Bayesian Regression Analysis on Factors Influencing Number of Covid-19 Cases in Different Provinces of Mainland China

Course project for ISyE 6420: Bayesian Statistics, Spring 2020

Jingyu Li
alanli@gatech.edu, 903520148

1 Problem Statement

Coronavirus disease 2019 (COVID-19) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). China is the first country officially identifying a wide range infection and spread. Now COVID-19 has spread globally, resulting in over 2.2 million confirmed cases and 152 thousand confirmed deaths (WHO, updated on April 18 [1]). COVID-19 has become an ongoing pandemic all over the world.

China is among the several countries that have already controlled the COVID-19 pandemic. Tracing back to the development of this event, the sign of potential outbreak appeared in the first half of January in Hubei Province. Chinese government started the Lockdown policy in Wuhan on January 23. Then provinces and cities all over the country followed and all residency were strictly self-quarantined at home. However, it was the Spring Festival travel season in January. A great amount of people travelled from their living cities back to their hometowns. Large proportion of the confirmed cases in other provinces except Hubei were identified as having travelling history in Hubei, especially during the early period of the pandemic.

Thus, my project focused on exploring the factors that influence the number of confirmed cases in each province of mainland China except Hubei. Did the provinces adjacent to Hubei geographically have more cases? Was the number of cases different between more developed and less developed provinces?

2 Data Collection and Exploration

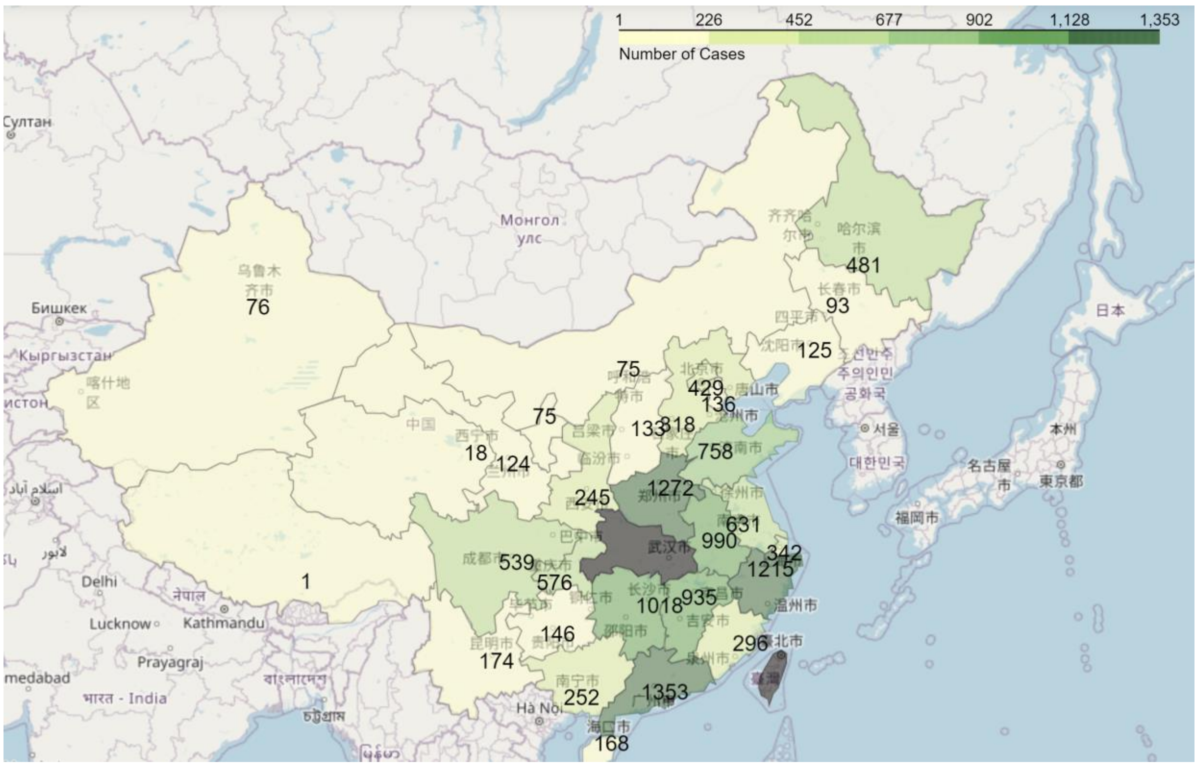
2.1 Data Collection

I collected the number of confirmed cases based on official announcements by the National Health Commission of China and the Health Commission of each province. I

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019

used the total confirmed cases by the end of March 10 as the response variable, because there were no local confirmed cases from March 6 in these provinces and the number of imported cases was still rare. This number is a good representation of how many cases each province confirmed. Following choropleth map shows the number of cases in mainland provinces.

Figure 1. Number of Confirmed Cases in Mainland China Provinces (Hubei excluded)



The predictors I collected includes:

- Population: Resident population of each province by the end of 2018 (Unit: 10,000, National Bureau of Statistics [2]).
- GDP: 2019 annual GDP of each province (Unit: 100 million RMB, National Bureau of Statistics [2]).
- Distance: Direct distance between province's capital city with Wuhan (Unit: kilometer).
- PassengerTurnover: 2018 annual railway passenger turnover, defined as the total number of passenger times the average travel distance per passenger (Unit: 100 million passengers * kilometer, National Bureau of Statistics [2]).

[2] http://www.stats.gov.cn/

- TravelConnection: Among all the sampled people who left Wuhan on January 15, the percentage that travelled to a certain province. The data was published by Baidu Map based on the data of its location-based services [3].

2.2 Data Exploration

To explore the data, I firstly calculated the correlation matrix of the response variable and different features. It showed that the correlation coefficients between number of cases and predictors are around 0.6 to 0.8, indicating that these features may be good explanatory variables.

Table 1. Correlation Matrix

	Cases	Population	GDP	Distance	Passenger Turnover	Travel Connection
Cases	1.00	0.67	0.69	-0.59	0.68	0.78
Population	0.67	1.00	0.84	-0.50	0.87	0.65
GDP	0.69	0.84	1.00	-0.51	0.69	0.49
Distance	-0.59	-0.50	-0.51	1.00	-0.57	-0.59
Passenger Turnover	0.68	0.87	0.69	-0.57	1.00	0.71
Travel Connection	0.78	0.65	0.49	-0.59	0.71	1.00

Then I applied linear regression to explore from the frequentist perspective. The adjusted R^2 of the model is 0.705. Among all the predictors, the coefficients of *GDP* and *TravelConnection* are significant.

Codes:

```
data=read.csv('covid.csv')
data=data[,5:10] #remove unneeded columns
data=data.frame(scale(data)) #scale the data
a=lm(Cases~.,data)
summary(a)
---------------------------------------------------------------------------
Residuals:
    Min         1Q       Median       3Q          Max
-1.01855  -0.26644  -0.08535   0.17654   1.28194
```

[3] http://qianxi.baidu.com/

```
Coefficients:
                    Estimate   Std. Error t value Pr(>|t|)
(Intercept)        -6.863e-17  9.925e-02   0.000 1.000000
Population         -2.964e-01  2.769e-01  -1.071 0.294973
GDP                 5.556e-01  1.978e-01   2.810 0.009711 **
Distance           -4.386e-02  1.351e-01  -0.325 0.748183
PassengerTurnover  9.726e-02  2.223e-01   0.438 0.665604
TravelConnection    6.067e-01  1.541e-01   3.936 0.000619 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.5436 on 24 degrees of freedom
Multiple R-squared:  0.7554,    Adjusted R-squared:  0.7045
F-statistic: 14.83 on 5 and 24 DF,  p-value: 1.131e-06
```

3 Bayesian Analysis

3.1 Theoretical Analysis

I assumed the prior joint distribution of β and σ^2 is non-informative, and the Bayesian regression model is:

$$y = number\ of\ confirmed\ cases$$

$$x_1 = Population, x_2 = GDP, x_3 = Distance,$$

$$x_4 = PassengerTurnover, x_5 = TravelConnection$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \epsilon, \epsilon \backsim^{iid} N(0, \sigma^2)$$

$$y\,|\beta, \sigma^2 \ \backsim\ N(X\beta, \sigma^2 I)$$

$$P(\beta, \sigma^2) \propto \frac{1}{\sigma^2}$$

∴ conditional posterior distribution of β and σ^2 are:

$$\beta|\sigma^2, y \ \backsim\ N((X^T X)^{-1} X^T y, \sigma^2 (X^T X)^{-1})$$

$$\sigma^2|\beta, y \ \backsim\ Inver - Gamma(\frac{n}{2}, \frac{e^T e}{2}), e = y - X\beta$$

3.2 Gibbs Sampling

Since the conditional posterior distribution of β and σ^2 are derived, I applied Gibbs Sampling to generate the values of parameters.

Codes:

```
library(MASS)
library(coda)
```

```
X=data[,-1]
n=dim(X)[1]
intercept=as.data.frame(rep(1,n))
colnames(intercept)='intercept'
X=as.matrix(cbind(intercept,X))
y=data$Cases

m=10000
beta=matrix(0,nrow=m,ncol=6)
sigma2=numeric(m)
sigma2[1]=summary(a)$sigma^2
Sinv=solve(t(X)%*%X)
betahat=Sinv%*%t(X)%*%y
for(i in 2:m)
{
  beta[i,]=mvrnorm(1,betahat,sigma2[i-1]*Sinv)
  e=y-X%*%beta[i,]
  sigma2[i]=1/rgamma(1,n/2,t(e)%*%e/2)
}
```

The effective size of Gibbs Sampling is shown Table 2. For all regression coefficients, the number of MC samples necessary to give the same precision as the Gibbs sample for estimating the mean is around 10,000, close to the sample size I experimented with in Gibbs Sampling. The effective size of σ^2 is relatively less. The effective size indicates that the sampled size of parameters is sufficient, there’s no need to generate more.

Table 2. MCMC Diagnostics: Effective Size

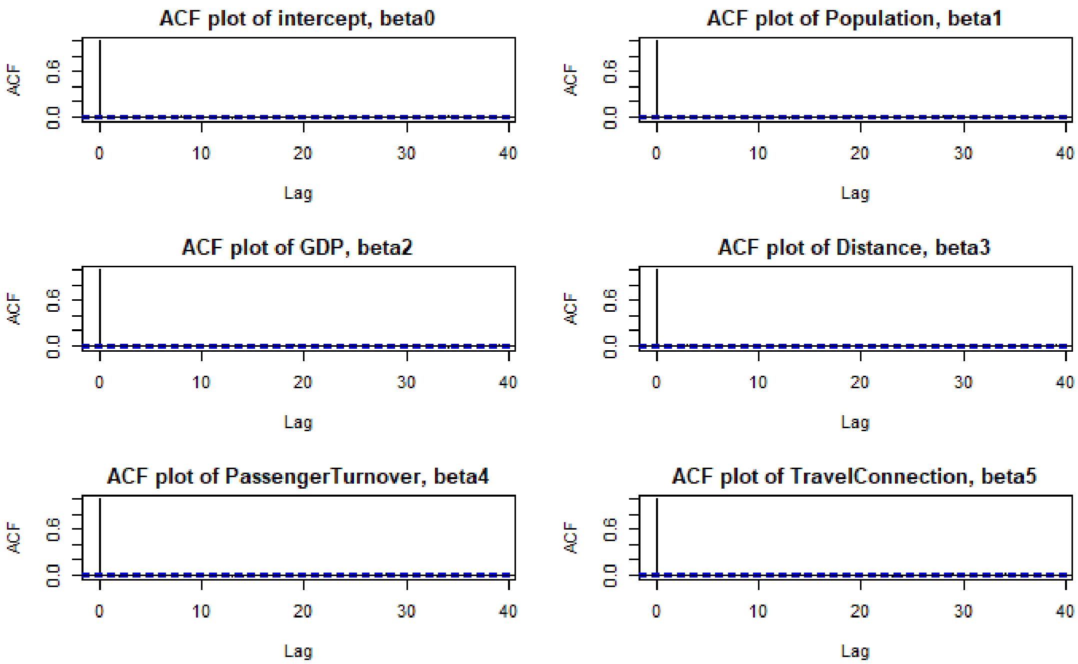
	$\boldsymbol{\beta_0}$	$\boldsymbol{\beta_1}$	$\boldsymbol{\beta_2}$	$\boldsymbol{\beta_3}$	$\boldsymbol{\beta_4}$	$\boldsymbol{\beta_5}$	$\boldsymbol{\sigma^2}$
Effective Size	9,408	10,000	10,000	9,679	10,000	10,000	6,360

Notes: Code please refer to appendix

One the other hand, ACF plots (Figure 2) showed that auto-correlation under different lags are all around zero, suggesting that the sample process is stable.

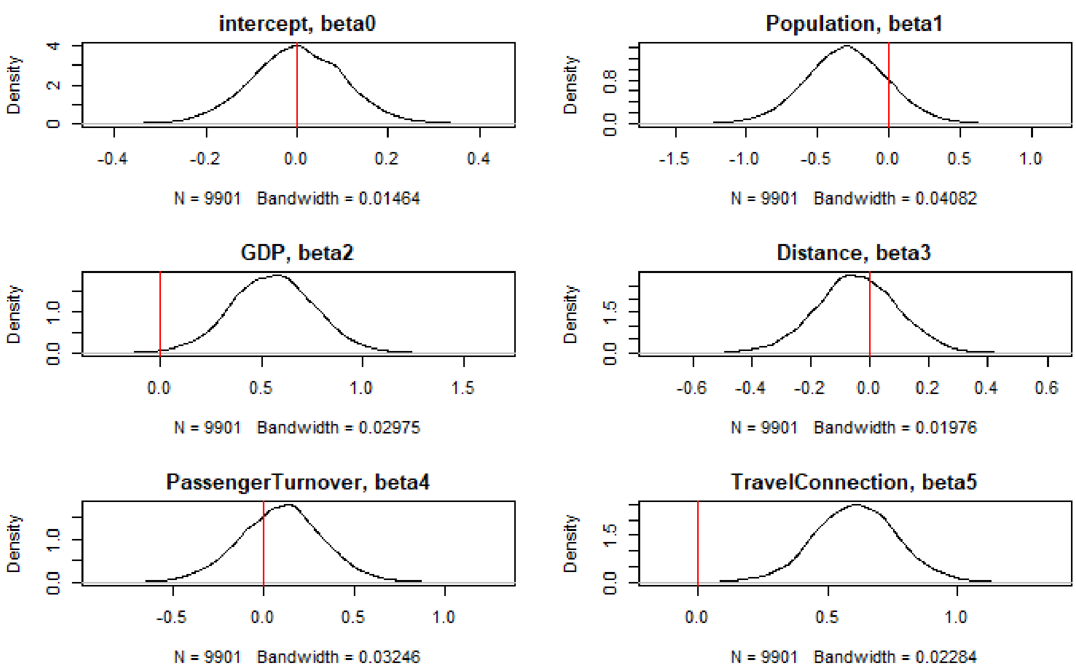
Based on the Gibbs sample generated, I calculated the 95% Highest Posterior Density (HPD) intervals of β. The density plots of all regression coefficients are also illustrated below. From both the table and plot, the coefficients of *GDP* and *TravelConnection* is away from zero with high probabilities. They both have positive influence on the number of confirmed cases in province level.

Figure 2. MCMC Diagnostics: ACF plot



Notes: Code please refer to appendix

Figure 3. Density Plots of Regression Coefficients



Notes: Code please refer to appendix

Table 3. 95% HPD Intervals of Regression Coefficients

	$\boldsymbol{\beta_0}$	$\boldsymbol{\beta_1}$	$\boldsymbol{\beta_2}$	$\boldsymbol{\beta_3}$	$\boldsymbol{\beta_4}$	$\boldsymbol{\beta_5}$
Lower	-0.200	-0.870	0.128	-0.313	-0.377	0.300
Upper	0.204	0.288	0.958	0.240	0.552	0.946

Notes: Code please refer to appendix

4 Conclusion and Discussion

This project aimed at exploring the explanatory factors for the number of confirmed cases in the provinces of Mainland China except Hubei. The factors I chose focused on the basic social and economic stats (e.g. population, GDP) and how close each province is connected with other provinces especially Hubei (e.g. distance to Wuhan, railway passenger turnover, travel connections).

Among the factors, this project found that *TravelConnection* and *GDP* are positively related to the number of confirmed COVID-19 cases. *TravelConnection* is an indirect matrics that reflects the number of people who travelled from Wuhan to other provinces and how close the two places are related with each other in terms of population migration before Spring Festival. As expected, it has a positive effect. We can also imagine that if we had access to the data like how many flights and high-speed trains were operated between Hubei and other provinces, these features may also have positive effects.

As for *GDP*, the result is a bit out of expectation. One possible explanation is that economic activities in China are highly correlated among provinces. People in provinces with higher GDP may have more travel needs. As a result, their exposure risk can also be higher.

In contrast, this project didn’t show any evidence that the population has an effect on the number of confirmed cases. One of the main reasons is that Chinese government took action to lockdown cities and force self-quarantine in a relatively quick way, and these orders were executed strictly, which effectively decreased inter-personal contact and slowed down local spread in each province.

Appendix R code

```
> # exploration
> data=read.csv('covid.csv')
> data=data[,5:10] #remove unneeded columns
> data=data.frame(scale(data)) #scale the data
> a=lm(Cases~.,data)
> summary(a)
```

```
Call:
lm(formula = Cases ~ ., data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.01855 -0.26644 -0.08535  0.17654  1.28194 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -6.863e-17  9.925e-02   0.000 1.000000    
Population        -2.964e-01  2.769e-01  -1.071 0.294973    
GDP                5.556e-01  1.978e-01   2.810 0.009711 ** 
Distance          -4.386e-02  1.351e-01  -0.325 0.748183    
PassengerTurnover  9.726e-02  2.223e-01   0.438 0.665604    
TravelConnection   6.067e-01  1.541e-01   3.936 0.000619 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5436 on 24 degrees of freedom
Multiple R-squared:  0.7554,	Adjusted R-squared:  0.7045 
F-statistic: 14.83 on 5 and 24 DF,  p-value: 1.131e-06
```

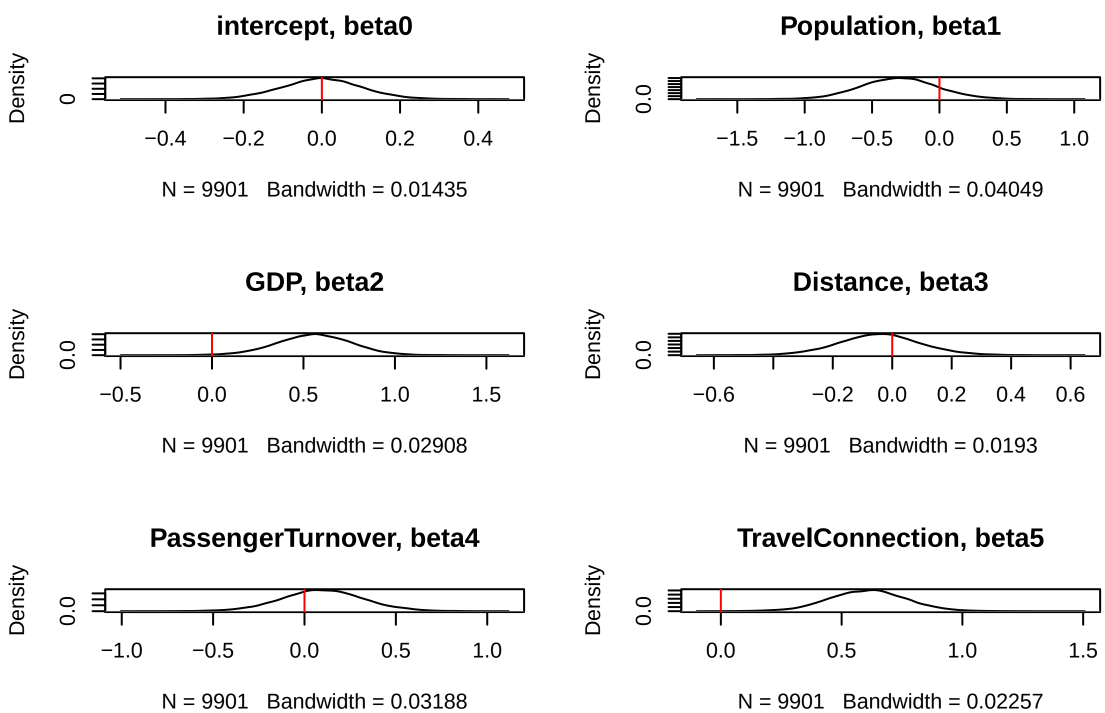
```
> #Gibbs sampling
> library(MASS)
> library(coda)
```

```
Warning: package 'coda' was built under R version 3.6.3
```

```
> X=data[,-1]
> n=dim(X)[1]
> intercept=as.data.frame(rep(1,n))
> colnames(intercept)='intercept'
> X=as.matrix(cbind(intercept,X))
> y=data$Cases
> 
> m=10000
> beta=matrix(0,nrow=m,ncol=6)
> sigma2=numeric(m)
> sigma2[1]=summary(a)$sigma^2
> Sinv=solve(t(X)%*%X)
> betahat=Sinv%*%t(X)%*%y
> for(i in 2:m)
+ {
+   beta[i,]=mvrnorm(1,betahat,sigma2[i-1]*Sinv)
```

```
+   e=y-X%*%beta[i,]
+   sigma2[i]=1/rgamma(1,n/2,t(e)%*%e/2)
+ }
```

```
> #Density plot
> par(mfrow=c(3,2))
> plot(density(beta[100:m,1]), main='intercept, beta0')
> abline(v=0,col=2)
> plot(density(beta[100:m,2]), main='Population, beta1')
> abline(v=0,col=2)
> plot(density(beta[100:m,3]), main='GDP, beta2')
> abline(v=0,col=2)
> plot(density(beta[100:m,4]), main='Distance, beta3')
> abline(v=0,col=2)
> plot(density(beta[100:m,5]), main='PassengerTurnover, beta4')
> abline(v=0,col=2)
> plot(density(beta[100:m,6]), main='TravelConnection, beta5')
> abline(v=0,col=2)
```



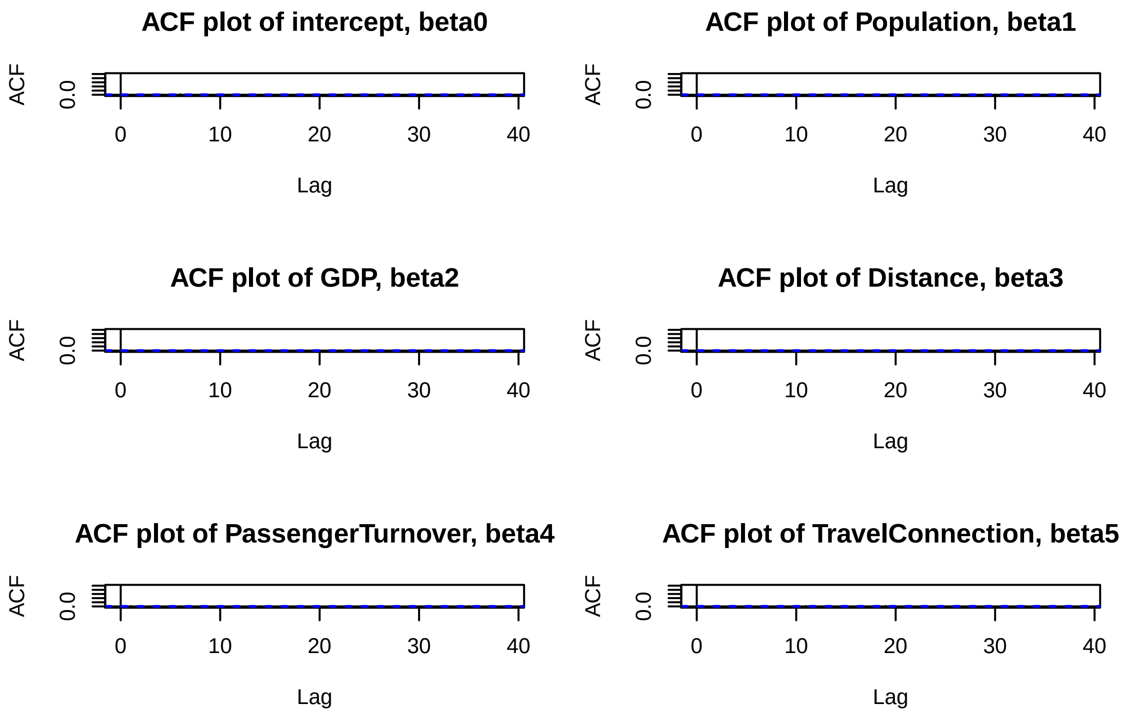

```
> #MCMC Diagnostics
> effectiveSize(beta)
```

```
     var1      var2      var3      var4      var5      var6
10000.000 10000.000 10000.000 10000.000  9992.465 10000.000
```

```
> effectiveSize(sigma2)
```

```
    var1
6204.613
```

```
> #MCMC Diagnostics: acf
> par(mfrow=c(3,2))
> acf(beta[100:m,1], main=NA)
> title('ACF plot of intercept, beta0')
> acf(beta[100:m,2], main=NA)
> title('ACF plot of Population, beta1')
> acf(beta[100:m,3], main=NA)
> title('ACF plot of GDP, beta2')
> acf(beta[100:m,4], main=NA)
> title('ACF plot of Distance, beta3')
> acf(beta[100:m,5], main=NA)
> title('ACF plot of PassengerTurnover, beta4')
> acf(beta[100:m,6], main=NA)
> title('ACF plot of TravelConnection, beta5')
```




```
> #HDI interval
> library(HDInterval)
```

```
Warning: package 'HDInterval' was built under R version 3.6.2
```

```
> hdi(beta)
```

```
            [,1]       [,2]      [,3]       [,4]       [,5]      [,6]
lower -0.2035259 -0.8425279 0.1573344 -0.3256662 -0.3325707 0.2888747
upper  0.2041301  0.2831595 0.9664854  0.2307766  0.5673654 0.9188323
attr(,"credMass")
[1] 0.95
```

Appendix Python code for visulization

April 19, 2020

[1]:
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[2]:
```python
data = pd.read_csv('covid.csv')
```

[3]:
```python
data.head()
```

[3]:
```
    Province         Lon        Lan Plot_id  Cases  Population    GDP  \
0      Anhui  117.283043  31.861191      34    990        6324  37114
1    Beijing  116.405289  39.904987      11    429        2154  35371
2  Chongqing  106.504959  29.533155      50    576        3102  23605
3     Fujian  119.306236  26.075302      35    296        3941  42395
4      Gansu  103.834170  36.061380      62    124        2637   8718

   Distance  PassengerTurnover  TravelConnection
0       457             786.38            0.0253
1      1171             154.57            0.0130
2      1078             227.09            0.0154
3       924             385.20            0.0100
4      1446             401.28            0.0041
```

[4]:
```python
data['Cases'] = data['Cases'].astype('float')
data['Plot_id'] = data['Plot_id'].astype('str')
```

[5]:
```python
import folium
from folium.features import DivIcon
import geojson

with open('china.json', 'rb') as f:
    districts = geojson.load(f)

m = folium.Map(
    location=[39.30029918615029, 103.88671875],
    zoom_start=4
)

folium.Choropleth(
    geo_data=districts,
```

```python
    name='choropleth',
    data=data,
    columns=['Plot_id', 'Cases'],
    key_on='properties.id',
    fill_color='YlGn',
    fill_opacity=0.5,
    line_opacity=0.2,
    legend_name='Number of Cases').add_to(m)

for i in range(0,len(data)):
    if i != 13:
        folium.map.Marker(
            [data.iloc[i]['Lan'], data.iloc[i]['Lon']],
            icon=DivIcon(
                icon_size=(20,15),
                icon_anchor=(10,7.5),
                html='<div style="font-size: 12pt; color:black">%s</div>' %␣
 ↪data.iloc[i]['Cases'].astype('int'))
        ).add_to(m)
    else:
        folium.map.Marker(
            [data.iloc[i]['Lan']+0.7, data.iloc[i]['Lon']+0.1],
            icon=DivIcon(
                icon_size=(20,15),
                icon_anchor=(10,7.5),
                html='<div style="font-size: 12pt; color:black">%s</div>' %␣
 ↪data.iloc[i]['Cases'].astype('int'))
        ).add_to(m)

m
```

[5]:
```
<folium.folium.Map at 0x200d1b20e80>
```

[6]:
```python
data.iloc[:,4:].corr()
```

[6]:
```
                      Cases  Population       GDP  Distance  \
Cases              1.000000    0.670640  0.694976 -0.589030
Population         0.670640    1.000000  0.842882 -0.501326
GDP                0.694976    0.842882  1.000000 -0.507243
Distance          -0.589030   -0.501326 -0.507243  1.000000
PassengerTurnover  0.682056    0.859728  0.688214 -0.574257
TravelConnection   0.784423    0.647999  0.494538 -0.586913

                   PassengerTurnover  TravelConnection
Cases                       0.682056          0.784423
Population                  0.859728          0.647999
GDP                         0.688214          0.494538
Distance                   -0.574257         -0.586913
```

```
PassengerTurnover          1.000000          0.712156
TravelConnection           0.712156          1.000000
```

[7]:
```
plt.figure(figsize=(8,6),dpi=72)
sns.heatmap(data.iloc[:,4:].corr())
```

[7]: `<matplotlib.axes._subplots.AxesSubplot at 0x200d1b62be0>`