0000000000759 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 405696  on 1621  degrees of freedom
Residual deviance:  64878  on 1607  degrees of freedom
AIC: 70893

Number of Fisher Scoring iterations: 5
```

R Code Script

```
#total_confirmed cases response variable:
library(dlookr)
library(car)
setwd("/Users/kyramelenciano/dana-4810/Project")

covid_agg_data <- read.csv('covid_aggregated_FInalDataset.csv')

## Predictive Analysis
summary(covid_agg_data)
dim(covid_agg_data)
str(covid_agg_data)

### Response Variable: total_confirmed_cases

##Based upon domain knowledge our variables of interest are:
## total_state_test, precipitation, temperature, virus_pressure,
##population_density, Education(3 variables), Religious_congregation_ratio, meat_plants, and
age_cat, political_party



## Normalizing the variables of interest
## total_state_test
hist(covid_agg_data$total_state_test, breaks = 10, xlab = "Total State Test",
    main = "Raw Total State Test")
hist(log10(covid_agg_data$total_state_test), breaks = 10, xlab = "Total State Test",
    main = "Transformed Total State Test")
covid_agg_data$total_state_test <- log10(covid_agg_data$total_state_test+1)

## precipitation
hist(covid_agg_data$precipitation, breaks = 10, xlab = "Precipitation",
```

```
    main = "Raw Precipitation")
hist(log10(covid_agg_data$precipitation), breaks = 10, xlab = "Precipitation",
    main = "Transformed Precipitation")
covid_agg_data$precipitation <- log10(covid_agg_data$precipitation+1)

## temperature
hist(covid_agg_data$temperature, breaks = 10, xlab = "Temperature",
    main = "Raw Temperature")
hist(log10(covid_agg_data$temperature), breaks = 10, xlab = "Temperature",
    main = "Transformed Temperature")
covid_agg_data$temperature <- log10(covid_agg_data$temperature+1)

##virus_pressure
hist(covid_agg_data$virus_pressure, breaks = 10, xlab = "Virus Pressure",
    main = "Raw Virus Pressure")
hist(log10(covid_agg_data$virus_pressure), breaks = 10, xlab = "Virus Pressure",
    main = "Transformed Virus Pressure")
covid_agg_data$virus_pressure <- log10(covid_agg_data$virus_pressure+1)

##population_density
hist(covid_agg_data$population_density, breaks = 10, xlab = "Population Density",
    main = "Raw Population Density")
hist(log10(covid_agg_data$population_density), breaks = 10, xlab = "Population Density",
    main = "Transformed Population Density")
covid_agg_data$population_density <- log10(covid_agg_data$population_density+1)



##less_than_high_school_diploma

hist(covid_agg_data$less_than_high_school_diploma, breaks = 10,
    xlab = "Less than High School Diploma",
```

```
    main = "Raw Less than High School Diploma")
hist(log10(covid_agg_data$less_than_high_school_diploma), breaks = 10,
    xlab = "Less than High School Diploma",
    main = "Transformed Less than High School Diploma")
covid_agg_data$less_than_high_school_diploma <-
log10(covid_agg_data$less_than_high_school_diploma+1)



##high_school_diploma_only

hist(covid_agg_data$high_school_diploma_only, breaks = 10,
    xlab = "High School Diploma Only",
    main = "Raw High School Diploma Only")
hist(log10(covid_agg_data$high_school_diploma_only), breaks = 10,
    xlab = "High School Diploma Only",
    main = "Transformed High School Diploma Only")
covid_agg_data$high_school_diploma_only <-
log10(covid_agg_data$high_school_diploma_only+1)

##some_college_or_higher

hist(covid_agg_data$some_college_or_higher, breaks = 10,
    xlab = "Some College or Higher",
    main = "Raw Some College or Higher")
hist(log10(covid_agg_data$some_college_or_higher), breaks = 10,
    xlab = "Some College or Higher",
    main = "Transformed Some College or Higher")
covid_agg_data$some_college_or_higher <- log10(covid_agg_data$some_college_or_higher+1)

##Religious_congregation_ratio
```

```
hist(covid_agg_data$Religious_congregation_ratio, breaks = 10,
    xlab = "Religious Congregation Ratio",
    main = "Raw Religious Congregation Ratio")
hist(log10(covid_agg_data$Religious_congregation_ratio), breaks = 10,
    xlab = "Religious Congregation Ratio",
    main = "Transformed Religious Congregation Ratio")
covid_agg_data$Religious_congregation_ratio <-
log10(covid_agg_data$Religious_congregation_ratio+1)


## political_party factorizing the categorical variable

covid_agg_data$political_party <- as.factor(covid_agg_data$political_party)
str(covid_agg_data)


## meat_plants
hist(covid_agg_data$meat_plants, breaks = 10,
    xlab = "Meat Plants",
    main = "Raw Meat Plants")
hist(log10(covid_agg_data$meat_plants), breaks = 10,
    xlab = "Meat Plants",
    main = "Transformed Meat Plants")
covid_agg_data$meat_plants <- log10(covid_agg_data$meat_plants+1)


## age_cat factorizing the categorical variable
covid_agg_data$age_cat <- as.factor(covid_agg_data$age_cat)
str(covid_agg_data)
```

```
################################
###### Response Variable: total_deaths

##Based upon domain knowledge our variables of interest are:
## total_state_test, precipitation, temperature, virus_pressure,
##population_density, Education(3 variables), Religious_congregation_ratio, meat_plants, and
age_cat, political_party

## hospital_beds_ratio
hist(covid_agg_data$hospital_beds_ratio, breaks = 10,
    xlab = "Hospital Beds Ratio",
    main = "Raw Hospital Beds Ratio")
hist(log10(covid_agg_data$hospital_beds_ratio), breaks = 10,
    xlab = "Hospital Beds Ratio",
    main = "Transformed Hospital Beds Ratio")
covid_agg_data$hospital_beds_ratio <- log10(covid_agg_data$hospital_beds_ratio+1)

##ventilator_capacity_ratio
hist(covid_agg_data$ventilator_capacity_ratio, breaks = 10,
    xlab = "Ventilator Capacity Ratio",
    main = "Raw Ventilator Capacity Ratio")
hist(log10(covid_agg_data$ventilator_capacity_ratio), breaks = 10,
    xlab = "Ventilator Capacity Ratio",
    main = "Transformed Ventilator Capacity Ratio")
covid_agg_data$ventilator_capacity_ratio <-
log10(covid_agg_data$ventilator_capacity_ratio+1)


###icu_beds_ratio
```

```
hist(covid_agg_data$icu_beds_ratio, breaks = 10,
    xlab = "ICU Beds Ratio",
    main = "Raw ICU Beds Ratio")
hist(log10(covid_agg_data$icu_beds_ratio), breaks = 10,
    xlab = "ICU Beds Ratio",
    main = "Transformed ICU Beds Ratio")
covid_agg_data$icu_beds_ratio <- log10(covid_agg_data$icu_beds_ratio+1)


##percent_smokers
hist(covid_agg_data$percent_smokers, breaks = 10,
    xlab = "Percent Smokers",
    main = "Raw Percent Smokers")
hist(log10(covid_agg_data$percent_smokers), breaks = 10,
    xlab = "Percent Smokers",
    main = "Transformed Percent Smokers")
covid_agg_data$percent_smokers <- log10(covid_agg_data$percent_smokers+1)

##percent_diabetes
hist(covid_agg_data$percent_diabetes, breaks = 10,
    xlab = "Percent Diabetes",
    main = "Raw Percent Diabetes")
hist(log10(covid_agg_data$percent_diabetes), breaks = 10,
    xlab = "Percent Diabetes",
    main = "Transformed Percent Diabetes")
covid_agg_data$percent_diabetes <- log10(covid_agg_data$percent_diabetes +1)


##median_household_income
hist(covid_agg_data$median_household_income, breaks = 10,
    xlab = "Median Household Income",
```

```
    main = "Raw Median Household Income")
hist(log10(covid_agg_data$median_household_income), breaks = 10,
    xlab = "Median Household Income",
    main = "Transformed Median Household Income")
covid_agg_data$median_household_income <-
log10(covid_agg_data$median_household_income +1)

##percent_insured
hist(covid_agg_data$percent_insured, breaks = 10,
    xlab = "Percent Insured",
    main = "Raw Percent Insured")
hist(log10(covid_agg_data$percent_insured), breaks = 10,
    xlab = "Percent Insured",
    main = "Transformed Percent Insured")
covid_agg_data$percent_insured <- log10(covid_agg_data$percent_insured +1)



View(covid_agg_data)
str(covid_agg_data)
diagnose(covid_agg_data)




## Model



###Splitting the data
library(caTools)
set.seed(123)
split = sample.split(covid_agg_data, SplitRatio = 0.7)
```

```
training_set = subset(covid_agg_data, split == TRUE)
test_set = subset(covid_agg_data, split==FALSE)

dim(training_set)
dim(test_set)

View(training_set)

#### Variable Selection: total_confirmed_cases

full_model_ConfirmedCases <- glm(total_confirmed_cases ~ total_state_test + precipitation +
temperature + virus_pressure + population_density + less_than_high_school_diploma +
high_school_diploma_only + some_college_or_higher + Religious_congregation_ratio +
political_party+ meat_plants + age_cat, family = poisson(link='log'), data = training_set)
summary(full_model_ConfirmedCases)

### Multicollinearity

car::vif(full_model_ConfirmedCases)



x_test <- subset(test_set, select =
c(total_state_test,precipitation,temperature,virus_pressure,population_density,less_than_high_sc
hool_diploma,high_school_diploma_only,some_college_or_higher,Religious_congregation_ratio
,political_party,meat_plants,age_cat))
y_test <- subset(test_set, select = c(total_confirmed_cases))

#Making prediction using the final model on test data
predictions <- round(predict(full_model_ConfirmedCases, test_set,type="response"))
predictions
```

```
# Model performance

# Bar graphs

cases <- data.frame(County = test_set$county_name,
                    State = test_set$state_name,
                    Actual_Cases = test_set$total_confirmed_cases,
                    Predicted_Cases = predictions)

most_cases <- filter(cases, Actual_Cases > 55000)
least_cases <- filter(cases, Actual_Cases < 8)[1:5,]

library(tidyverse)
library(reshape2)
most_cases %>%
  melt %>%
  ggplot(aes(County, value, fill=variable)) +  geom_col(position = 'dodge')+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(title = 'Model Performance for 5 Counties with Highest Number of Cases',
       x = 'County', y = 'Number of Cases') +
  scale_fill_discrete(name = 'Actual vs Predicted',
                      labels=c("Actual Cases", "Predicted Cases"))

least_cases %>%
  melt %>%
  ggplot(aes(County, value, fill=variable)) +  geom_col(position = 'dodge')+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(title = 'Model Performance for 5 Counties with Least Number of Cases',
       x = 'County', y = 'Number of Cases') +
  scale_fill_discrete(name = 'Actual vs Predicted',
                      labels=c("Actual Cases", "Predicted Cases"))
```

```
## Evaluation Metrics

library(Metrics)
Metrics::mae(test_set$total_confirmed_cases, predictions)
Metrics::rmse(test_set$total_confirmed_cases, predictions)
mean(test_set$total_confirmed_cases)

#### Bootstrap coefficients confirmed cases

# Containers for the coefficients
sample_coef_intercept <- NULL
sample_coef_test <- NULL
sample_coef_precip <- NULL
sample_coef_temp <- NULL
sample_coef_vpress <- NULL
sample_coef_popden <- NULL
sample_coef_lhd <- NULL
sample_coef_hd <- NULL
sample_coef_c <- NULL
sample_coef_religion <- NULL
sample_coef_polparty <- NULL
sample_coef_meat <- NULL
sample_coef_ageless50 <- NULL
sample_coef_agesame <- NULL

for (i in 1:1000) {
  #Creating a resampled dataset from the sample data
  sample_d = training_set[sample(1:nrow(training_set), nrow(training_set), replace = TRUE), ]

  #Running the regression on these data
```

```
  model_bootstrap <- glm(total_confirmed_cases ~ total_state_test + precipitation +
                           temperature + virus_pressure + population_density +
                           less_than_high_school_diploma + high_school_diploma_only +
                           some_college_or_higher + Religious_congregation_ratio +
                           political_party+ meat_plants + age_cat,
                         family = poisson(link='log'), data = sample_d)


  #Saving the coefficients
  sample_coef_intercept <- c(sample_coef_intercept, model_bootstrap$coefficients[1])
  sample_coef_test <- c(sample_coef_test, model_bootstrap$coefficients[2])
  sample_coef_precip <- c(sample_coef_precip, model_bootstrap$coefficients[3])
  sample_coef_temp <- c(sample_coef_temp, model_bootstrap$coefficients[4])
  sample_coef_vpress <- c(sample_coef_vpress, model_bootstrap$coefficients[5])
  sample_coef_popden <- c(sample_coef_popden, model_bootstrap$coefficients[6])
  sample_coef_lhd <- c(sample_coef_lhd, model_bootstrap$coefficients[7])
  sample_coef_hd <- c(sample_coef_hd, model_bootstrap$coefficients[8])
  sample_coef_c <- c(sample_coef_c, model_bootstrap$coefficients[9])
  sample_coef_religion <- c(sample_coef_religion, model_bootstrap$coefficients[10])
  sample_coef_polparty <- c(sample_coef_polparty, model_bootstrap$coefficients[11])
  sample_coef_meat <- c(sample_coef_meat, model_bootstrap$coefficients[12])
  sample_coef_ageless50 <- c(sample_coef_ageless50, model_bootstrap$coefficients[13])
  sample_coef_agesame <- c(sample_coef_agesame, model_bootstrap$coefficients[14])

}

coefs <- as.data.frame(t(rbind(sample_coef_intercept, sample_coef_test, sample_coef_precip,
                               sample_coef_temp, sample_coef_vpress, sample_coef_popden,
sample_coef_lhd,
                               sample_coef_hd, sample_coef_c, sample_coef_religion,
```

```
                    sample_coef_polparty, sample_coef_meat,sample_coef_ageless50,
sample_coef_agesame)))
summary(coefs)

# Combining the results in a table
means.boot = c(mean(sample_coef_intercept), mean(sample_coef_test),
          mean(sample_coef_precip), mean(sample_coef_temp), mean(sample_coef_vpress),
          mean(sample_coef_popden), mean(sample_coef_lhd), mean(sample_coef_hd),
          mean(sample_coef_c), mean(sample_coef_religion),
          mean(sample_coef_polparty),mean(sample_coef_meat),
          mean(sample_coef_ageless50), mean(sample_coef_agesame))
knitr::kable(round(
  cbind(# population = coef(summary(step_model))[, 1],
    training = coef(summary(full_model_ConfirmedCases))[, 1],
    bootstrap = means.boot),4),
  "simple", caption = "Coefficients in different models")



library(tidyverse)
library(caret)

### Bootstrap model performance

# Define training control
train.control <- trainControl(method = "boot", number = 1000)
# Train the model
model_cases <- train(total_confirmed_cases ~ total_state_test + precipitation +
                     temperature + virus_pressure + population_density +
                     less_than_high_school_diploma + high_school_diploma_only +
                     some_college_or_higher + Religious_congregation_ratio +
```

```
                political_party+ meat_plants + age_cat, data = training_set,family =
poisson(link='log'),
                method = "glm",
                trControl = train.control)
# Summarize the results
print(model_cases)


#### Variable Selection: total_deaths

full_model_DeathCases <- glm(total_deaths ~ total_confirmed_cases+ total_state_test +
virus_pressure + population_density + hospital_beds_ratio + ventilator_capacity_ratio +
icu_beds_ratio  + percent_smokers + percent_diabetes+  political_party+ meat_plants + age_cat
+percent_insured + median_household_income, family = poisson(link='log'), data = training_set)
summary(full_model_DeathCases)


vif(full_model_DeathCases)


full_model_DeathCases_2 <- glm(total_deaths ~ total_confirmed_cases + total_state_test +
virus_pressure + population_density + hospital_beds_ratio  + ventilator_capacity_ratio  +
percent_smokers + percent_diabetes+  political_party+ meat_plants + age_cat  + percent_insured
+ median_household_income , family = poisson(link='log'), data = training_set)
summary(full_model_DeathCases_2)

car::vif(full_model_DeathCases_2)
```

```
x_test_deaths <- subset(test_set, select = c(total_confirmed_cases, virus_pressure ,
population_density , hospital_beds_ratio  , ventilator_capacity_ratio  , percent_smokers ,
percent_diabetes,  political_party, meat_plants , age_cat  ,percent_insured ))
y_test_deaths <- subset(test_set, select = c(total_deaths))


predictions_deaths <- round(predict(full_model_DeathCases_2, test_set,type="response"))
predictions_deaths

# Model performance

# Bar graphs
deaths <- data.frame(County = test_set$county_name,
                     State = test_set$state_name,
                     Actual_Deaths = test_set$total_deaths,
                     Predicted_Deaths = predictions_deaths)

most_deaths <- filter(deaths, Actual_Deaths > 2000)
least_deaths <- filter(deaths, Actual_Deaths == 1)[1:5,]

most_deaths %>%
  melt %>%
  ggplot(aes(County, value, fill=variable)) +  geom_col(position = 'dodge')+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(title = 'Model Performance for 5 Counties with Highest Number of Deaths',
       x = 'County', y = 'Number of Deaths') +
  scale_fill_discrete(name = 'Actual vs Predicted',
                      labels=c("Actual Deaths", "Predicted Deaths"))

least_deaths %>%
  melt %>%
```

```
  ggplot(aes(County, value, fill=variable)) +  geom_col(position = 'dodge')+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(title = 'Model Performance for 5 Counties with Least Number of Deaths',
       x = 'County', y = 'Number of Deaths') +
  scale_fill_discrete(name = 'Actual vs Predicted',
                      labels=c("Actual Deaths", "Predicted Deaths"))



# Performance metrics
library(Metrics)
Metrics::mae(test_set$total_deaths, predictions_deaths)
Metrics::rmse(test_set$total_deaths, predictions_deaths)
mean(test_set$total_deaths)

### Bootstrap coefficients deaths

# Containers for the coefficients
sample_coef_intercept <- NULL
sample_coef_cases <- NULL
sample_coef_test <- NULL
sample_coef_vpress <- NULL
sample_coef_popden <- NULL
sample_coef_hbr <- NULL
sample_coef_vcr <- NULL
sample_coef_smokers <- NULL
sample_coef_diabetes <- NULL
sample_coef_polparty <- NULL
sample_coef_meat <- NULL
sample_coef_ageless50 <- NULL
sample_coef_agesame <- NULL
sample_coef_insured <- NULL
```

```
sample_coef_income <- NULL


for (i in 1:1000) {
  #Creating a resampled dataset from the sample data
  sample_d = training_set[sample(1:nrow(training_set), nrow(training_set), replace = TRUE), ]

  #Running the regression on these data
  model_bootstrap <- glm(total_deaths ~ total_confirmed_cases + total_state_test +
virus_pressure +
                         population_density + hospital_beds_ratio  +
                         ventilator_capacity_ratio  + percent_smokers + percent_diabetes+
                         political_party+ meat_plants + age_cat  +percent_insured
+median_household_income,
                       family = poisson(link='log'), data = sample_d)


  #Saving the coefficients
  sample_coef_intercept <- c(sample_coef_intercept, model_bootstrap$coefficients[1])
  sample_coef_cases <- c(sample_coef_cases, model_bootstrap$coefficients[2])
  sample_coef_test <- c(sample_coef_test, model_bootstrap$coefficients[3])
  sample_coef_vpress <- c(sample_coef_vpress, model_bootstrap$coefficients[4])
  sample_coef_popden <- c(sample_coef_popden, model_bootstrap$coefficients[5])
  sample_coef_hbr <- c(sample_coef_hbr, model_bootstrap$coefficients[6])
  sample_coef_vcr <- c(sample_coef_vcr, model_bootstrap$coefficients[7])
  sample_coef_smokers <- c(sample_coef_smokers, model_bootstrap$coefficients[8])
  sample_coef_diabetes <- c(sample_coef_diabetes, model_bootstrap$coefficients[9])
  sample_coef_polparty <- c(sample_coef_polparty, model_bootstrap$coefficients[10])
  sample_coef_meat <- c(sample_coef_meat, model_bootstrap$coefficients[11])
  sample_coef_ageless50 <- c(sample_coef_ageless50, model_bootstrap$coefficients[12])
  sample_coef_agesame <- c(sample_coef_agesame, model_bootstrap$coefficients[13])
```

```
  sample_coef_insured <- c(sample_coef_insured, model_bootstrap$coefficients[14])
  sample_coef_income <- c(sample_coef_income, model_bootstrap$coefficients[15])

}

coefs <- as.data.frame(t(rbind(sample_coef_intercept, sample_coef_cases,sample_coef_test,
                        sample_coef_vpress, sample_coef_popden, sample_coef_hbr,
                        sample_coef_vcr, sample_coef_smokers, sample_coef_diabetes,
                        sample_coef_polparty, sample_coef_meat,sample_coef_ageless50,
                        sample_coef_agesame, sample_coef_insured, sample_coef_income)))
summary(coefs)

# Combining the results in a table
means.boot = c(mean(sample_coef_intercept),
mean(sample_coef_cases),mean(sample_coef_test),
             mean(sample_coef_vpress),
             mean(sample_coef_popden), mean(sample_coef_hbr), mean(sample_coef_vcr),
             mean(sample_coef_smokers), mean(sample_coef_diabetes),
             mean(sample_coef_polparty),mean(sample_coef_meat),
             mean(sample_coef_ageless50), mean(sample_coef_agesame),
             mean(sample_coef_insured), mean(sample_coef_income))

knitr::kable(round(
  cbind(# population = coef(summary(step_model))[, 1],
    training = coef(summary(full_model_DeathCases_2))[, 1],
    bootstrap = means.boot),4),
  "simple", caption = "Coefficients in different models")



library(tidyverse)
library(caret)
```

Bootstrap model performance

```
# Define training control
train.control <- trainControl(method = "boot", number = 1000)
# Train the model
model_deaths <- train(total_deaths ~ total_confirmed_cases + total_state_test +  virus_pressure
+
                      population_density + hospital_beds_ratio  +
                      ventilator_capacity_ratio  + percent_smokers + percent_diabetes+
                      political_party+ meat_plants + age_cat  +percent_insured +
median_household_income,
                    data = training_set,
                    family = poisson(link='log'),
                    method = "glm",
                    trControl = train.control)
# Summarize the results
print(model_deaths)
```