Regression Analysis

Group 4

3/7/2021

Hypothesis 1

Our group decided to test out 2 hypothesis for this project. This is the document for our first hypothesis.

Read input files from github

Get the link which has the merged data from github, then read the csv file and save it as a dataframe in R for analysis.

```
link='https://raw.githubusercontent.com/Public-Policy-
COVID/students_merge/main/Merged_data.csv'
myFile=url(link)
fromPy=read.csv(file = myFile)
```

Summary results of merged data

In order to understand our merged dataset, we need to know some of the basic statistical summary for it.

```
summary(fromPy)
```

```
##  Number_of_beds    Number_of_hospitals   Location         Urban_Rural_Code
##  Min.   :    0.0   Min.   :  0         Length:133         Length:133
##  1st Qu.:   25.0   1st Qu.:  1         Class :character   Class :character
##  Median :  131.0   Median :  2         Mode  :character   Mode  :character
##  Mean   :  885.4   Mean   :  5
##  3rd Qu.:  553.0   3rd Qu.:  4
##  Max.   :26672.0   Max.   :112
##   Deaths_COVID   Deaths_total       never              rarely
##  Min.   :   0   Min.   :    0   Min.   :0.00100   Min.   :0.00000
##  1st Qu.:   0   1st Qu.:    0   1st Qu.:0.01600   1st Qu.:0.01400
##  Median :  22   Median :  637   Median :0.02600   Median :0.02800
##  Mean   : 206   Mean   : 2896   Mean   :0.03513   Mean   :0.03806
##  3rd Qu.: 128   3rd Qu.: 2537   3rd Qu.:0.04500   3rd Qu.:0.05600
##  Max.   :8034   Max.   :75463   Max.   :0.14000   Max.   :0.20600
##    sometimes         frequently         always          mask_score
##  Min.   :0.00400   Min.   :0.0580   Min.   :0.3050   Min.   :2.470
##  1st Qu.:0.04800   1st Qu.:0.1410   1st Qu.:0.6160   1st Qu.:3.301
##  Median :0.06900   Median :0.1680   Median :0.6810   Median :3.464
##  Mean   :0.07167   Mean   :0.1736   Mean   :0.6814   Mean   :3.428
##  3rd Qu.:0.09100   3rd Qu.:0.2040   3rd Qu.:0.7540   3rd Qu.:3.591  
##  Max.   :0.21300   Max.   :0.3320   Max.   :0.8890   Max.   :3.822  
##  total_population   white_total_pct black_total_pct  aian_total_pct 
##  Min.   :    1129   Min.   :49.28   Min.   : 0.000   Min.   : 0.590  
##  1st Qu.:   24658   1st Qu.:82.16   1st Qu.: 0.770   1st Qu.: 1.430  
##  Median :   79481   Median :88.64   Median : 1.260   Median : 2.010  
##  Mean   :  385537   Mean   :85.50   Mean   : 2.318   Mean   : 2.985  
##  3rd Qu.:  283111   3rd Qu.:91.84   3rd Qu.: 2.620   3rd Qu.: 3.070  
##  Max.   :10039107   Max.   :96.13   Max.   :14.770   Max.   :25.690  
##  asian_total_pct  nhopi_total_pct  multiracial_total_pct
##  Min.   : 0.500   Min.   :0.0000   Min.   :1.200        
##  1st Qu.: 1.210   1st Qu.:0.2100   1st Qu.:3.160        
##  Median : 1.870   Median :0.2800   Median :3.720        
##  Mean   : 4.961   Mean   :0.3838   Mean   :3.856        
##  3rd Qu.: 5.840   3rd Qu.:0.4500   3rd Qu.:4.440        
##  Max.   :39.020   Max.   :1.7100   Max.   :7.800        
```

Test hypothesis

We use regression when we have a continuous outcome or dependent variable, and a set of independent variables which can be of different types.

Run a regression to test the hypothesis that as number of hospitals, number of hospital beds and total_population increases, covid deaths increase and as mask score increases covid deaths decrease.

We use the glm function in R to run a linear regression.

```
hypo1 = formula(Deaths_COVID~
Number_of_hospitals+Number_of_beds+mask_score+total_population)
gauss1=glm(hypo1,
           data = fromPy,
           family = 'gaussian')
```

See the results of our linear regression.

```
summary(gauss1)

## 
## Call:
## glm(formula = hypo1, family = "gaussian", data = fromPy)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -670.42   -18.47    24.92    62.44   648.61  
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)  
## (Intercept)          4.873e+02  2.254e+02   2.161 0.032530 * 
## Number_of_hospitals  2.234e+01  9.516e+00   2.348 0.020429 * 
## Number_of_beds       7.056e-02  3.975e-02   1.775 0.078268 . 
## mask_score          -1.679e+02  6.596e+01  -2.545 0.012106 *
## total_population     3.114e-04  9.110e-05   3.418 0.000845 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 26565.43)
##
##     Null deviance: 76635132  on 132  degrees of freedom
## Residual deviance:  3400375  on 128  degrees of freedom
## AIC: 1739.3
##
## Number of Fisher Scoring iterations: 2
```

Get R squared of the model. R-squared is the percentage of the dependent variable variation that a linear model explains. R-squared helps us decide how effective this model is and compare it with other hypothesis as well.

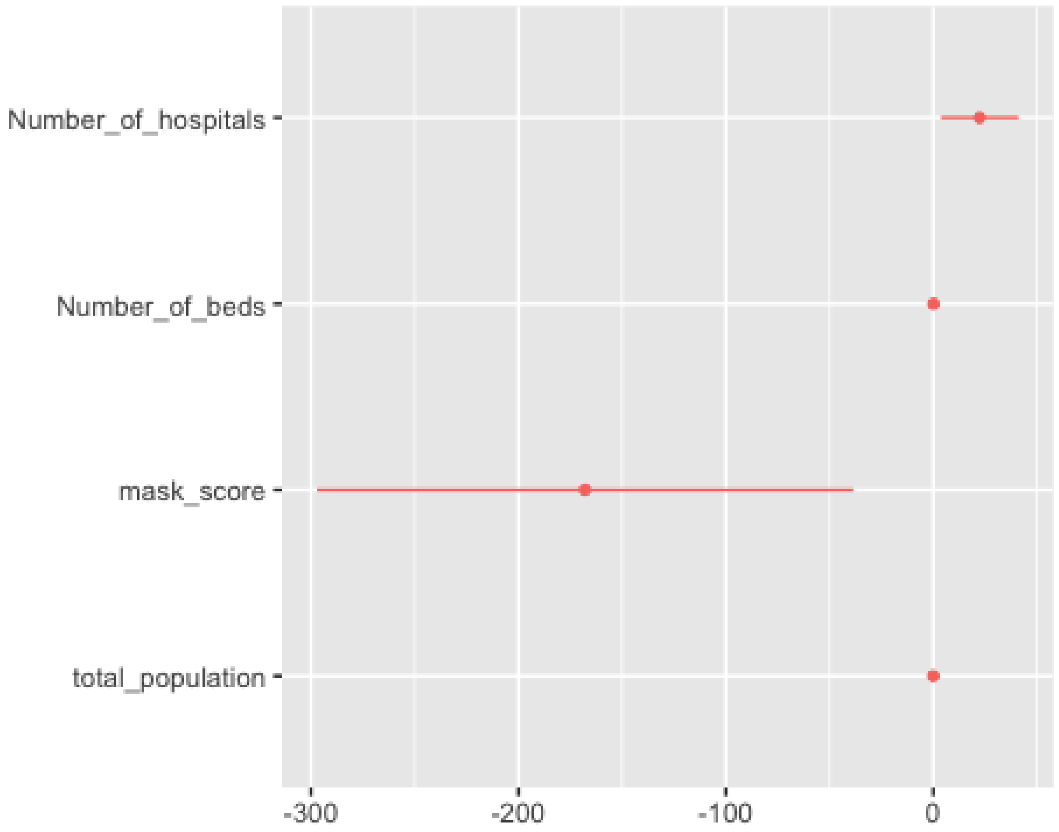
```
library(rsq)
rsq(gauss1,adj=T)

## [1] 0.9542424
```

Summary plots

In order to visualize the results of our hypothesis and the dependent variables, we use different summary plots.

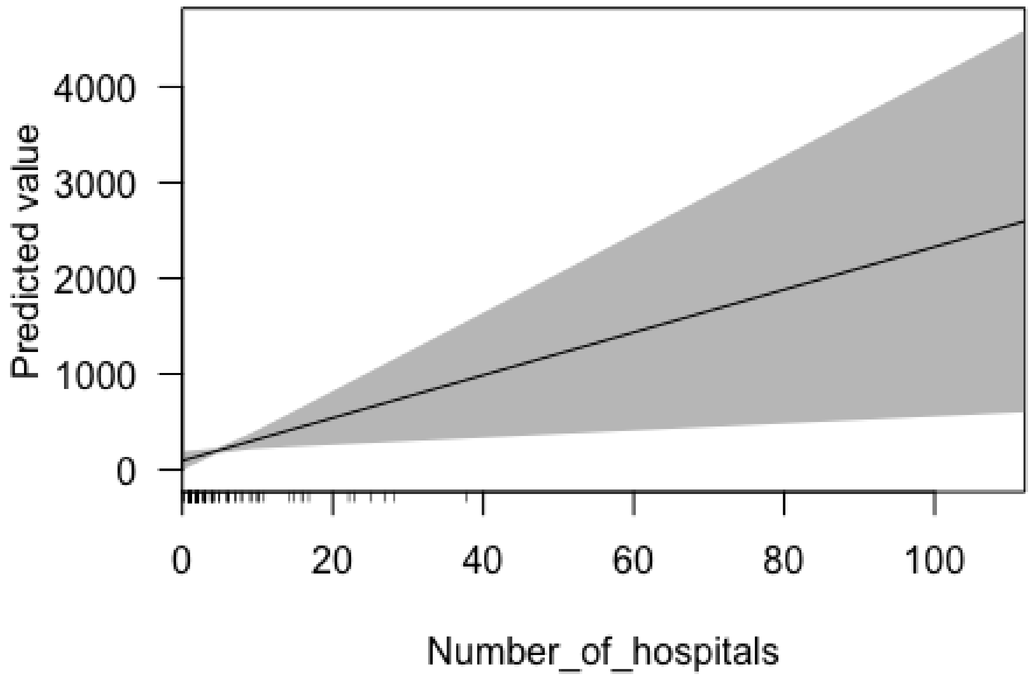
Plotting the regression coefficients.

```
# Summary plots
library(dotwhisker)

## Loading required package: ggplot2

## Warning in checkMatrixPackageVersion(): Package version inconsistency
detected.
## TMB was built with Matrix version 1.3.2
## Current Matrix version is 1.2.18
## Please re-install 'TMB' from source using install.packages('TMB', type =
'source') or ask CRAN for a binary version of 'TMB' matching CRAN's 'Matrix'
package

## Registered S3 method overwritten by 'broom.mixed':
##   method      from
##   tidy.gamlss broom

dwplot(gauss1,by_2sd = F)
```



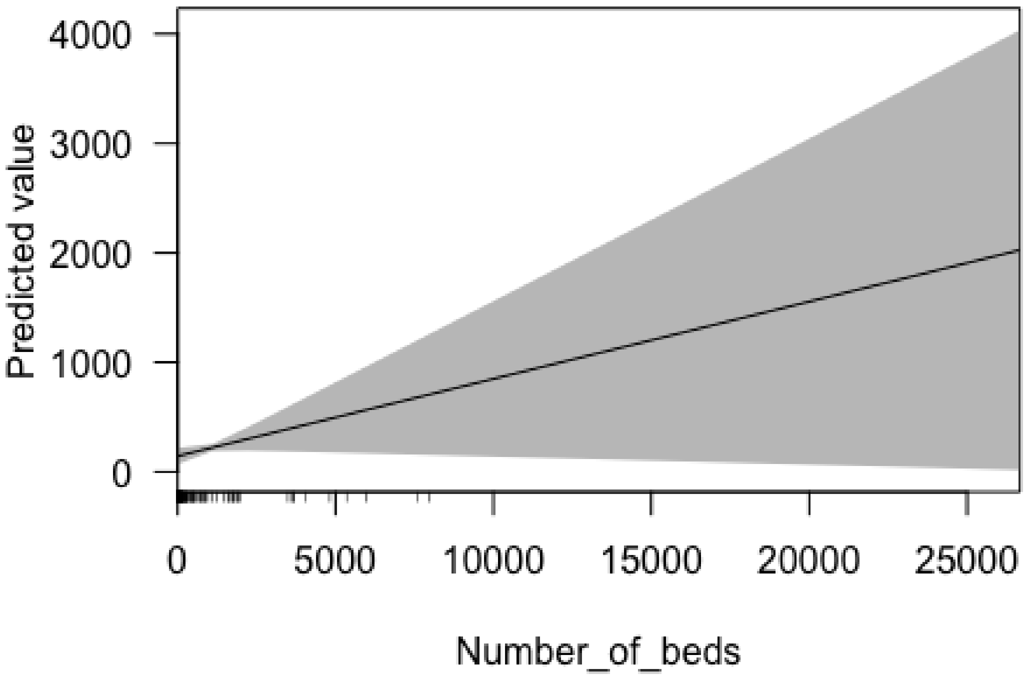
Margin plot for number of hospitals variable.

```
library(margins)
cplot(gauss1,'Number_of_hospitals')
```



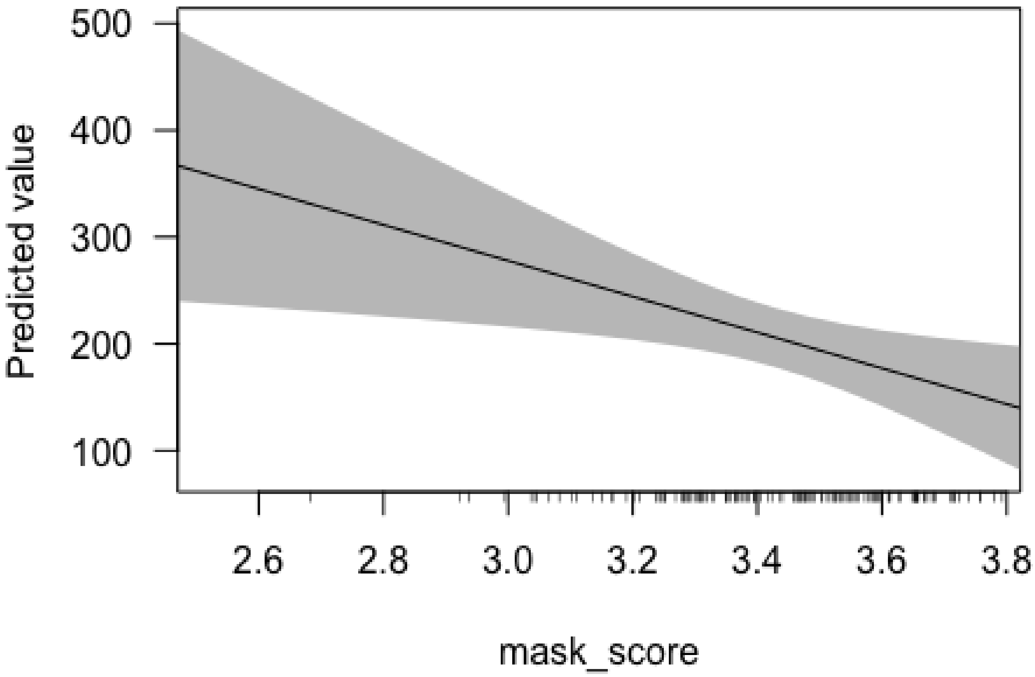
Margin plot for number of beds variable.

```
cplot(gauss1,'Number_of_beds')
```



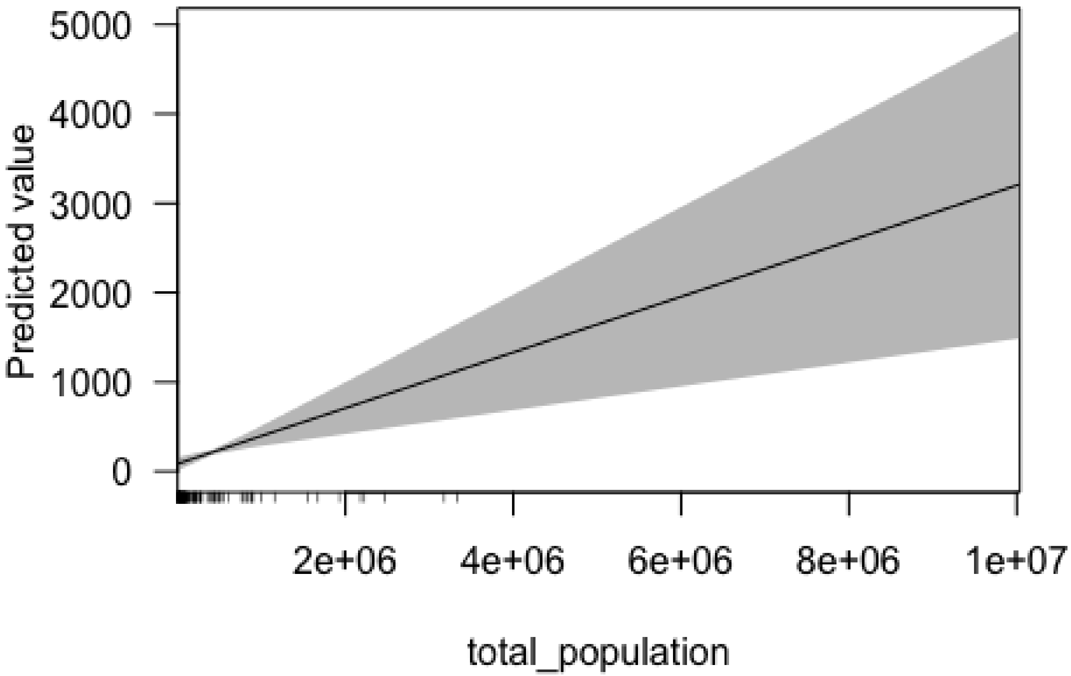
Margin plot for mask score variable.

```
cplot(gauss1,'mask_score')
```



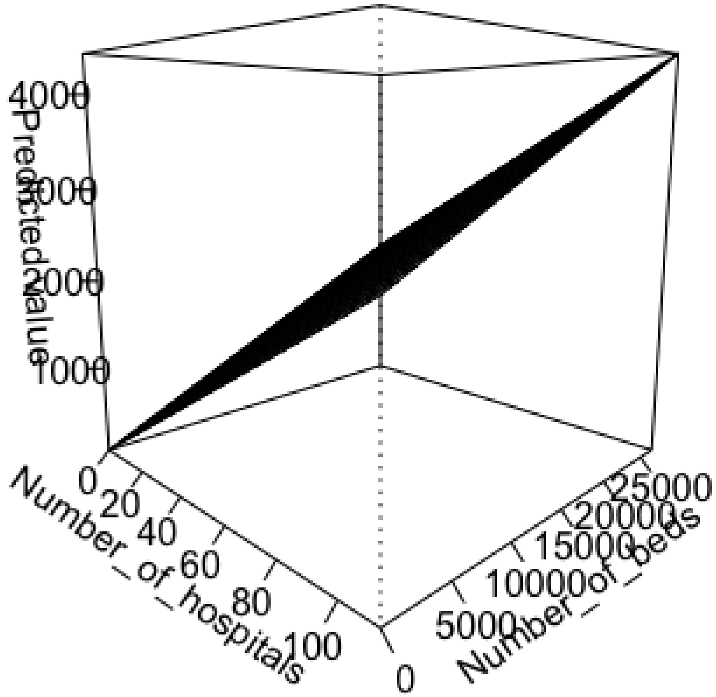
Margin plot for total population variable.

```
cplot(gauss1,'total_population')
```



Plot interaction between variables.

```
persp(gauss1)
```


Predicted value
4000
3000
2000
1000
Number_of_hospitals
0
20
40
60
80
100
Number_of_beds
0
5000
10000
15000
20000
25000