An Analysis of Mexican COVID-19 survival from 2020 to 2021

Enbo Tian [*], Diyu Yang [†],
Advisor: Buddika Peiris

October 21, 2022

A project report submitted to Worcester Polytechnic Institute
in partial fulfillment of the requirements for the Degree of Master of Science
in Applied Statistics

[*] Department of Mathematics, Email: enbotian0@gmail.com

[†] Department of Mathematics, Email: dyang3@wpi.edu

Abstract

This study provides an analysis on Mexican COVID-19 dataset using survival analysis by focusing on analyzing patterns and changes of the survival rate of Mexican COVID-19 patients. COVID-19 has already been a global epidemic for more than two years, but improving understanding on COVID-19 and increasing immunization rate across Mexico result in a increasing survival rate for COVID-19 patients.

Acknowledgements

Thanks to our advisor, Professor Buddika Peiris of the WPI Mathematical Science Department, for his advice and guidance on our project and on our report.

Contents

1 Introduction

COVID-19 has already been a global epidemic for more than two years. The number of COVID-19 infections continues to rise in these two years. In Mexico, Hospitals are keeping recording the information of patients starting from January 2020. COVID-19 also affects different populations in different ways. Most infected people develop mild to moderate symptoms and recover without needing to be hospitalized. However, There are still three percent of people who died of COVID-19.
In recent research, There is an article talked about the survival analysis of COVID-19 in Mexico. The method of that article is based on the whole population of Mexico with using Kaplan-Meier method. In this report, we are going to do some more processing to the data and use more method.

2 Data Source

Our data are collected from the Salud website of Mexico, which include 40 parameters and 13244807 data for each parameters. However, many of the parameter was not used in this research. So we reduced the data set and leave the following parameter:

data_updated: The database is fed daily, this variable allows to identify the date of the last update.
ID: Case identification number
SEX: Identifies the patient's gender.
HOSPITAL:Identifies the type of care received by the patient in the unit. It is called outpatient if you went home, or it is called inpatient if you were admitted to hospital.
entry_date: Identifies the date of admission of the patient to the care unit.
date_symptoms: Identifies the date on which the patient's symptoms began.
date_died: Identifies the date the patient died.
intubated: Identifies if the required intubation.
pneumonia: Identifies if the patient has been diagnosed with pneumonia.
age: Identifies the age of the patient.
pregnancy: Identifies if the patient is pregnant.
home_language: Identifies if the patient speaks an indigenous language.
indigenous: Identifies if the patient self-identifies as an indigenous person.

diabetes: Identifies if the patient has a diagnosis of diabetes.
COPD: Identifies if the patient has a diagnosis of COPD (chronic obstructive pulmonary disease)
asthma: Identifies if the patient has a diagnosis of asthma.
INSUMPR: Identifies if the patient has immunosuppression.
HYPERTENSION: Identifies if the patient has a diagnosis of hypertension.
other_diseases: Identifies if the patient has a diagnosis of other diseases.
cardiovascular: Identifies if the patient has a diagnosis of cardiovascular diseases.
obesity: Identifies if the patient has a diagnosis of obesity.
renal_chronic: Identifies if the patient has a diagnosis of chronic renal failure.
smoking: Identifies if the patient has a smoking habit.
ICU: Identifies if the patient required admission to an Intensive Care Unit.

We want to find the survival curve of each month. We only have observed variable when the patient enter the hospital, so we separate the data set into 25 groups for each month by the entry date. We only have 18 days of January 2022, which is the date we collected the data.

3 Methodology

3.1 kaplan-Meier Method

Since the virus of COVID-19 was mutated over time, and the policy of Mexico government was also changed over time, we would like to separate the data into each month to fit a better Survival Curve of the patients who got COVID-19.

We use the data of death date minus the data of symptoms date to get the survival time data of each patients. If the patients did not die, we count the death date as a dummy variable 10000, which the time is much longer than our observed time. Since these patients did not die, we count these as 0 in the death judgment, and for those died patient, we count them as 1.

We now can calculate the estimator of the survival function $S(t)$, which means the probability that life is longer than a certain moment t. This estimator is called Kaplan Meier Estimator, and is given by

$$\hat{S}(x) = \prod_{i:t_i<t} (1 - \frac{d_i}{n_i}),$$

where t_i is a time when at least one event happened, d_i is the number of events (e.g. deaths) that happened at time, n_i is the individuals known to have survived (have not yet had an event or been censored) up to time.

With the Kaplan-Meier Estimator we got, we can consider the survival function for the patients, which can be written as:

$$S(t, X) = P(T > t, X),$$

where $X = (X_1, X_2, ..., X_m)$ is the variable of patients, T is the lifetime of observed patients, and t is a certain moment that we expect.

Since the data set we are considering is the death data of patients. We also would like to know the Death function of the patients, which can be written as

$$F(t, X) = P(T < t, X).$$

This Death function we get is cumulative density function, which is shown the death of patients in the over all days during a month.

To get the death data of patients for each day, we would like to find the death probability density function, which is the derivative of the Death Function

$$f(t, X) = F'(t, X).$$

We can also get the instantaneous mortality rate of the patients X at the time t when the survival time has been reached to this certain time. This function is called Hazard Function

$$h(t, X) = \frac{f(t, X)}{S(t, X)}.$$

We recall that all the above function are estimated based on the Kaplan-Meier Estimator for each month. Thus, we want to get an exact function from the survival data. An easy way is to fit the distribution with using the estimate death probability density curve. From the shape of the estimate distribution curve (left skewness), we would like to consider three distribution: log-normal distribution, gamma distribution, and weibull distribution. To fit these distribution with using the estimate data, we would like to use the way of Maximum Likelihood Estimation (MLE) to fit these three distribution. The likelihood function is defined by

$$L(\theta) = \prod_{i=1}^{n} f(x_i; \theta).$$

To fit the distribution parameters we can use Newton's method. We have parameters λ and α in Weibull distribution. By using Newton's method, we firstly give an initial parameter as λ_0 for λ. Then we get a

$$\lambda_1 = \frac{h(\lambda_0)}{h'(\lambda_0)},$$

where

$$h(\lambda) = \frac{1}{\lambda} + \frac{u}{n} + \frac{w}{v},$$

$$h'(\lambda) = -\frac{1}{\lambda^2} + \frac{w^2}{v^2} + \frac{s}{v},$$

and

$$u = \sum_{i=1}^{n} ln(x_i),$$

$$v = \sum_{i=1}^{n} x_i^{\lambda},$$

$$w = \sum_{i=1}^{n} x_i^{\lambda} ln(x_i),$$

$$s = \sum_{i=1}^{n} x_i^{\lambda} (ln(x_i))^2.$$

We repeat this process until β_k converges, We can also get the parameter of

$$\alpha = (\frac{v}{n})^{\frac{1}{\lambda}}.$$

For Gamma distribution, we also would like to use Newton's Method to fit the distribution. We firstly would like to find the estimation of α. After given a known parameter α_0, the estimation of α_1 to α_k can be given as

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} - \frac{l'(\alpha_{k-1})}{l''(\alpha_{k-1})},$$

where

$$l'(\alpha_{k-1}) = nlog(\frac{\alpha}{\bar{X}}) - n\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} + \sum_{i=1}^{n} log(X_i),$$

$$l''(\alpha_{k-1}) = \frac{n}{\alpha} - n(\frac{\Gamma'(\alpha)}{\Gamma(\alpha)})'.$$

Then we can find the MLE of β as

$$\hat{\beta} = \frac{\bar{X}}{\hat{\alpha}}.$$

We would like to find the estimated parameters for Log-normal Distribution now. The parameters for Log-normal Distribution are μ and σ^2. From the distribution, the likelihood function of the Log-normal Distribution can be given by

$$L(\mu, \sigma|X) = (2\pi\sigma^2)^{-\frac{n}{2}} \prod_{i=1}^{n} X_i^{-1} exp[\sum_{i=1}^{n} \frac{-(\ln(X_i) - \mu)^2}{2\sigma^2}].$$

Thus, the log likelihood function can be given by

$$LL(\mu, \sigma|X) = -\frac{n}{2}\ln(2\pi\sigma^2) - \sum_{i=1}^{n} \ln(X_i) - \frac{\sum_{i=1}^{n} \ln(X_i)^2}{2\sigma^2} + \frac{\sum_{i=1}^{n} \ln(X_i)\mu}{\sigma^2} - \frac{n\mu^2}{2\sigma^2}.$$

By using Newton's method for the LL function, we can find that

$$\hat{\mu} = \frac{\sum_{i=1}^{n} \ln(X_i)}{n},$$

and

$$\hat{\sigma^2} = \frac{\sum_{i=1}^{n} (\ln(X_i) - \hat{\mu})^2}{n}.$$

As well, In R programming, we can use the library of MASS and the function $fitdistr()$ to fit these distribution.

The next step is to compare the distribution curve of these three distribution to get a distribution which fitted best to the estimated curve. For the parameters of the distribution, we can get the following functions for each distribution:

Distribution	Hazard Rate $h(x)$	Survival Function $S(x)$	Probability Density Function $f(x)$
Weibull	$\alpha\lambda x^{\alpha-1}$	$exp[-\lambda x^{\alpha}]$	$\alpha\lambda x^{\alpha-1} exp(-\lambda x^{\alpha})$
Gamma	$\frac{f(x)}{S(x)}$	$1 - I(\lambda x, \beta)$	$\frac{\lambda^{\beta} x^{\beta-1} exp(-\lambda x)}{\Gamma(\beta)}$
Log Normal	$\frac{f(x)}{S(x)}$	$1 - \Phi[\frac{\ln x - \mu}{\sigma}]$	$\frac{exp[-\frac{1}{2}(\frac{\ln x - \mu}{\sigma})^2]}{x(2\pi)^{1/2}\sigma}$

Table 1: Functions for Distributions

3.2 COX Proportional Hazards(COXph) Method

We also want to consider if there are any other factors that related with the death data of COVID-19. We considered Sex, Age and other factors such as the diseases that introduced in the part of Data Source. The way we are using is to consider the COXph Method.When $S(t, X)$ is effected by many factors Xi, We consider regression equation.The response variable is $h(t, X)$ and the function can be write as

$$h(t, X) = h_0(t)exp(\beta_1 X_1 + \beta_2 X_2 + ... + \beta_m X_m),$$

where $h_0(t)$ is the basic risk rate when $X = 0$. It is a semi-parametric regression model, and is not related with the time t. Thus this function can be transformed into

$$ln[h(t, X)/h_0(t)] = lnRR = \beta_1 X_1 + \beta_2 X_2 + ... + \beta_m X_m,$$

and the hazard Ratio (RR) can be written as

$$RR = \frac{h(t, X_i)}{h(t, X_j)} = \frac{h_0(t)exp(\beta' X_i)}{h_0(t)exp(\beta' X_j)} = exp(\beta'(X_i - x_j)).$$

4 Result of data analysis

Since we are analyzing the survival time of patients in this data set, and our goal is to find a way to generate survival cure, we generate the survival time of each patient in this dataset.

4.1 Time Group Survival Curves

Recall that our data goes from Jan 01, 2020 to Jan 18, 2022. We then divided data into 25 groups, depending on what month their first symptom was discovered.Figure **??** shows the number of observations in each group. The first three months have relatively small number of observation than other months, especially first two months.

symptom date	num of observations
Jan-20	12241
Feb-20	12092
Mar-20	43133
Apr-20	120186
May-20	243660
Jun-20	364323
Jul-20	443082
Aug-20	410715
Sep-20	373415
Oct-20	448484
Nov-20	533182
Dec-20	861430
Jan-21	1118956
Feb-21	685987
Mar-21	592554
Apr-21	507616
May-21	408238
Jun-21	476030
Jul-21	1149983

Aug-21	1335249
Sep-21	802870
Oct-21	581389
Nov-21	474157
Dec-21	522055
Jan-22	1224478

Table 2: Number of observations

After divided into groups, data was fit using Kaplan-Meier method. Since our data recorded the date of first symptoms and the exact date of death (or 9999-99-99 if not died), there are no observations are considered as censored. Therefore, Kaplan-Meier estimation should be the same as the empirical distribution of the survival curve for Covid patients in Mexico. Figure 1 shows the survival curve for some selected time groups based on Kaplan-Meier estimates. and figure 2 shows the hazard function. For group of Jul, 2020, Jan, 2021, Jul 2021, and Dec 2021, in a timeline sequence, the survival rate becomes higher and higher. However, survival rate of Jan, 2020 is higher than that of Jul, 2021, but lower than Jan, 2021. That possibly is a result of small number of observations for Jan, 2020.

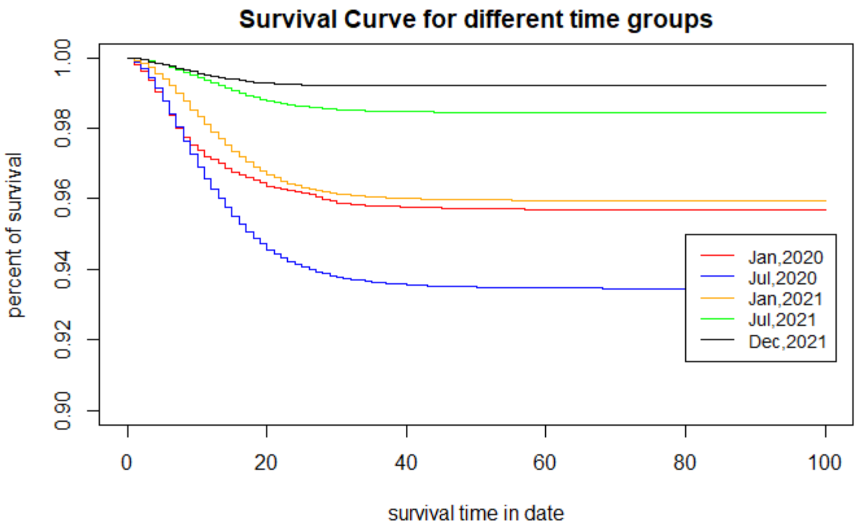


Figure 1: Survival Curve for different time groups

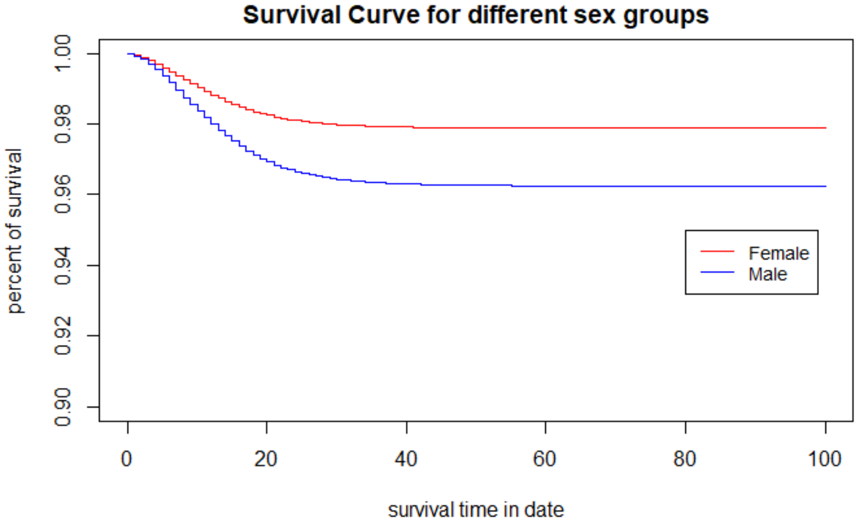


Figure 2: Hazard function for different time groups

Figure 3 shows the probability density function for some selected time groups. It looks like it's not normalized, since patients in our data are not all died. Since for the probability distribution, we are only interested in patients who are died. Therefore, we normalize it into an analysis for patients who are died.

4.2 Parametric model for different time groups

Based on the normalized probability density function, we want to fit a parametric model. Recall that this normalized probability density function is based on Kaplan-Meier estimates. Then, our parametric model tends to fit the Kaplan-Meier estimates.

We have tried several parametric model, but only lognormal distribution and gamma distributions have a comparably good fit. Figure 4 shows the lognormal fit and gamma fit for died patients who has first symptoms discovered in July, 2020. Gamma fit is slightly better than lognormal fit for July, 2020.

We would like to compare these two parametric models. We found a way to compare based on their sum squared of errors. Figure **??** shows the results.

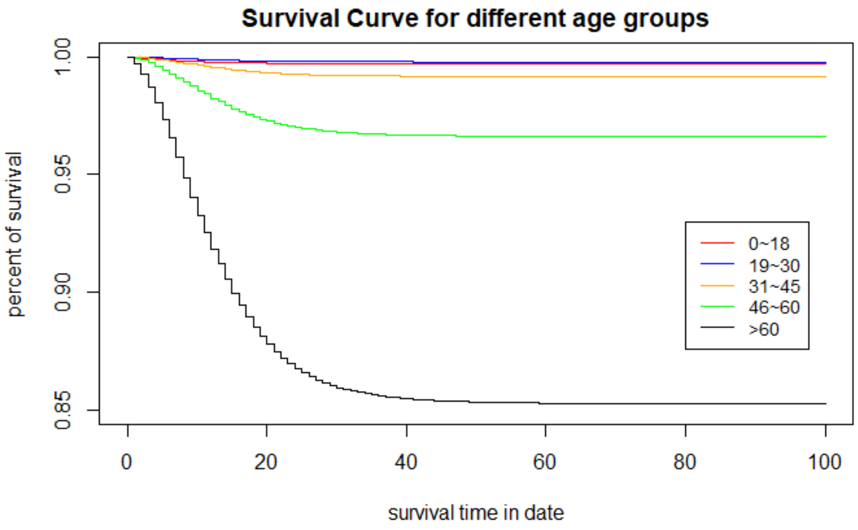


Figure 3: Probability density function for different time groups

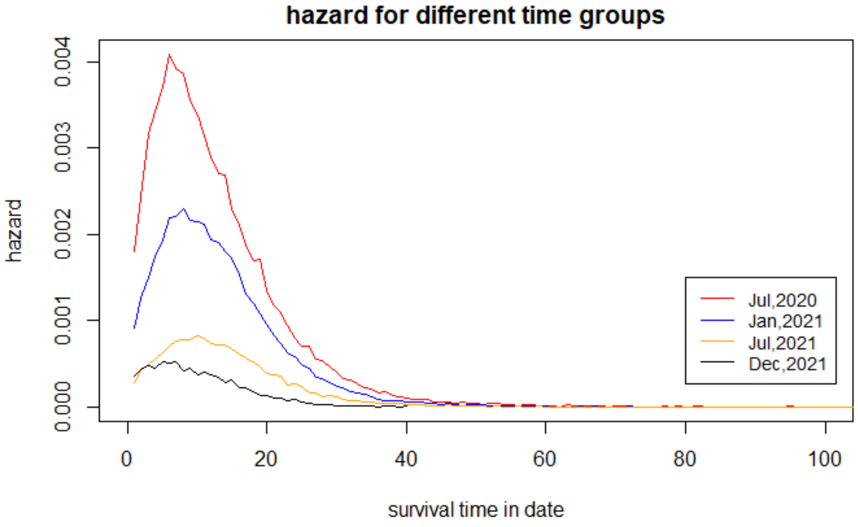


Figure 4: Parametric fit for probability function of July, 2020

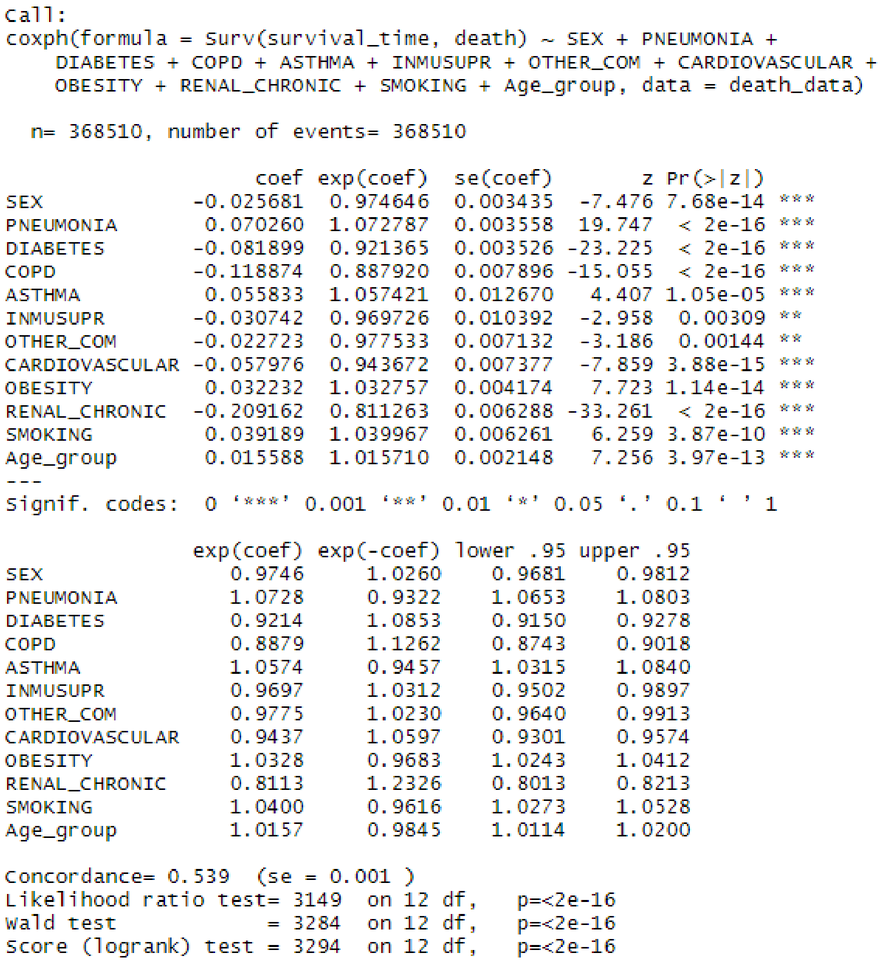
Gamma	Log Normal
0.0053742306	0.002546309
0.0054541423	0.001782083
0.0012494078	0.001648042
0.0008174137	0.001596434
0.0004307609	0.001649047
0.0003000906	0.001786014
0.0001610581	0.001424106
0.0001286677	0.001784706
0.0001629078	0.001963174
0.0001447844	0.002024588
0.0001441141	0.002078876
0.0002092306	0.002144527
0.0002340276	0.002212901
0.0003474512	0.002627428
0.0003059776	0.002460148
0.0002435700	0.002359773
0.0002693895	0.002337075
0.0004021138	0.002554048
0.0004066099	0.002482108
0.0002171116	0.002177117
0.0002224416	0.002229522
0.0002544576	0.002286482
0.0004176083	0.002371456
0.0008566056	0.003771397

Table 3: SSE for Gamma fit and Log normal fit

4.3 COXPH model

We consider only observations which involve dead patients for COXPH model. After data cleaning, we now have observations. Then, we fit a COXPH model into these data. This model has 14 predictors, and then we did a model selection based on AIC. The selected has 13 predictors. Here's a summary of our selected model.

After seeing this summary, we wanted to check the goodness of fit on this COXPH model. We compared the survival curve from COXPH model when we plugged the mean of covariates in the model to the survival curve we got from Kaplan-Meier Estimates.



```
Call:
coxph(formula = Surv(survival_time, death) ~ SEX + PNEUMONIA +
    DIABETES + COPD + ASTHMA + INMUSUPR + OTHER_COM + CARDIOVASCULAR +
    OBESITY + RENAL_CHRONIC + SMOKING + Age_group, data = death_data)

  n= 368510, number of events= 368510

                    coef exp(coef)  se(coef)       z Pr(>|z|)
SEX            -0.025681  0.974646  0.003435  -7.476 7.68e-14 ***
PNEUMONIA       0.070260  1.072787  0.003558  19.747  < 2e-16 ***
DIABETES       -0.081899  0.921365  0.003526 -23.225  < 2e-16 ***
COPD           -0.118874  0.887920  0.007896 -15.055  < 2e-16 ***
ASTHMA          0.055833  1.057421  0.012670   4.407 1.05e-05 ***
INMUSUPR       -0.030742  0.969726  0.010392  -2.958  0.00309 **
OTHER_COM      -0.022723  0.977533  0.007132  -3.186  0.00144 **
CARDIOVASCULAR -0.057976  0.943672  0.007377  -7.859 3.88e-15 ***
OBESITY         0.032232  1.032757  0.004174   7.723 1.14e-14 ***
RENAL_CHRONIC  -0.209162  0.811263  0.006288 -33.261  < 2e-16 ***
SMOKING         0.039189  1.039967  0.006261   6.259 3.87e-10 ***
Age_group       0.015588  1.015710  0.002148   7.256 3.97e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

               exp(coef) exp(-coef) lower .95 upper .95
SEX               0.9746     1.0260    0.9681    0.9812
PNEUMONIA         1.0728     0.9322    1.0653    1.0803
DIABETES          0.9214     1.0853    0.9150    0.9278
COPD              0.8879     1.1262    0.8743    0.9018
ASTHMA            1.0574     0.9457    1.0315    1.0840
INMUSUPR          0.9697     1.0312    0.9502    0.9897
OTHER_COM         0.9775     1.0230    0.9640    0.9913
CARDIOVASCULAR    0.9437     1.0597    0.9301    0.9574
OBESITY           1.0328     0.9683    1.0243    1.0412
RENAL_CHRONIC     0.8113     1.2326    0.8013    0.8213
SMOKING           1.0400     0.9616    1.0273    1.0528
Age_group         1.0157     0.9845    1.0114    1.0200

Concordance= 0.539  (se = 0.001 )
Likelihood ratio test= 3149  on 12 df,   p=<2e-16
Wald test            = 3284  on 12 df,   p=<2e-16
Score (logrank) test = 3294  on 12 df,   p=<2e-16
```

Figure 5: Summary of selected COXPH model

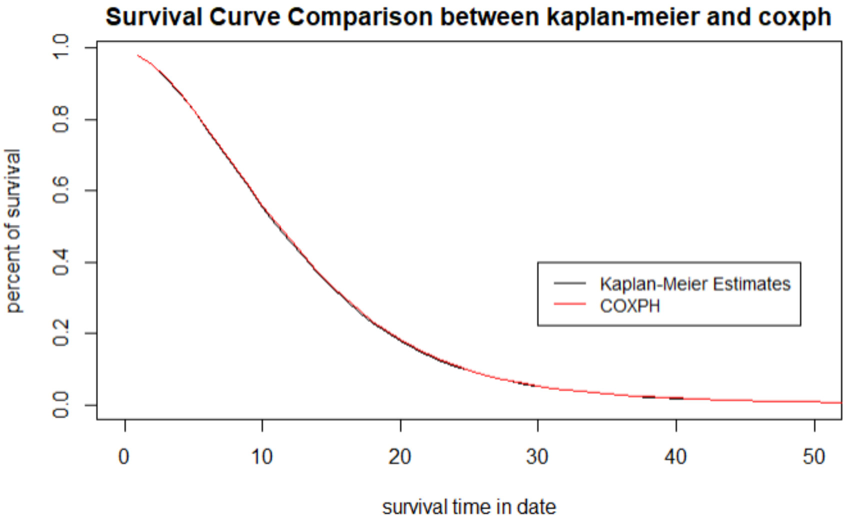


Figure 6: Survival Curve Comparison between Kaplan-Meier and COXPH

4.4 Comparing COXPH model with Parametric model

Firstly, we divided the data into three groups based on their risks, which respectively are low-risk, medium-risk and high-risk. The risks can be calculated from COXPH model using linear predictors. Then, we calculate a survival curve estimates based on the COXPH model for each groups, using the mean of covariates within the group. We also fit a gamma distribution to our grouped dataset and calculate fitted value for survival curve. Then, We compare the sum squared error(SSE) of COXPH model and the Parametric model from Kaplan-Meier Estimates.

The SSE for COXPH model is (0.007481744 0.000129936 0.007050822) for low, median and high respectively, and the SSE for gamma fit is (0.006039999 0.005675512 0.004926047) for low, median and high respectively.

5 Conclusions

5.1 Grouped data analysis

From monthly grouped Kaplan Meier estimate, we can conclude that before March 2020, the survival curve does not have any pattern in a timeline sequence. However, after April 2020, the survival rate become higher and higher in a timeline sequence, which may result from increasingly understanding on how to treat COVID-19 and increasing immunization rate. The hazard rate is increasing in first few days, and then it starts to decrease. The date for it reaching maximum becomes earlier as time goes on. However, the maximum risk for DEC 2021, which is on day 5, is still lower than the risk on day 5 for any other previous month groups.

From the sex grouped data analysis, we can conclude that Female has a higher survival rate on COVID-19 than Male.

In addition, from the age grouped data analysis, we can conclude that smaller age has a higher survival rate on COVID-19

5.2 Parametric model

Based on our comparison results, we can conclude that for the first three months, lognormal has a slightly better fit, but for other months, gamma has a much better fit. The reason for this is that, the data size is small for the first three months. We can also conclude that since we didn't know how to treat Covid patients during first three months, so they might have comparably larger sum square errors, and be closer to lognormal fit. However, based on the experience from the first three months, we got some understanding on how to treat patient, so we could have a much more predictable shape of probability distribution.

5.3 COXPH model and comparison

The COXPH is a good fit for our model since all parameter are having significant p-values. Based on the SSE for comparison, for low and high risk groups, the SSE is smaller for gamma fit, while the SSE is smaller for COXPH model. Therefore, we can conclude that for analysis on low-risk and high-risk patients, gamma parametric model gives us a better survival function. However, for analysis on median-risk patients, COXPH model gives us a better survival function.

References

https://www.gob.mx/salud/documentos/datos-abiertos-152127

Salinas-Escudero, G., Carrillo-Vega, M.F., Granados-García, V. et al. A survival analysis of COVID-19 in the Mexican population. BMC Public Health 20, 1616 (2020). https://doi.org/10.1186/s12889-020-09721-2

R code

```
library(plyr)
library(survival)
library(stats4)
library(MASS)
library(survminer)
library(gmodels)

### GROUP DATA
##read data
data = read.csv("refined_data_220118.csv")
date1 = as.Date(data$ADMISSION_DATE)
date2 = as.Date(data$DATE_SYMPTOMS)
date3 = rep("2050-01-01",13244807)
for (i in 1:13244807){
  if (data$DATE_DEF[i]!="9999-99-99"){
    date3[i] = data$DATE_DEF[i]
  }
}
date3 = as.Date(date3)
#time of patient survival after symptom, 10000 if not died
survival_time = as.numeric(date3-date2)
for (i in 1:13244807){
  if (data$DATE_DEF[i]=="9999-99-99"){
    survival_time[i] = 10000
  }
}

data$survival_time = survival_time

#read cleaned data
survival_time_group1 = read.csv("survival_time_group1.csv")
survival_time_group2 = read.csv("survival_time_group2.csv")
survival_time_group3 = read.csv("survival_time_group3.csv")
survival_time_group4 = read.csv("survival_time_group4.csv")
survival_time_group5 = read.csv("survival_time_group5.csv")
survival_time_group6 = read.csv("survival_time_group6.csv")
survival_time_group7 = read.csv("survival_time_group7.csv")
survival_time_group8 = read.csv("survival_time_group8.csv")
survival_time_group9 = read.csv("survival_time_group9.csv")
survival_time_group10 = read.csv("survival_time_group10.csv")
survival_time_group11 = read.csv("survival_time_group11.csv")
survival_time_group12 = read.csv("survival_time_group12.csv")
```

```
survival_time_group13 = read.csv("survival_time_group13.csv")
survival_time_group14 = read.csv("survival_time_group14.csv")
survival_time_group15 = read.csv("survival_time_group15.csv")
survival_time_group16 = read.csv("survival_time_group16.csv")
survival_time_group17 = read.csv("survival_time_group17.csv")
survival_time_group18 = read.csv("survival_time_group18.csv")
survival_time_group19 = read.csv("survival_time_group19.csv")
survival_time_group20 = read.csv("survival_time_group20.csv")
survival_time_group21 = read.csv("survival_time_group21.csv")
survival_time_group22 = read.csv("survival_time_group22.csv")
survival_time_group23 = read.csv("survival_time_group23.csv")
survival_time_group24 = read.csv("survival_time_group24.csv")
survival_time_group25 = read.csv("survival_time_group25.csv")
survival_sex_group1 = read.csv("survival_sex_group1.csv")
survival_sex_group2 = read.csv("survival_sex_group2.csv")
survival_age_group1 = read.csv("survival_age_group1.csv")
survival_age_group2 = read.csv("survival_age_group2.csv")
survival_age_group3 = read.csv("survival_age_group3.csv")
survival_age_group4 = read.csv("survival_age_group4.csv")
survival_age_group5 = read.csv("survival_age_group5.csv")

#get rid of false data
survival_time_group1 = survival_time_group1[survival_time_group1$survival_time>0,]
survival_time_group2 = survival_time_group2[survival_time_group2$survival_time>0,]
survival_time_group3 = survival_time_group3[survival_time_group3$survival_time>0,]
survival_time_group4 = survival_time_group4[survival_time_group4$survival_time>0,]
survival_time_group5 = survival_time_group5[survival_time_group5$survival_time>0,]
survival_time_group6 = survival_time_group6[survival_time_group6$survival_time>0,]
survival_time_group7 = survival_time_group7[survival_time_group7$survival_time>0,]
survival_time_group8 = survival_time_group8[survival_time_group8$survival_time>0,]
survival_time_group9 = survival_time_group9[survival_time_group9$survival_time>0,]
survival_time_group10 = survival_time_group10[survival_time_group10$survival_time>0,]
survival_time_group11 = survival_time_group11[survival_time_group11$survival_time>0,]
survival_time_group12 = survival_time_group12[survival_time_group12$survival_time>0,]
survival_time_group13 = survival_time_group13[survival_time_group13$survival_time>0,]
survival_time_group14 = survival_time_group14[survival_time_group14$survival_time>0,]
survival_time_group15 = survival_time_group15[survival_time_group15$survival_time>0,]
survival_time_group16 = survival_time_group16[survival_time_group16$survival_time>0,]
survival_time_group17 = survival_time_group17[survival_time_group17$survival_time>0,]
survival_time_group18 = survival_time_group18[survival_time_group18$survival_time>0,]
survival_time_group19 = survival_time_group19[survival_time_group19$survival_time>0,]
survival_time_group20 = survival_time_group20[survival_time_group20$survival_time>0,]
survival_time_group21 = survival_time_group21[survival_time_group21$survival_time>0,]
survival_time_group22 = survival_time_group22[survival_time_group22$survival_time>0,]
survival_time_group23 = survival_time_group23[survival_time_group23$survival_time>0,]
```

```
survival_time_group24 = survival_time_group24[survival_time_group24$survival_time>0,]
survival_time_group25 = survival_time_group25[survival_time_group25$survival_time>0,]
survival_sex_group1 = survival_sex_group1[survival_sex_group1$survival_time>0,]
survival_sex_group2 = survival_sex_group2[survival_sex_group2$survival_time>0,]
survival_age_group1 = survival_age_group1[survival_age_group1$survival_time>0,]
survival_age_group2 = survival_age_group2[survival_age_group2$survival_time>0,]
survival_age_group3 = survival_age_group3[survival_age_group3$survival_time>0,]
survival_age_group4 = survival_age_group4[survival_age_group4$survival_time>0,]
survival_age_group5 = survival_age_group5[survival_age_group5$survival_time>0,]

#km_models
km_model = function(data){
  time = data$survival_time
  death = rep(1,length(time))
  death[time==10000]= 0
  return(survfit(Surv(time,death)˜1,type = "kaplan-meier"))
}
km_model_group1 = km_model(survival_time_group1)
km_model_group2 = km_model(survival_time_group2)
km_model_group3 = km_model(survival_time_group3)
km_model_group4 = km_model(survival_time_group4)
km_model_group5 = km_model(survival_time_group5)
km_model_group6 = km_model(survival_time_group6)
km_model_group7 = km_model(survival_time_group7)
km_model_group8 = km_model(survival_time_group8)
km_model_group9 = km_model(survival_time_group9)
km_model_group10 = km_model(survival_time_group10)
km_model_group11 = km_model(survival_time_group11)
km_model_group12 = km_model(survival_time_group12)
km_model_group13 = km_model(survival_time_group13)
km_model_group14 = km_model(survival_time_group14)
km_model_group15 = km_model(survival_time_group15)
km_model_group16 = km_model(survival_time_group16)
km_model_group17 = km_model(survival_time_group17)
km_model_group18 = km_model(survival_time_group18)
km_model_group19 = km_model(survival_time_group19)
km_model_group20 = km_model(survival_time_group20)
km_model_group21 = km_model(survival_time_group21)
km_model_group22 = km_model(survival_time_group22)
km_model_group23 = km_model(survival_time_group23)
km_model_group24 = km_model(survival_time_group24)
km_model_group25 = km_model(survival_time_group25)
km_model_sex1 = km_model(survival_sex_group1)
km_model_sex2 = km_model(survival_sex_group2)
km_model_age1 = km_model(survival_age_group1)
```

```
km_model_age2 = km_model(survival_age_group2)
km_model_age3 = km_model(survival_age_group3)
km_model_age4 = km_model(survival_age_group4)
km_model_age5 = km_model(survival_age_group5)

#test result
summary(km_model_group1)
plot(km_model_group1,conf.int=F,ylim = c(0.90,1),xlim = c(0,100),col = ’red’,
    main = "Survival Curve for different time groups",xlab = ’survival time in date’,
    ylab = ’percent of survival’)
lines(km_model_group7,conf.int=F,col = ’blue’)
lines(km_model_group13,conf.int=F,col = ’orange’)
lines(km_model_group19,conf.int=F,col = ’green’)
lines(km_model_group24,conf.int=F,col = ’black’)
legend(80, 0.95, legend=c("Jan,2020", "Jul,2020",’Jan,2021’,’Jul,2021’,’Dec,2021’),
       col=c("red", "blue",’orange’,’green’,’black’),lty = 1,cex = 0.85)

plot(km_model_sex1,conf.int=F,ylim = c(0.90,1),xlim = c(0,100),col = ’red’,
    main = "Survival Curve for different sex groups",xlab = ’survival time in date’,
    ylab = ’percent of survival’)
lines(km_model_sex2,conf.int=F,col = ’blue’)
legend(80, 0.95, legend=c("Female",’Male’),
       col=c("red", "blue"),lty = 1,cex = 0.85)

plot(km_model_age1,conf.int=F,ylim = c(0.85,1),xlim = c(0,100),col = ’red’,
    main = "Survival Curve for different age groups",xlab = ’survival time in date’,
    ylab = ’percent of survival’)
lines(km_model_age2,conf.int=F,col = ’blue’)
lines(km_model_age3,conf.int=F,col = ’orange’)
lines(km_model_age4,conf.int=F,col = ’green’)
lines(km_model_age5,conf.int=F,col = ’black’)
legend(80, 0.93, legend=c("0˜18", "19˜30",’31˜45’,’46˜60’,’>60’),
       col=c("red", "blue",’orange’,’green’,’black’),lty = 1,cex = 0.85)

#hazard function
hazard = function(km_model){
  time = summary(km_model)$time
  time = time[1:length(time)-1]
  hazard = numeric(length(time))
  surv = summary(km_model)$surv
  for (i in 1:length(time)){
    hazard[i] = 1-surv[i+1]/surv[i]
  }
  return (list(time = time,hazard = hazard))
}
```

```
hazard1 = hazard(km_model_group1)
hazard2 = hazard(km_model_group2)
hazard3 = hazard(km_model_group3)
hazard4 = hazard(km_model_group4)
hazard5 = hazard(km_model_group5)
hazard6 = hazard(km_model_group6)
hazard7 = hazard(km_model_group7)
hazard8 = hazard(km_model_group8)
hazard9 = hazard(km_model_group9)
hazard10 = hazard(km_model_group10)
hazard11 = hazard(km_model_group11)
hazard12 = hazard(km_model_group12)
hazard13 = hazard(km_model_group13)
hazard14 = hazard(km_model_group14)
hazard15 = hazard(km_model_group15)
hazard16 = hazard(km_model_group16)
hazard17 = hazard(km_model_group17)
hazard18 = hazard(km_model_group18)
hazard19 = hazard(km_model_group19)
hazard20 = hazard(km_model_group20)
hazard21 = hazard(km_model_group21)
hazard22 = hazard(km_model_group22)
hazard23 = hazard(km_model_group23)
hazard24 = hazard(km_model_group24)
hazard25 = hazard(km_model_group25)

plot(hazard7$time,hazard7$hazard,type = ’l’,col = ’red’,xlim = c(0,100),
    main = "hazard for different time groups",xlab = ’survival time in date’,
    ylab = ’hazard’ )
points(hazard13$time,hazard13$hazard,type = ’l’,col = ’blue’)
points(hazard19$time,hazard19$hazard,type = ’l’,col = ’orange’)
points(hazard24$time,hazard24$hazard,type = ’l’,col = ’black’)
legend(80, 0.0015, legend=c( "Jul,2020",’Jan,2021’,’Jul,2021’,’Dec,2021’),
       col=c("red", "blue",’orange’,’black’),lty = 1,cex = 0.85)


#probability density function
f_function = function(km_model){
  time = summary(km_model)$time
  time = time[1:length(time)]
  f = numeric(length(time))
  surv = summary(km_model)$surv
  f[1] = 1-surv[1]
  for (i in 2:length(time)){
    f[i] = surv[i-1]-surv[i]
  }
  return (list(time = time,pdf = f))
}

f1 = f_function(km_model_group1)
f2 = f_function(km_model_group2)
f3 = f_function(km_model_group3)
f4 = f_function(km_model_group4)
f5 = f_function(km_model_group5)
f6 = f_function(km_model_group6)
f7 = f_function(km_model_group7)
f8 = f_function(km_model_group8)
f9 = f_function(km_model_group9)
f10 = f_function(km_model_group10)
f11 = f_function(km_model_group11)
f12 = f_function(km_model_group12)
f13 = f_function(km_model_group13)
f14 = f_function(km_model_group14)
f15 = f_function(km_model_group15)
f16 = f_function(km_model_group16)
f17 = f_function(km_model_group17)
f18 = f_function(km_model_group18)
f19 = f_function(km_model_group19)
f20 = f_function(km_model_group20)
f21 = f_function(km_model_group21)
f22 = f_function(km_model_group22)
f23 = f_function(km_model_group23)
f24 = f_function(km_model_group24)
f25 = f_function(km_model_group25)
fage1 = f_function(km_model_age1)
fage2 = f_function(km_model_age2)
fage3 = f_function(km_model_age3)
fage4 = f_function(km_model_age4)
fage5 = f_function(km_model_age5)
fsex1 = f_function(km_model_sex1)
fsex2 = f_function(km_model_sex2)

plot(f7$time,f7$pdf,type = ’l’,col = ’red’,xlim = c(0,100),
    main = "pdf for different time groups",xlab = ’survival time in date’,
    ylab = ’P’ )
points(f13$time,f13$pdf,type = ’l’,col = ’blue’)
points(f19$time,f19$pdf,type = ’l’,col = ’orange’)
points(f24$time,f24$pdf,type = ’l’,col = ’black’)
legend(80, 0.0015, legend=c( "Jul,2020",’Jan,2021’,’Jul,2021’,’Dec,2021’),
       col=c("red", "blue",’orange’,’black’),lty = 1,cex = 0.85)
```

```
#gamma model using mle
iter_gamma = function(pdf){
  p = round(pdf$pdf*1000000)
  iter = fitdistr(rep(pdf$time,p),"gamma")
  return (iter)
}
iter_logn = function(pdf){
  p = round(pdf$pdf*1000000)
  iter = fitdistr(rep(pdf$time,p),"lognormal")
  return (iter)
}
gamma1 = iter_gamma(f1)
gamma2 = iter_gamma(f2)
gamma3 = iter_gamma(f3)
gamma4 = iter_gamma(f4)
gamma5 = iter_gamma(f5)
gamma6 = iter_gamma(f6)
gamma7 = iter_gamma(f7)
gamma8 = iter_gamma(f8)
gamma9 = iter_gamma(f9)
gamma10 = iter_gamma(f10)
gamma11 = iter_gamma(f11)
gamma12 = iter_gamma(f12)
gamma13 = iter_gamma (f13)
gamma14 = iter_gamma(f14)
gamma15 = iter_gamma(f15)
gamma16 = iter_gamma(f16)
gamma17 = iter_gamma(f17)
gamma18 = iter_gamma(f18)
gamma19 = iter_gamma(f19)
gamma20 = iter_gamma(f20)
gamma21 = iter_gamma(f21)
gamma22 = iter_gamma(f22)
gamma23 = iter_gamma(f23)
gamma24 = iter_gamma(f24)
gamma25 = iter_gamma(f25)

logn1 = iter_logn(f1)
logn2 = iter_logn(f2)
logn3 = iter_logn(f3)
logn4 = iter_logn(f4)
logn5 = iter_logn(f5)
logn6 = iter_logn(f6)
logn7 = iter_logn(f7)
```

```
logn8 = iter_logn(f8)
logn9 = iter_logn(f9)
logn10 = iter_logn(f10)
logn11 = iter_logn(f11)
logn12 = iter_logn(f12)
logn13 = iter_logn(f13)
logn14 = iter_logn(f14)
logn15 = iter_logn(f15)
logn16 = iter_logn(f16)
logn17 = iter_logn(f17)
logn18 = iter_logn(f18)
logn19 = iter_logn(f19)
logn20 = iter_logn(f20)
logn21 = iter_logn(f21)
logn22 = iter_logn(f22)
logn23 = iter_logn(f23)
logn24 = iter_logn(f24)
logn25 = iter_logn(f25)

find_residuals = function(logn,g,f){
  gamma_fit = dgamma(1:40,shape=g$estimate[1],rate =g$estimate[2])
  logn_fit = dlnorm(1:40,meanlog = logn$estimate[1],sdlog = logn$estimate[2])
  result = f$pdf/sum(f$pdf)
  result = result[1:40]
  gamma_residuals = sum((gamma_fit-result)^2)
  logn_residuals = sum((logn_fit-result)^2)
  return (c(gamma_residuals,logn_residuals))
}

ssr = find_residuals(logn1,gamma1,f1)
ssr = rbind(ssr,find_residuals(logn2,gamma2,f2))
ssr = rbind(ssr,find_residuals(logn3,gamma3,f3))
ssr = rbind(ssr,find_residuals(logn4,gamma4,f4))
ssr = rbind(ssr,find_residuals(logn5,gamma5,f5))
ssr = rbind(ssr,find_residuals(logn6,gamma6,f6))
ssr = rbind(ssr,find_residuals(logn7,gamma7,f7))
ssr = rbind(ssr,find_residuals(logn8,gamma8,f8))
ssr = rbind(ssr,find_residuals(logn9,gamma9,f9))
ssr = rbind(ssr,find_residuals(logn10,gamma10,f10))
ssr = rbind(ssr,find_residuals(logn11,gamma11,f11))
ssr = rbind(ssr,find_residuals(logn12,gamma12,f12))
ssr = rbind(ssr,find_residuals(logn13,gamma13,f13))
ssr = rbind(ssr,find_residuals(logn14,gamma14,f14))
ssr = rbind(ssr,find_residuals(logn15,gamma15,f15))
ssr = rbind(ssr,find_residuals(logn16,gamma16,f16))
ssr = rbind(ssr,find_residuals(logn17,gamma17,f17))
ssr = rbind(ssr,find_residuals(logn18,gamma18,f18))
ssr = rbind(ssr,find_residuals(logn19,gamma19,f19))
ssr = rbind(ssr,find_residuals(logn20,gamma20,f20))
ssr = rbind(ssr,find_residuals(logn21,gamma21,f21))
ssr = rbind(ssr,find_residuals(logn22,gamma22,f22))
ssr = rbind(ssr,find_residuals(logn23,gamma23,f23))
ssr = rbind(ssr,find_residuals(logn24,gamma24,f24))
```

```
plot(f7$time,f7$pdf/sum(f7$pdf),type = "l",xlim = c(0,50),ylim=c(0,0.1),
    main = "pdf for July,2020",xlab = ’survival time in date’,ylab = ’P’ )
points(1:100,dgamma(1:100,shape=gamma7$estimate[1],rate =gamma7$estimate[2]),
    ylim=c(0,0.1),col = "red",type = "l")
points(1:100,dlnorm(1:100,meanlog = lognormal7$estimate[1],
    sdlog = lognormal7$estimate[2]),
    ylim=c(0,0.1),col = "blue",type = "l")
legend(40, 0.03, legend=c( "gamma",’lognormal’,’original’),
       col=c("red", "blue","black"),lty = 1,cex = 0.85)
```

```
gamma_estimates = data.frame(shape = numeric(25),rate = numeric(25))
gamma_estimates$shape[1] = gamma1$estimate[1]
gamma_estimates$shape[2] = gamma2$estimate[1]
gamma_estimates$shape[3] = gamma3$estimate[1]
gamma_estimates$shape[4] = gamma4$estimate[1]
gamma_estimates$shape[5] = gamma5$estimate[1]
gamma_estimates$shape[6] = gamma6$estimate[1]
gamma_estimates$shape[7] = gamma7$estimate[1]
gamma_estimates$shape[8] = gamma8$estimate[1]
gamma_estimates$shape[9] = gamma9$estimate[1]
gamma_estimates$shape[10] = gamma10$estimate[1]
gamma_estimates$shape[11] = gamma11$estimate[1]
gamma_estimates$shape[12] = gamma12$estimate[1]
gamma_estimates$shape[13] = gamma13$estimate[1]
gamma_estimates$shape[14] = gamma14$estimate[1]
gamma_estimates$shape[15] = gamma15$estimate[1]
gamma_estimates$shape[16] = gamma16$estimate[1]
gamma_estimates$shape[17] = gamma17$estimate[1]
gamma_estimates$shape[18] = gamma18$estimate[1]
gamma_estimates$shape[19] = gamma19$estimate[1]
gamma_estimates$shape[20] = gamma20$estimate[1]
gamma_estimates$shape[21] = gamma21$estimate[1]
```

```
gamma_estimates$shape[22] = gamma22$estimate[1]
gamma_estimates$shape[23] = gamma23$estimate[1]
gamma_estimates$shape[24] = gamma24$estimate[1]
gamma_estimates$shape[25] = gamma25$estimate[1]

gamma_estimates$rate[1] = gamma1$estimate[2]
gamma_estimates$rate[2] = gamma2$estimate[2]
gamma_estimates$rate[3] = gamma3$estimate[2]
gamma_estimates$rate[4] = gamma4$estimate[2]
gamma_estimates$rate[5] = gamma5$estimate[2]
gamma_estimates$rate[6] = gamma6$estimate[2]
gamma_estimates$rate[7] = gamma7$estimate[2]
gamma_estimates$rate[8] = gamma8$estimate[2]
gamma_estimates$rate[9] = gamma9$estimate[2]
gamma_estimates$rate[10] = gamma10$estimate[2]
gamma_estimates$rate[11] = gamma11$estimate[2]
gamma_estimates$rate[12] = gamma12$estimate[2]
gamma_estimates$rate[13] = gamma13$estimate[2]
gamma_estimates$rate[14] = gamma14$estimate[2]
gamma_estimates$rate[15] = gamma15$estimate[2]
gamma_estimates$rate[16] = gamma16$estimate[2]
gamma_estimates$rate[17] = gamma17$estimate[2]
gamma_estimates$rate[18] = gamma18$estimate[2]
gamma_estimates$rate[19] = gamma19$estimate[2]
gamma_estimates$rate[20] = gamma20$estimate[2]
gamma_estimates$rate[21] = gamma21$estimate[2]
gamma_estimates$rate[22] = gamma22$estimate[2]
gamma_estimates$rate[23] = gamma23$estimate[2]
gamma_estimates$rate[24] = gamma24$estimate[2]
gamma_estimates$rate[25] = gamma25$estimate[2]
gamma_estimates


gamma_age1 = iter_gamma(fage1)
gamma_age2 = iter_gamma(fage2)
gamma_age3 = iter_gamma(fage3)
gamma_age4 = iter_gamma(fage4)
gamma_age5 = iter_gamma(fage5)
gamma_estimates_age = data.frame(shape = numeric(5),rate = numeric(5))
gamma_estimates_age$shape[1] = gamma_age1$estimate[1]
gamma_estimates_age$shape[2] = gamma_age2$estimate[1]
gamma_estimates_age$shape[3] = gamma_age3$estimate[1]
gamma_estimates_age$shape[4] = gamma_age4$estimate[1]
gamma_estimates_age$shape[5] = gamma_age5$estimate[1]
gamma_estimates_age$rate[1] = gamma_age1$estimate[2]
```

```
gamma_estimates_age$rate[2] = gamma_age2$estimate[2]
gamma_estimates_age$rate[3] = gamma_age3$estimate[2]
gamma_estimates_age$rate[4] = gamma_age4$estimate[2]
gamma_estimates_age$rate[5] = gamma_age5$estimate[2]
gamma_estimates_age

plot(fage2$time,fage2$pdf/sum(fage2$pdf),type = "l",xlim = c(0,50),
    ylim=c(0,0.1))
points(1:100,dgamma(1:100,shape=gamma_age2$estimate[1],
    rate =gamma_age2$estimate[2]),
    ylim=c(0,0.1),col = "red")

#coxph data cleaning
death_data = data[data$survival_time>0&data$survival_time!=10000,]
death_data$death = rep(1,383896)

death_data$Age_group = rep(1,383896)
death_data$Age_group[death_data$AGE>=18&death_data$AGE<30] = 2
death_data$Age_group[death_data$AGE>=30&death_data$AGE<45] = 3
death_data$Age_group[death_data$AGE>=45&death_data$AGE<60] = 4
death_data$Age_group[death_data$AGE>=60] = 5
death_data$INTUBATED[death_data$INTUBATED!=1&death_data$INTUBATED!=2]=NA
death_data$PNEUMONIA[death_data$PNEUMONIA!=1&death_data$PNEUMONIA!=2]=NA
#death_data$PREGNANCY[death_data$PREGNANCY!=1&death_data$PREGNANCY!=2]=NA
death_data$home_language[death_data$home_language==99]=NA
death_data$INDIGENOUS[death_data$INDIGENOUS!=1&death_data$INDIGENOUS!=2]=NA
death_data$DIABETES[death_data$DIABETES==98]=NA
death_data$COPD[death_data$COPD==98]=NA
death_data$ASTHMA[death_data$ASTHMA!=1&death_data$ASTHMA!=2]=NA
death_data$INMUSUPR[death_data$INMUSUPR!=1&death_data$INMUSUPR!=2]=NA
death_data$HYPERTENSION[death_data$HYPERTENSION!=1&death_data$HYPERTENSION!=2]=NA
death_data$OTHER_COM[death_data$OTHER_COM!=1&death_data$OTHER_COM!=2]=NA
death_data$CARDIOVASCULAR[death_data$CARDIOVASCULAR!=1&death_data$CARDIOVASCULAR!=2]=NA
death_data$OBESITY[death_data$OBESITY!=1&death_data$OBESITY!=2]=NA
death_data$RENAL_CHRONIC[death_data$RENAL_CHRONIC!=1&death_data$RENAL_CHRONIC!=2]=NA
death_data$SMOKING[death_data$SMOKING!=1&death_data$SMOKING!=2]=NA

death_data = death_data[-c(1,2,3,5,6,7,8,9,11,12,26)]
death_data = na.omit(death_data)

coxph_wholemodel = coxph(Surv(survival_time,death)~SEX+Age_group+PNEUMONIA+home_language
    +INDIGENOUS+DIABETES+COPD+ASTHMA+INMUSUPR+HYPERTENSION+OTHER_COM+CARDIOVASCULAR+OBESITY
    +RENAL_CHRONIC+SMOKING,data = death_data)
summary(coxph_wholemodel)
```

```
#model selection
step(coxph(Surv(survival_time,death)~1,data=death_data),direction = "forward")
stepAIC(coxph(Surv(survival_time,death)~1,data=death_data),
    scope = formula(coxph_wholemodel),direction = "forward",k=2)
#the result is without variable hypertension, so we will use this model

selected_model = coxph(Surv(survival_time,death)~SEX+PNEUMONIA+DIABETES+COPD+
    ASTHMA+INMUSUPR+OTHER_COM+CARDIOVASCULAR+OBESITY+RENAL_CHRONIC+SMOKING+
    Age_group,data = death_data)
summary(selected_model)
mean_covariates = data.frame(SEX = mean(death_data$SEX),Age_group =
    mean(death_data$Age_group),PNEUMONIA = mean(death_data$PNEUMONIA),
    home_language=mean(death_data$home_language),INDIGENOUS=
    mean(death_data$INDIGENOUS),DIABETES=mean(death_data$DIABETES),
    COPD=mean(death_data$COPD),ASTHMA=mean(death_data$ASTHMA),
    INMUSUPR=mean(death_data$INMUSUPR),HYPERTENSION=mean(death_data$HYPERTENSION),
    OTHER_COM=mean(death_data$OTHER_COM),CARDIOVASCULAR=mean(death_data$CARDIOVASCULAR)
    ,OBESITY=mean(death_data$OBESITY),RENAL_CHRONIC=death_data$RENAL_CHRONIC,
    SMOKING=mean(death_data$SMOKING))
fit = survfit(selected_model,newdata = mean_covariates)
km_deathdata = km_model(death_data)
plot(km_deathdata$surv,type = "l",xlim = c(0,50),main = "Survival Curve Comparison
    between kaplan-meier and coxph",xlab = ’survival time in date’,
    ylab = ’percent of survival’)
legend(30, 0.4, legend=c("Kaplan-Meier Estimates",’COXPH’),
       col=c("black", "red"),lty = 1,cex = 0.85)

points(1:100,fit$surv[1:100],type = "l",col = "red")
```