

PRELIMINARY ANALYSIS: Princess Diamond Corona Virus Risk Analysis

Introduction

This report is a preliminary analysis of the risk of infection (symptomatic and asymptomatic) and death (fatality rate or FR) from the corona virus. The analysis is based on the data derived from the Diamond Princess, a cruise ship which experienced an outbreak of the virus in February 2020. The passengers and crew were forced into a quarantine on the boat for a period of time. The quarantine had the unintended consequence of exposing the people to the virus and dramatically increasing the rates of infection. In total, there were 3,711 people on the boat, 619 became infected and to-date 7 have died.

The current date of this analysis is March 10, 2020.

The Diamond Princess data is unique in that all (or almost all) the passengers and crew were tested for the virus. As far as I know, this is the only meaningful data set where the denominator of the corona risk rates is known. However, there are still a significant number of passengers who have not recovered, or worse, are still in critical condition. Therefore, the numerator of the risk rates of death is only partially known. On the other hand, the risk rates of infection are known.

Warning: This analysis is a “naive” analysis, in that I do not adjust for future expected deaths. The fatality rates therefore represent a floor. I will comment on this further in the report.

Summary of Results

- The CFR for those below age 60 is very low. The fatality rate increases from age 60 and is very high for those age 70 and above: about 1% for 70s, 8% for the 80s and 16% in the 90s. That is, mortality risk is highly skewed towards the elderly.
- The risk of infection is high across all ages. The belief that young children or young adults are not susceptible to infection is false.
- The risk of becoming symptomatic is also high across all ages.
- The risk of symptomatic and asymptomatic infection increases significantly from age 50 onwards
- Roughly half the infected population is asymptomatic.

Methodology

Software: I used Greta (<https://greta-stats.org/index.html>) and Greta GP (<https://github.com/greta-dev/greta.gp>) to conduct a Bayesian analysis.

Models: The *basic model* is binomial model with a uniform prior assumption on the risk probability. For the youngest and very oldest ages there is very little data and this prior influences the results.

I developed a model combining all age groups and a model that analyzed each age group separately. For the latter, I used two types of models: 1. The first model assumes each age category is independent from the others. 2. The second model assumes a correlation structure between each age category. This was achieved by applying a latent Gaussian Process over the age-dependent risk parameters, using a RBF (radial basis function) kernel. I tested two GP models.

2.1. The first fixed the ρ parameter of the RBF at 10 years. That is, the correlation between age x and $x+d$ is $e^{-(\frac{d}{\rho})^2}$. So if $d = 10$ and $\rho = 10$, the correlation between two adjacent 10-year age buckets is $e^{-(\frac{10}{10})^2} = 0.36$. If $\rho = 20$, the correlation is 0.778. The tested model fixed $\rho = 10$. 2.2. The second model, treated ρ as a parameter. For this second model, I set the prior for ρ to have a mean of 20 and a standard deviation of 5. The model estimated the posterior mean value of ρ of 22 with a 95% credible interval of 14 to 35.

The GP Model under 2.2 represents an *advanced model* and I think gives a more accurate analysis of the fatality rates. The GP model has the advantage of smoothing out the risk rates across the age buckets and recognizing that the risk rates between age buckets is almost certainly correlated.

Recommended Actions

Disclaimer: I am not an epidemiologist so my analysis is very much one of an armchair scientist. However, I am an actuary, so I know a thing or two about risk rates.

The analysis suggests the following:

- There will be a lot of people walking around asymptomatic but capable of spreading the disease. Social distancing will be key to avoiding the spread of the disease. Also, if face masks are helpful in stopping an infected person from spreading the diseases, then for sure, people in crowds should be wearing masks even if they feel healthy. The Surgeon General's suggestion not to wear a mask assuming they are available (which they are currently not) is ridiculous.
- The elderly need to exhibit extreme caution.
- Even if you are healthy and young, the risk of an outbreak on a cruise line is very high and the chance of being forced into an involuntary quarantine seems, to me, to be higher than I would be willing to bear.
- Likewise, avoiding large gatherings would be preferred. We are starting to see the closing of college classes and schools. I expect this to accelerate.

Future Deaths

Based on my quick analysis, it would not surprise me if there will be additional deaths in the exposed population. Most of the infections have now occurred at least 20 days ago. I would expect the fatality rates to possibly climb proportionately by 20% to 50% (e.g. a 10% "naive" fatality rates could translate to an ultimate 12% to 15% rate), but am doubtful these ultimate fatality rates would double over the naive levels. This area still need further analysis (see Russel below for more information).

However, the analysis below is very useful in its own right, even ignoring future deaths.

The next sections go through the detail of the code and the analysis. To skip to the final results, go to the "*Summary of Results*" section at the end of the report.

Data Sources

Russel et al (https://cmmid.github.io/topics/covid19/severity/diamond_cruise_cfr_estimates.html) National Institute of Infectious Diseases (<https://www.niid.go.jp/niid/en/2019-ncov-e.html>) Wikipedia (https://en.wikipedia.org/wiki/2020_coronavirus_outbreak_on_cruise_ships#Diamond_Princess)

Note that Russel conducts a similar analysis but adjusts for outstanding deaths. My data of the age distribution differ from Russel and are based on information from Wikipedia. There are 7 deaths, 4 in the 80s, 2 in the 70s and 1 unknown. I assigned the unknown proportionately to the 80s and 70s buckets.

Initialize Data

```
# Time Series Analysis of Survival from Breast Cancer

library(greta)
library(greta.gp)
library(ggplot2)
library(dplyr)
library(purrr)
library(tidyr)
library(readr)
library(stringr)
library(bayesplot)

# MCMC Pars
n_samples_base = 1000
warmup_base = 250
chains_base = 20

# Set Directories
working_dir = paste0(getwd(), "/")
data_dir = paste0(working_dir, "data/")

# Load Data
diamond = read_csv(file = paste0(data_dir, "diamond.csv"))
head(diamond, 10)
```

| group <chr> | age <dbl> | symptomatic <dbl> | asymptomatic <dbl> | confirmed <dbl> | exposure <dbl> | deaths <dbl> |
|----------------|--------------|----------------------|-----------------------|--------------------|-------------------|-----------------|
| 00-09 | 5 | 0 | 1 | 1 | 16 | 0.000000 |
| 10-19 | 15 | 2 | 3 | 5 | 23 | 0.000000 |
| 20-29 | 25 | 25 | 3 | 28 | 347 | 0.000000 |
| 30-39 | 35 | 37 | 7 | 34 | 428 | 0.000000 |
| 40-49 | 45 | 19 | 8 | 27 | 334 | 0.000000 |
| 50-59 | 55 | 28 | 31 | 59 | 398 | 0.000000 |
| 60-69 | 65 | 76 | 101 | 177 | 923 | 0.000000 |
| 70-79 | 75 | 95 | 139 | 234 | 1015 | 2.333333 |
| 80-89 | 85 | 27 | 25 | 52 | 216 | 4.666667 |
| 90-99 | 95 | 2 | 0 | 2 | 11 | 0.000000 |

1-10 of 10 rows

Infection Rate Analysis

Greta Infection Rate Analysis

Beta Binom Model: Single Parameter

```
# Identify the data input for Greta using the as_data function

categories=1
infected = as_data(diamond$confirmed)
exposure = as_data(diamond$exposure)
deaths = as_data(diamond$deaths)
symptomatic = as_data(diamond$symptomatic)
asymptomatic = as_data(diamond$asymptomatic)

# Define the prior for the theta parameter
theta_infected = beta(shape1 = 1, shape2 = 1, dim = categories)
theta_symptomatic = beta(shape1 = 1, shape2 = 1, dim = categories)
theta_asymptomatic = beta(shape1 = 1, shape2 = 1, dim = categories)
theta_deaths = beta(shape1 = 1, shape2 = 1, dim = categories)

# Define the Likelihood for the model
distribution(infected) = binomial(size = exposure, prob = theta_infected)
distribution(symptomatic) = binomial(size = exposure, prob = theta_symptomatic)
distribution(asymptomatic) = binomial(size = exposure, prob = theta_asymptomatic)
distribution(deaths) = binomial(size = infected, prob = theta_deaths)
```

We can now establish and compile the model:

```
# Establish the model
m= model(theta_infected, theta_symptomatic, theta_asymptomatic, theta_deaths)

n_samples = n_samples_base; chains=chains_base; warmup=warmup_base
S = n_samples * chains # Total number of simulations
draws=mcmc(m, n_samples = n_samples, warmup = warmup, chains = chains)
```

```
##
## running 20 chains simultaneously on up to 12 cores
```

```
##
warmup                                0/250 | eta:  ?s
warmup =====                      50/250 | eta: 19s
warmup =====                      100/250 | eta: 12s
warmup =====                      150/250 | eta:  8s
warmup =====                      200/250 | eta:  4s
warmup =====                      250/250 | eta:  0s

##
sampling                              0/1000 | eta:  ?s
sampling ==                          50/1000 | eta: 46s
sampling ====                        100/1000 | eta:  1m
sampling =====                    150/1000 | eta:  1m
sampling =====                    200/1000 | eta:  1m
sampling =====                    250/1000 | eta:  1m
sampling =====                    300/1000 | eta:  1m
sampling =====                    350/1000 | eta:  1m
sampling =====                    400/1000 | eta: 48s
sampling =====                    450/1000 | eta: 45s
sampling =====                    500/1000 | eta: 41s
sampling =====                    550/1000 | eta: 37s
sampling =====                    600/1000 | eta: 33s
sampling =====                    650/1000 | eta: 29s
sampling =====                    700/1000 | eta: 24s
sampling =====                    750/1000 | eta: 20s
sampling =====                    800/1000 | eta: 16s
sampling =====                    850/1000 | eta: 12s
sampling =====                    900/1000 | eta:  8s
sampling =====                    950/1000 | eta:  4s
sampling =====                   1000/1000 | eta:  0s
```

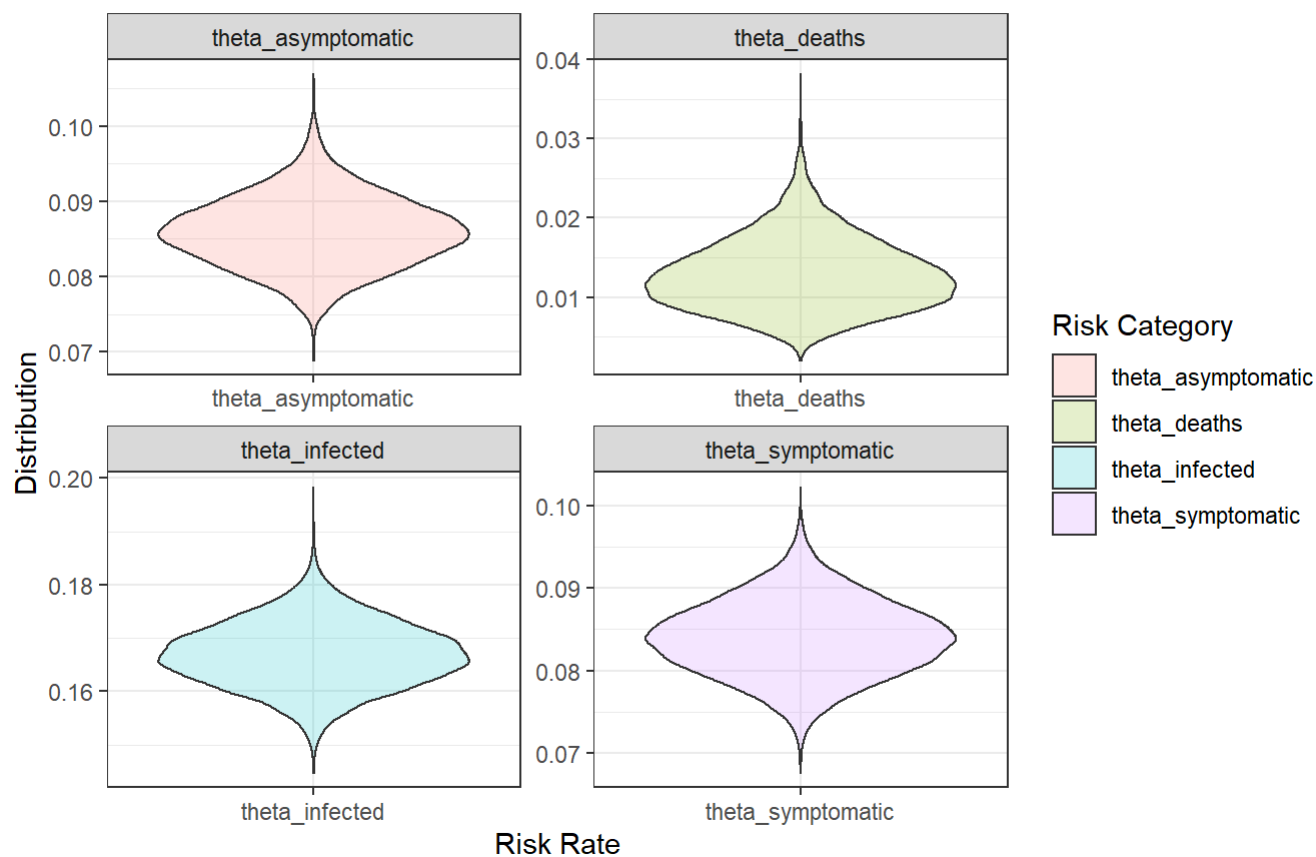
Organize Output for Plotting

```
theta_post =
  draws %>%
  reduce(rbind)
colnames(theta_post) =
  paste0(rep(c("theta_infected_", "theta_symptomatic_", "theta_asymptomatic_", "theta_deaths_"),
    each=categories) , 1:categories)
theta_post = cbind(S=1:S, as_tibble(theta_post)) %>%
  gather(posterior, value, -S) %>%
  mutate(risk_cat = as.factor(gsub(pattern = "_([_]*)$", "", (posterior))),
    age_cat = as.factor((as.numeric(extract_numeric(posterior)))*10-5)
  )
```

extract_numeric() is deprecated: please use readr::parse_number() instead

```
combined_risk_rates = ggplot(theta_post) +
  geom_violin(aes(risk_cat, value, fill=risk_cat), alpha=0.2) +
  labs(title="Diamond Princess: Overall Risks\nInfection, Symptomatic, Asymptomatic, Death", col
or="", x="Risk Rate", y="Distribution", fill="Risk Category") +
  theme_bw()+
  facet_wrap(~risk_cat, scales="free")
combined_risk_rates
```

Diamond Princess: Overall Risks Infection, Symptomatic, Asymptomatic, Death



Beta Binom Model: Vector Parameter

This model assumes that each age category is independent from the others.

```

# Identify the data input for Greta using the as_data function

categories = nrow(diamond)
exposure = as_data(diamond$exposure)

infected = as_data(diamond$confirmed)
symptomatic= as_data(diamond$symptomatic)
asymptomatic= as_data(diamond$asymptomatic)
deaths = as_data(diamond$deaths)

# Define the prior for the theta parameter
theta_infected = beta(shape1 = 1, shape2 = 1, dim = categories)
theta_symptomatic = beta(shape1 = 1, shape2 = 1, dim = categories)
theta_asymptomatic = beta(shape1 = 1, shape2 = 1, dim = categories)
theta_deaths = beta(shape1 = 1, shape2 = 1, dim = categories)

# Define the Likelihood for the model
distribution(infected) = binomial(size = exposure, prob = theta_infected, dim = categories)
distribution(symptomatic) = binomial(size = exposure, prob = theta_symptomatic, dim = categories)
distribution(asymptomatic) = binomial(size = exposure, prob = theta_asymptomatic, dim = categories)
distribution(deaths) = binomial(size = infected, prob = theta_deaths, dim = categories)

```

We can now establish and compile the model:

```

# Establish the model
m_vec= model(theta_infected, theta_symptomatic, theta_asymptomatic, theta_deaths)

n_samples = n_samples_base; chains=chains_base; warmup=warmup_base

S = n_samples * chains # Total number of simulations
draws_vec=mcmc(m_vec, n_samples = n_samples, warmup = warmup, chains = chains)

```

```

##
## running 20 chains simultaneously on up to 12 cores

```

```
##
warmup                                0/250 | eta:  ?s
warmup =====                      50/250 | eta: 19s
warmup =====                      100/250 | eta: 13s
warmup =====                      150/250 | eta:  8s | 4% bad
warmup =====                      200/250 | eta:  4s | 3% bad
warmup =====                      250/250 | eta:  0s | 2% bad

##
sampling                              0/1000 | eta:  ?s
sampling ==                          50/1000 | eta: 49s
sampling ====                        100/1000 | eta:  1m
sampling =====                    150/1000 | eta:  1m
sampling =====                    200/1000 | eta:  1m
sampling =====                    250/1000 | eta: 47s
sampling =====                    300/1000 | eta: 46s
sampling =====                    350/1000 | eta: 45s
sampling =====                    400/1000 | eta: 42s
sampling =====                    450/1000 | eta: 40s
sampling =====                    500/1000 | eta: 37s
sampling =====                    550/1000 | eta: 33s
sampling =====                    600/1000 | eta: 29s
sampling =====                    650/1000 | eta: 26s
sampling =====                    700/1000 | eta: 23s
sampling =====                    750/1000 | eta: 19s
sampling =====                    800/1000 | eta: 15s
sampling =====                    850/1000 | eta: 11s
sampling =====                    900/1000 | eta:  7s
sampling =====                    950/1000 | eta:  4s
sampling =====                   1000/1000 | eta:  0s
```

Organize Output for Plotting

```
theta_post_vec =
  draws_vec %>%
  reduce(rbind)
colnames(theta_post_vec) =
  paste0(rep(c("theta_infected_", "theta_symptomatic_", "theta_asymptomatic_", "theta_deaths_"),
    each=categories) , 1:categories)
theta_post_vec = cbind(S=1:S, as_tibble(theta_post_vec)) %>%
  gather(posterior, value, -S) %>%
  mutate(risk_cat = as.factor(gsub(pattern = "_([_])*$", "", (posterior))),
    age_cat = as.factor((as.numeric(extract_numeric(posterior)))*10-5)
  )
```

extract_numeric() is deprecated: please use readr::parse_number() instead

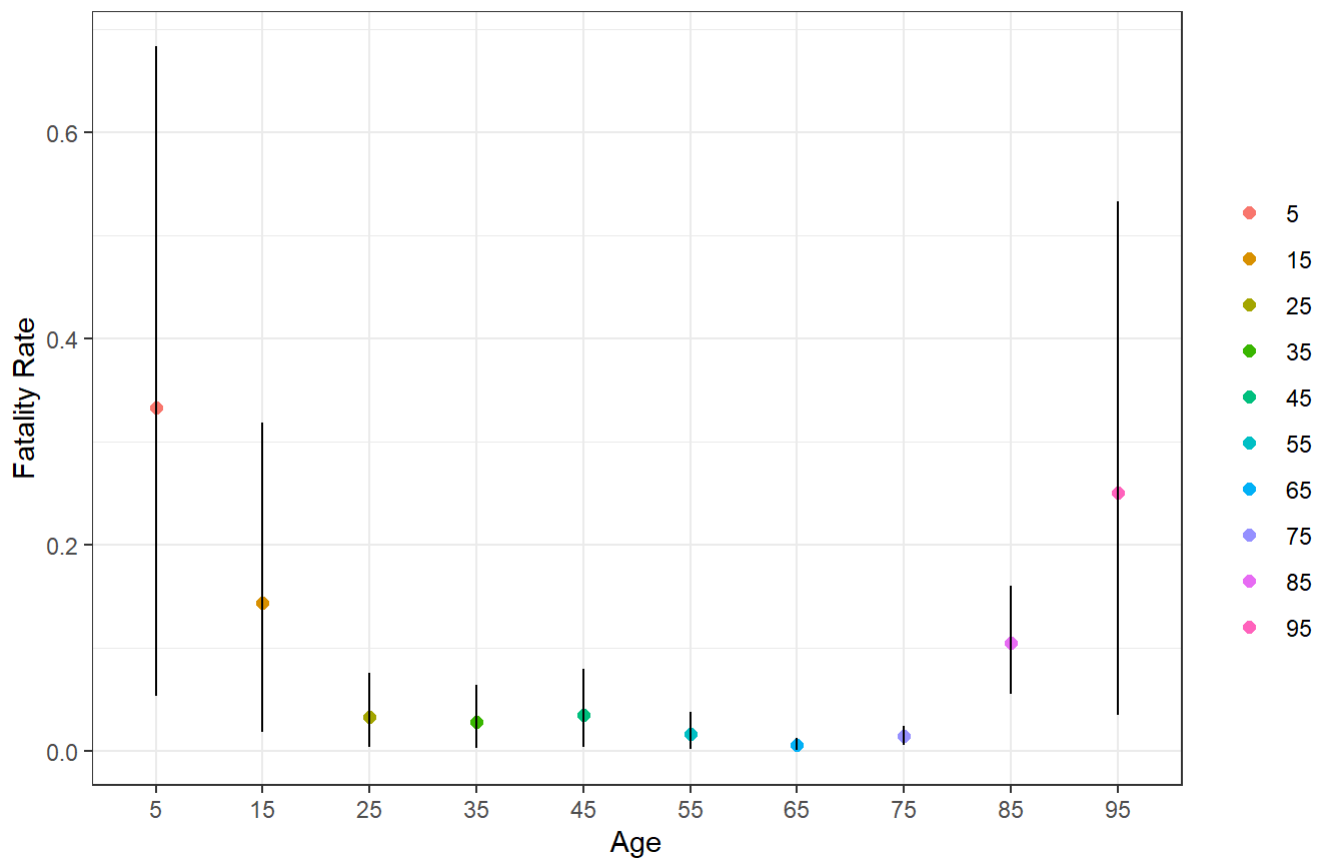

```

theta_post_vec_sum = theta_post_vec %>%
  filter(risk_cat == "theta_deaths") %>%
  group_by(age_cat) %>%
  summarize(mean = mean(value),
            low = quantile(value, 0.1),
            high = quantile(value, 0.9)
            ) %>%
  gather(range, value, -mean, -age_cat)

ggplot(theta_post_vec_sum) +
  geom_point(aes(age_cat, mean, color = age_cat), size=2) +
  geom_line(aes(age_cat, value, group = age_cat)) +
  labs(title="Diamond Princess: Risk of Death\n with 80% Credible Interval", color="", y="Fatali
ty Rate", x="Age") +
  theme_bw()

```

**Diamond Princess: Risk of Death
with 80% Credible Interval**

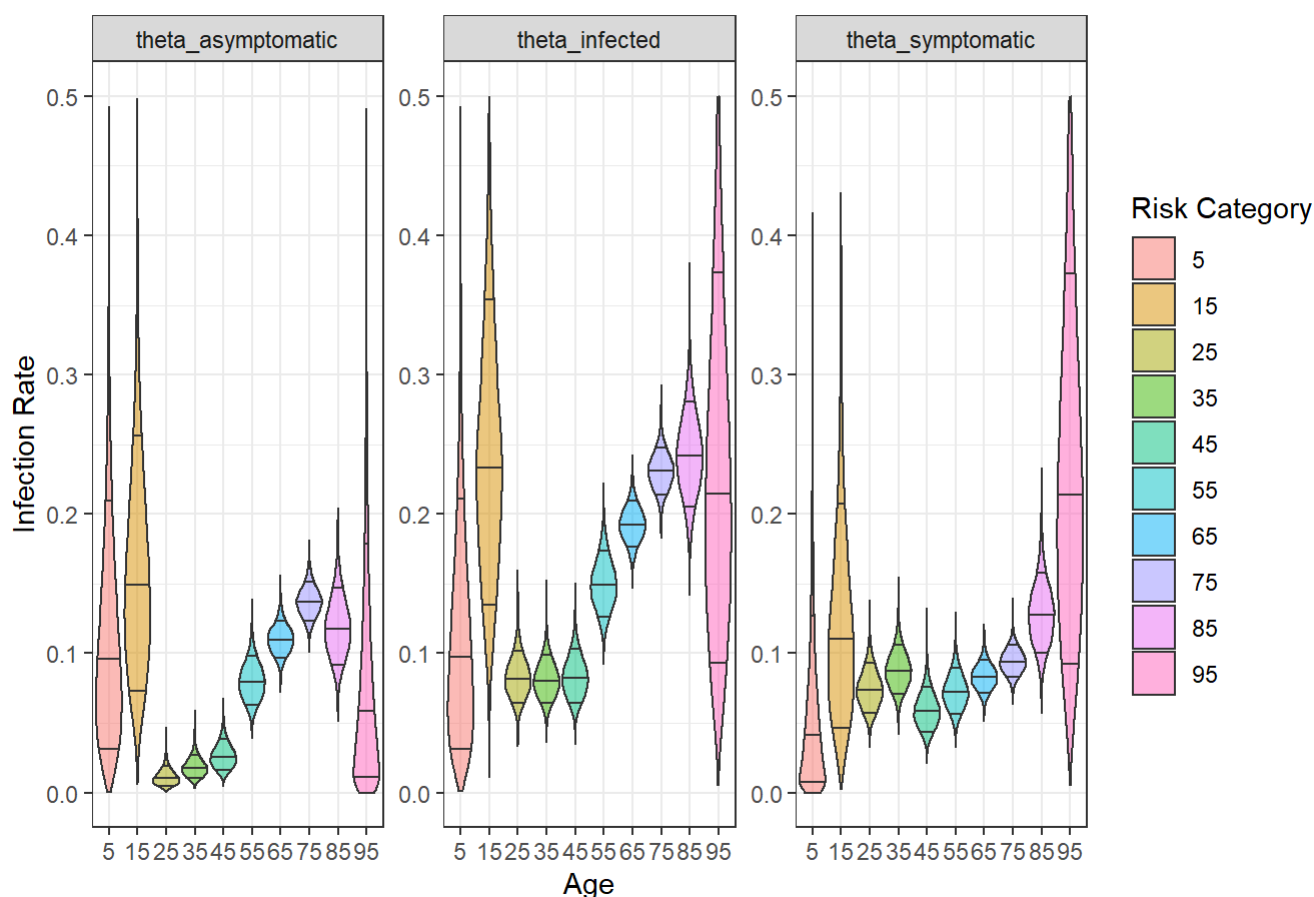


```
# ggplot(theta_post_vec %>% filter(risk_cat != "theta_deaths")) +
#   geom_density(aes(value, fill=(age_cat)), alpha=0.2) +
#   labs(title="Diamond Princess: Risk of Infection", color="") +
#   facet_wrap(~risk_cat, scales = "free")+
#   theme_bw()

vector_risk_rates = ggplot(theta_post_vec %>% filter(risk_cat != "theta_deaths")) +
  geom_violin(aes(age_cat, value, fill = age_cat), alpha=0.5, draw_quantiles = c(0.1, 0.5, 0.9),
  scale = "width") +
  labs(title="Diamond Princess: Risk of Infection", color="", y="Infection Rate", x="Age", fill=
  "Risk Category") +
  facet_wrap(~risk_cat, scales = "free")+
  ylim(0,.5)+
  theme_bw()
vector_risk_rates
```

```
## Warning: Removed 896 rows containing non-finite values (stat_ydensity).
```

Diamond Princess: Risk of Infection

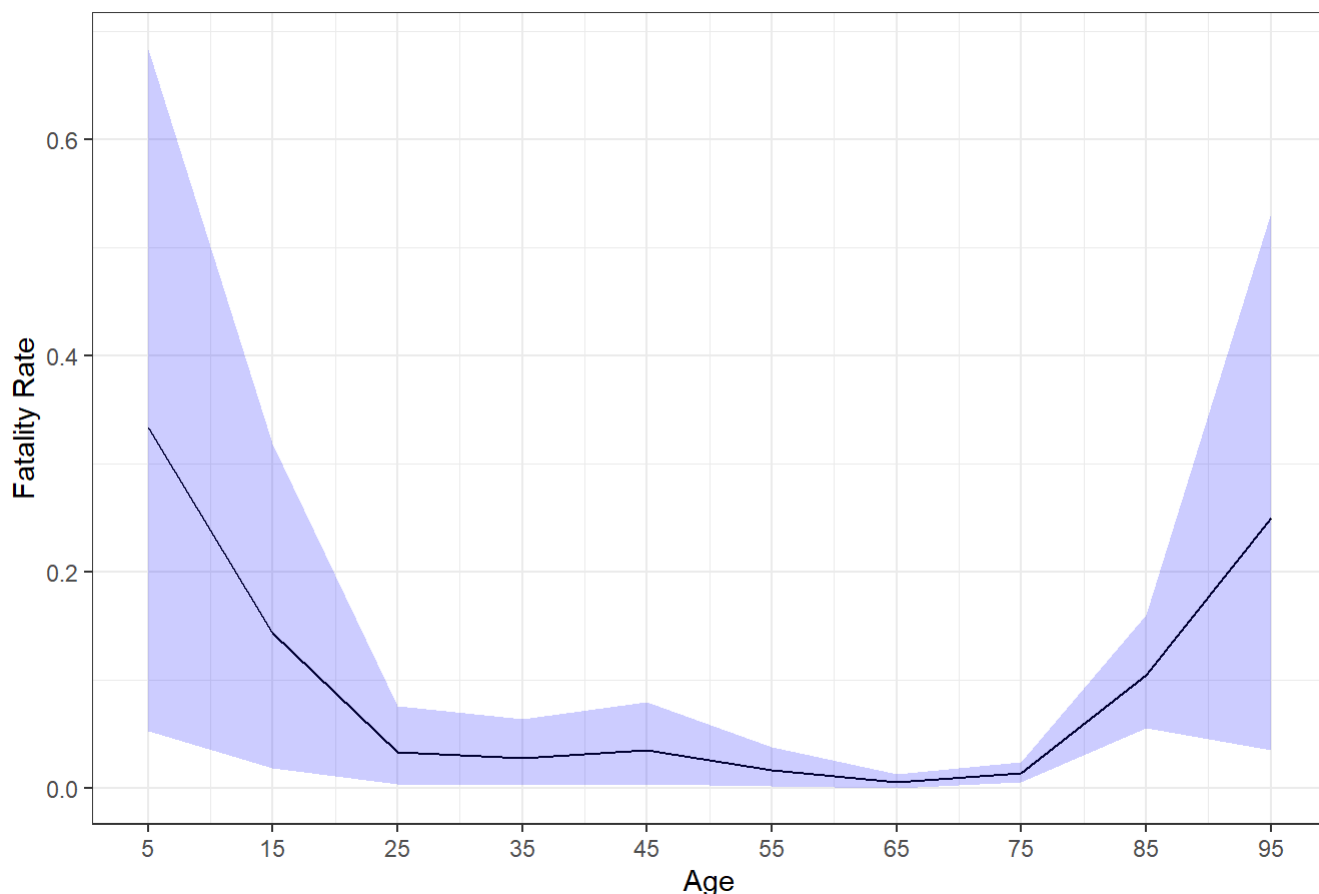


```
# ggplot(theta_post_vec %>% filter(risk_cat == "theta_deaths")) +
#   geom_violin(aes(age_cat, value, fill = age_cat), alpha=0.5, draw_quantiles = c(0.1, 0.5, 0.9), scale = "width") +
#   labs(title="Diamond Princess: Risk of Death from Infection", color="", y="Death Rate", x="Age", fill="Risk Category") + facet_wrap(~risk_cat, scales = "free")+
#   ylim(0,.5)+
#   theme_bw()

risk_band = theta_post_vec %>% filter(risk_cat == "theta_deaths") %>%
  group_by(age_cat) %>%
  summarize(mortality_mean = mean(value),
            mortality_low = quantile(value, probs = .1),
            mortality_high = quantile(value, probs = .9)) %>%
  mutate(age= as.numeric(as.character(age_cat)))

vector_mortality_rates = ggplot(risk_band ) +
  geom_line(aes(age, mortality_mean)) +
  geom_ribbon(aes(age, ymin=mortality_low, ymax=mortality_high), fill= "blue", alpha=.2)+
  theme_bw() +
  labs(title="Independent Model - Fatality Rate: 80% Credible Interval", y= "Fatality Rate", x=
"Age")+
  scale_x_continuous(breaks = diamond$age)
vector_mortality_rates
```

Independent Model - Fatality Rate: 80% Credible Interval



risk_band

| age_cat <fctr> | mortality_mean <dbl> | mortality_low <dbl> | mortality_high <dbl> | age <dbl> |
|--------------------------|--------------------------------|-------------------------------|--------------------------------|---------------------|
| 5 | 0.332922691 | 0.0532374321 | 0.68325596 | 5 |
| 15 | 0.143964930 | 0.0184090493 | 0.31841167 | 15 |
| 25 | 0.033298224 | 0.0036765068 | 0.07583711 | 25 |
| 35 | 0.027890969 | 0.0030307976 | 0.06412223 | 35 |
| 45 | 0.034947283 | 0.0037698420 | 0.07962472 | 45 |
| 55 | 0.016488417 | 0.0017121177 | 0.03805937 | 55 |
| 65 | 0.005566022 | 0.0005880944 | 0.01278123 | 65 |
| 75 | 0.014172794 | 0.0055513583 | 0.02441283 | 75 |
| 85 | 0.104809846 | 0.0553292500 | 0.15978646 | 85 |
| 95 | 0.250342981 | 0.0350151910 | 0.53230700 | 95 |
| 1-10 of 10 rows | | | | |

Notes: There are very limited data for the age categories 5, 15 and 95. Here the uniform plays an outsize role, the resultant posterior values for these age buckets suggests a uniform prior might not be the most optimal prior, and a prior skewed towards lower risk rates might be more appropriate.

Gaussian Process Model - Fixed Rho

```

# Identify the data input for Greta using the as_data function

categories = nrow(diamond)
exposure = as_data(diamond$exposure)
ages = as_data(diamond$age)

infected = as_data(diamond$confirmed)
symptomatic= as_data(diamond$symptomatic)
asymptomatic= as_data(diamond$asymptomatic)
deaths = as_data(diamond$deaths)

# Define the prior for the theta parameter
#theta_infected = beta(shape1 = 1, shape2 = 1, dim = categories)
#theta_symptomatic = beta(shape1 = 1, shape2 = 1, dim = categories)
#theta_asymptomatic = beta(shape1 = 1, shape2 = 1, dim = categories)

eta_deaths = lognormal(0, 20)

rho_deaths = 10

# kernel & GP
kernel = rbf(rho_deaths, eta_deaths)
f = gp(ages, kernel)
theta_deaths = ilogit(f) # convert to unit interval

# Define the Likelihood for the model
#distribution(infected) = binomial(size = exposure, prob = theta_infected, dim = categories)
#distribution(symptomatic) = binomial(size = exposure, prob = theta_symptomatic, dim = categories)
#distribution(asymptomatic) = binomial(size = exposure, prob = theta_asymptomatic, dim = categories)
distribution(deaths) = binomial(size = infected, prob = theta_deaths, dim = categories)

```

We can now establish and compile the model:

```

# Establish the model
#m_gp= model(theta_deaths, eta_deaths, rho_deaths)
m_gp= model(theta_deaths, eta_deaths)

n_samples = n_samples_base; chains=chains_base; warmup=warmup_base*4

S = n_samples * chains # Total number of simulations
draws_gp=mcmc(m_gp, n_samples = n_samples, warmup = warmup,chains = chains)

```

```

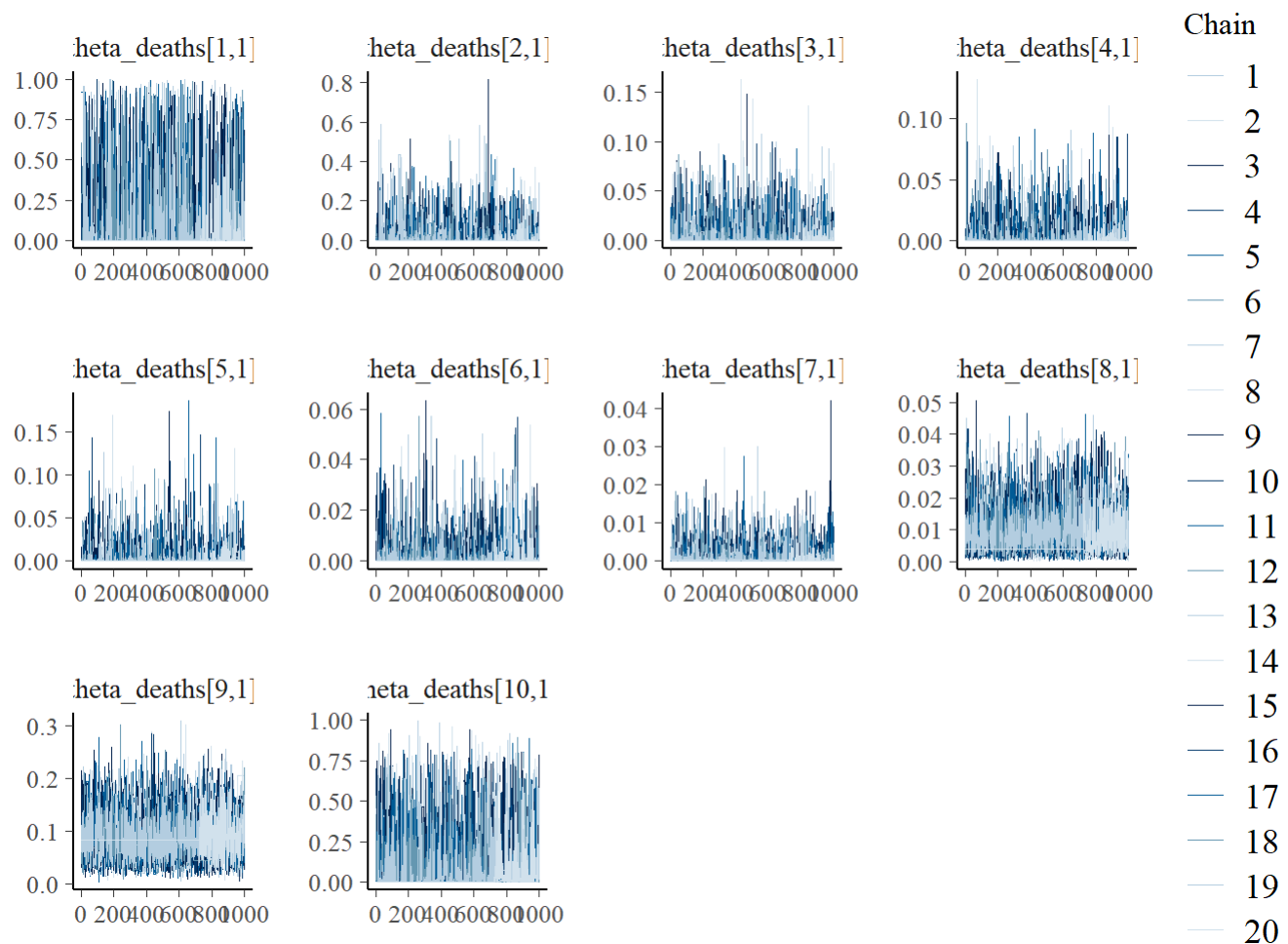
##
## running 20 chains simultaneously on up to 12 cores

```

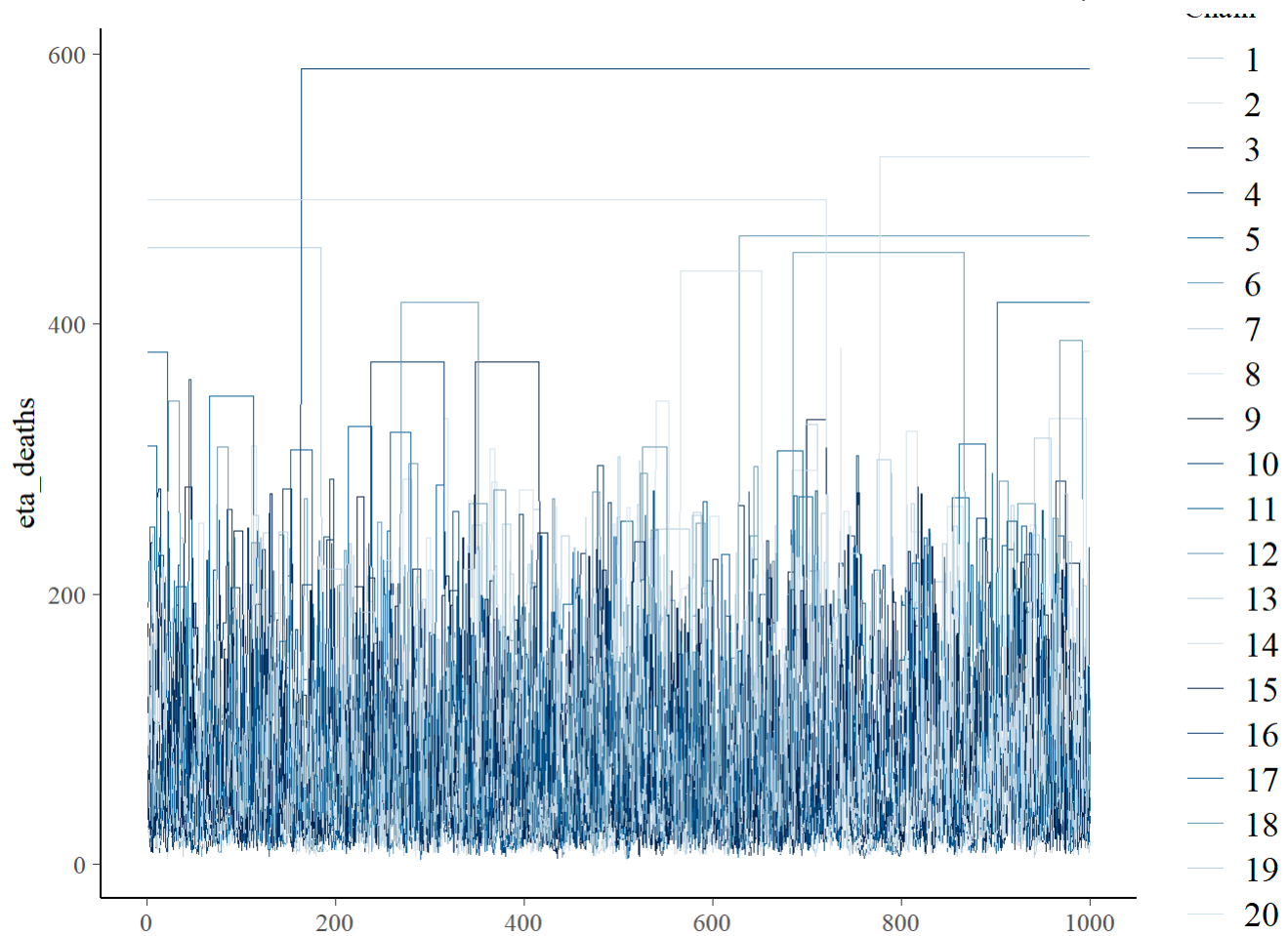
```
##
warmup                                0/1000 | eta:  ?s
warmup ==                             50/1000 | eta:  1m
warmup ====                           100/1000 | eta: 49s
warmup =====                        150/1000 | eta: 45s
warmup =====                       200/1000 | eta: 42s
warmup =====                      250/1000 | eta: 40s
warmup =====                     300/1000 | eta: 38s
warmup =====                    350/1000 | eta: 35s
warmup =====                   400/1000 | eta: 32s
warmup =====                  450/1000 | eta: 29s
warmup =====                 500/1000 | eta: 27s
warmup =====                550/1000 | eta: 24s
warmup =====               600/1000 | eta: 21s
warmup =====              650/1000 | eta: 18s
warmup =====             700/1000 | eta: 16s
warmup =====            750/1000 | eta: 13s
warmup =====           800/1000 | eta: 10s
warmup =====          850/1000 | eta:  8s
warmup =====         900/1000 | eta:  5s
warmup =====        950/1000 | eta:  3s
warmup =====       1000/1000 | eta:  0s
```

```
##
sampling                                0/1000 | eta:  ?s
sampling ==                             50/1000 | eta: 39s
sampling ====                           100/1000 | eta: 40s
sampling =====                        150/1000 | eta: 42s
sampling =====                      200/1000 | eta: 37s
sampling =====                    250/1000 | eta: 36s
sampling =====                   300/1000 | eta: 34s
sampling =====                  350/1000 | eta: 33s
sampling =====                 400/1000 | eta: 30s
sampling =====                450/1000 | eta: 27s
sampling =====               500/1000 | eta: 24s
sampling =====              550/1000 | eta: 22s
sampling =====             600/1000 | eta: 20s
sampling =====            650/1000 | eta: 17s
sampling =====           700/1000 | eta: 15s
sampling =====          750/1000 | eta: 12s
sampling =====         800/1000 | eta: 10s
sampling =====        850/1000 | eta:  7s
sampling =====       900/1000 | eta:  5s
sampling =====      950/1000 | eta:  2s
sampling =====     1000/1000 | eta:  0s
```

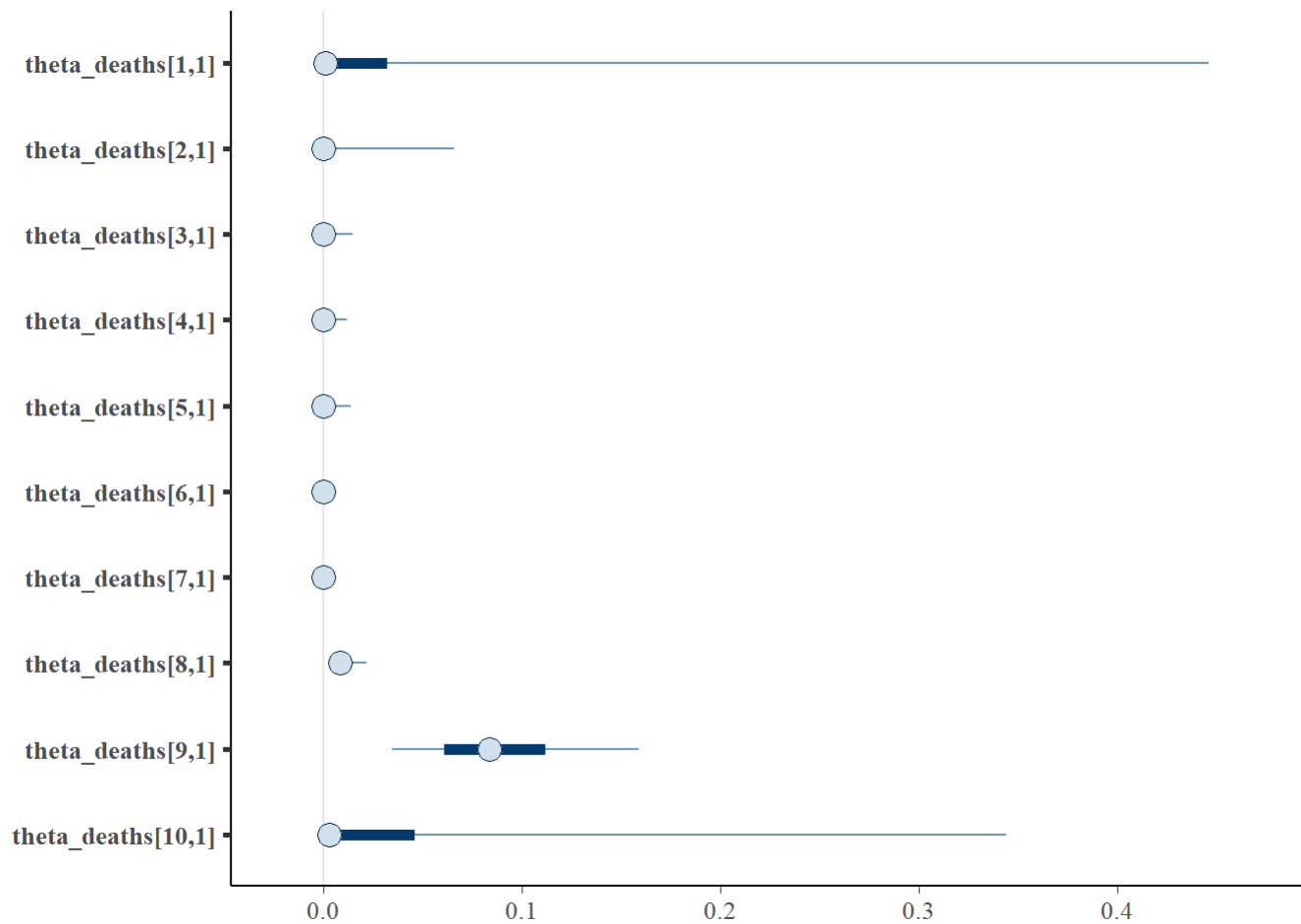
```
# diagnostics
theta_names = colnames(draws_gp[[1]])
mcmc_trace(draws_gp, pars = theta_names[1:10])
```



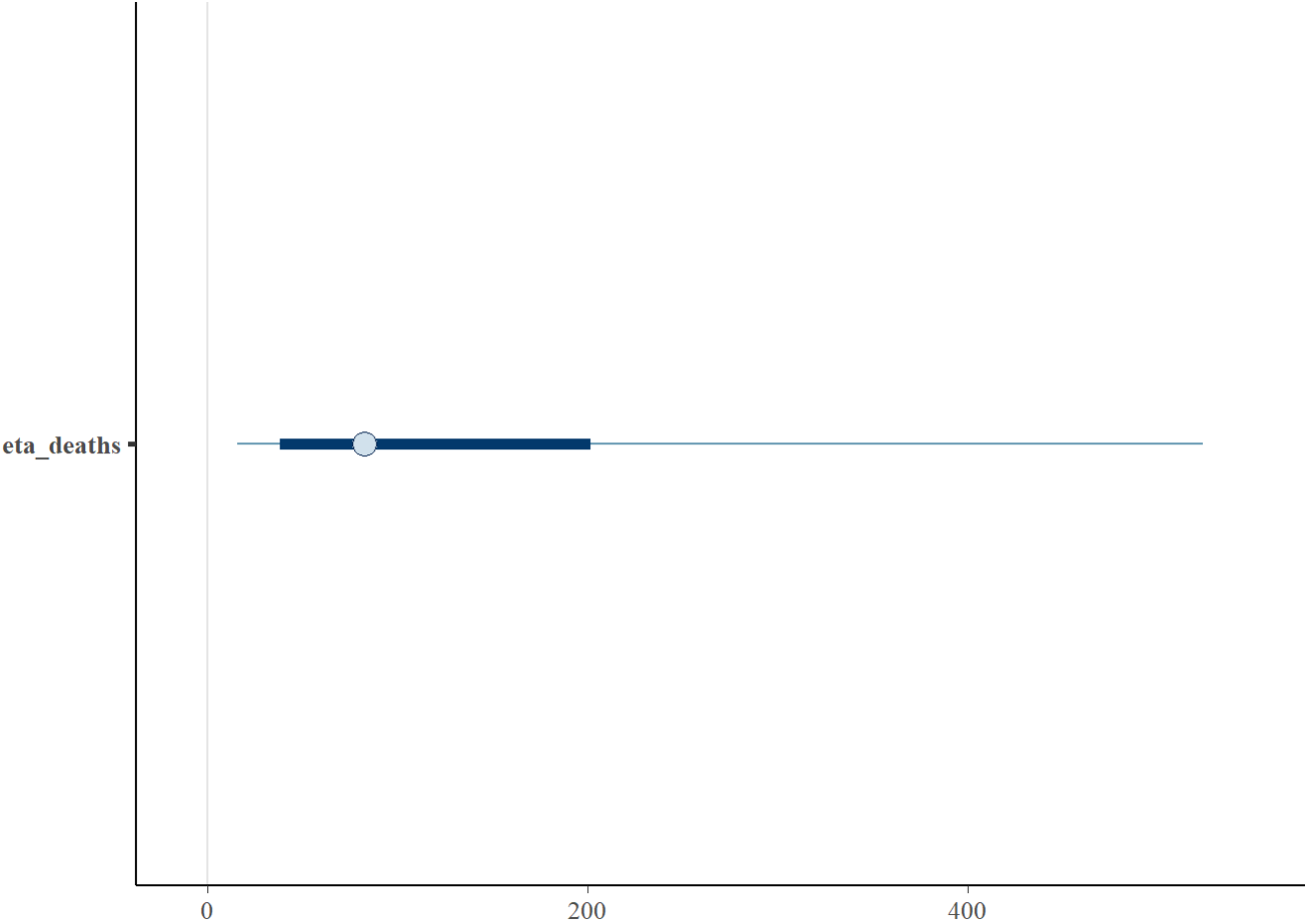
```
mcmc_trace(draws_gp, pars = theta_names[11:11])
```



```
mcmc_intervals(draws_gp, pars = theta_names[1:10])
```

```
mcmc_intervals(draws_gp, theta_names[11:11])
```



```
summary(draws_gp)
```

```
##
## Iterations = 1:1000
## Thinning interval = 1
## Number of chains = 20
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##
```

| | Mean | SD | Naive SE | Time-series SE |
|-----------------------|-----------|-----------|-----------|----------------|
| ## theta_deaths[1,1] | 6.825e-02 | 1.647e-01 | 1.165e-03 | 2.525e-03 |
| ## theta_deaths[2,1] | 1.096e-02 | 3.737e-02 | 2.642e-04 | 5.373e-04 |
| ## theta_deaths[3,1] | 2.545e-03 | 8.119e-03 | 5.741e-05 | 7.928e-05 |
| ## theta_deaths[4,1] | 2.025e-03 | 6.712e-03 | 4.746e-05 | 7.255e-05 |
| ## theta_deaths[5,1] | 2.472e-03 | 8.541e-03 | 6.039e-05 | 8.782e-05 |
| ## theta_deaths[6,1] | 1.081e-03 | 3.710e-03 | 2.624e-05 | 4.979e-05 |
| ## theta_deaths[7,1] | 6.635e-04 | 1.670e-03 | 1.181e-05 | 2.220e-05 |
| ## theta_deaths[8,1] | 9.823e-03 | 6.514e-03 | 4.606e-05 | 1.532e-04 |
| ## theta_deaths[9,1] | 8.776e-02 | 3.849e-02 | 2.722e-04 | 7.796e-04 |
| ## theta_deaths[10,1] | 5.793e-02 | 1.255e-01 | 8.872e-04 | 1.590e-03 |
| ## eta_deaths | 1.533e+02 | 1.636e+02 | 1.157e+00 | 1.159e+01 |

```
##
## 2. Quantiles for each variable:
##
##
```

| | 2.5% | 25% | 50% | 75% | 97.5% |
|-----------------------|-----------|-----------|-----------|-----------|-----------|
| ## theta_deaths[1,1] | 1.627e-15 | 3.945e-06 | 9.194e-04 | 3.226e-02 | 6.451e-01 |
| ## theta_deaths[2,1] | 2.714e-19 | 2.080e-08 | 4.154e-05 | 2.672e-03 | 1.098e-01 |
| ## theta_deaths[3,1] | 6.458e-17 | 2.250e-07 | 1.565e-05 | 9.021e-04 | 2.548e-02 |
| ## theta_deaths[4,1] | 2.082e-15 | 3.678e-08 | 2.452e-05 | 6.948e-04 | 1.938e-02 |
| ## theta_deaths[5,1] | 5.801e-18 | 7.607e-08 | 3.560e-05 | 7.926e-04 | 2.341e-02 |
| ## theta_deaths[6,1] | 1.056e-15 | 4.101e-08 | 2.157e-05 | 3.807e-04 | 1.035e-02 |
| ## theta_deaths[7,1] | 5.655e-12 | 7.635e-07 | 5.225e-05 | 4.881e-04 | 5.339e-03 |
| ## theta_deaths[8,1] | 1.785e-03 | 4.449e-03 | 8.248e-03 | 1.326e-02 | 2.516e-02 |
| ## theta_deaths[9,1] | 2.861e-02 | 6.090e-02 | 8.338e-02 | 1.118e-01 | 1.779e-01 |
| ## theta_deaths[10,1] | 2.638e-12 | 3.142e-05 | 2.744e-03 | 4.577e-02 | 4.733e-01 |
| ## eta_deaths | 1.214e+01 | 3.841e+01 | 8.269e+01 | 2.016e+02 | 5.893e+02 |

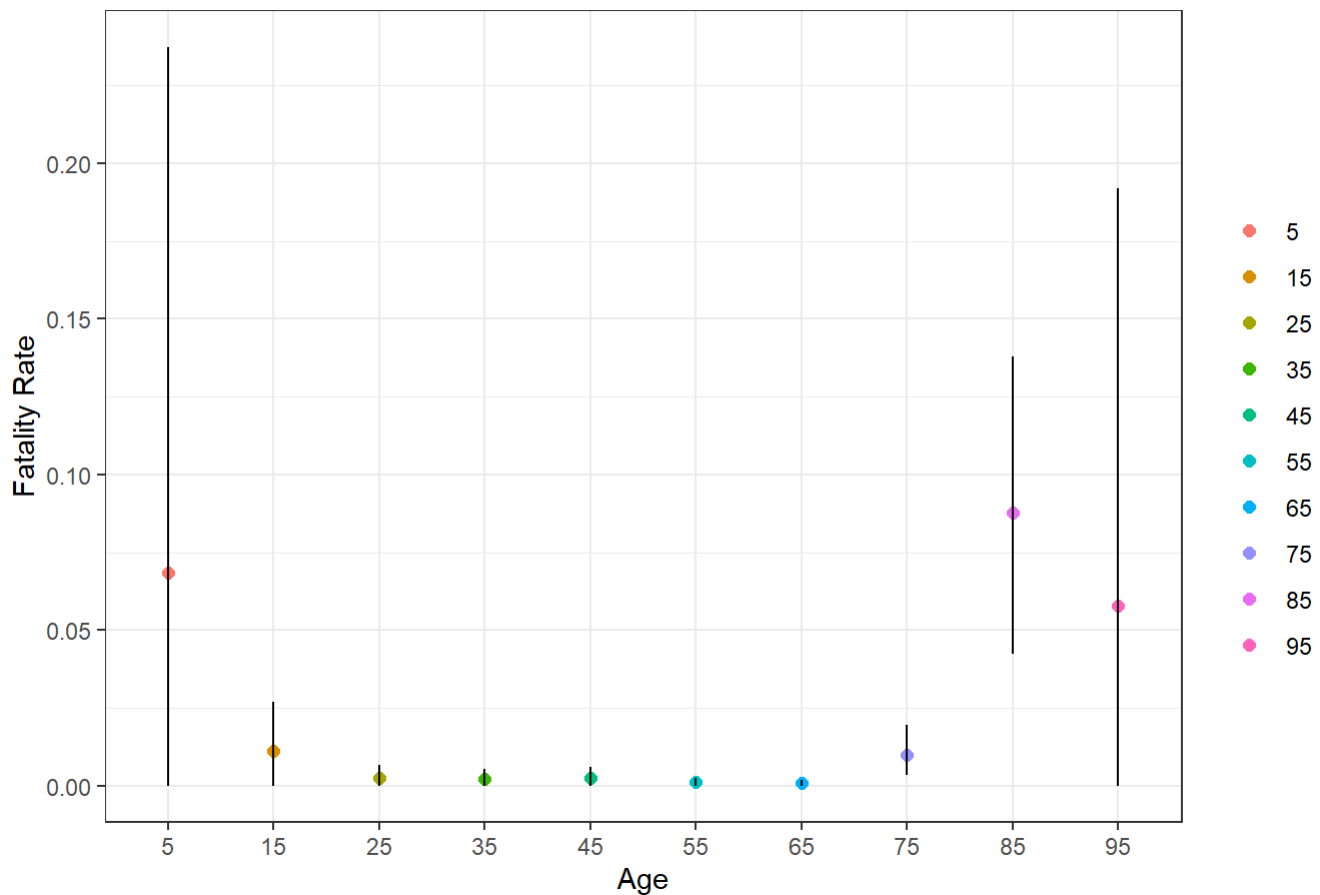
```
theta_post_gp =
  draws_gp %>%
  reduce(rbind) %>%
  as_tibble() %>%
  select(-eta_deaths)

colnames(theta_post_gp) =
  paste0(rep(c( "theta_deaths_"),
              each=categories) , 1:categories)
theta_post_gp = cbind(S=1:S,as_tibble(theta_post_gp)) %>%
  gather(posterior, value, -S) %>%
  mutate(risk_cat = as.factor(gsub(pattern = "_([_]*)$", "", (posterior))),
         age_cat = as.factor((as.numeric(extract_numeric(posterior)))*10-5))
```

```
## extract_numeric() is deprecated: please use readr::parse_number() instead
```

```
theta_post_gp_sum = theta_post_gp %>%  
  group_by(age_cat) %>%  
  summarize(mean = mean(value),  
            low = quantile(value, 0.1),  
            high = quantile(value, 0.9)  
            ) %>%  
  gather(range, value, -mean, -age_cat)  
  
ggplot(theta_post_gp_sum) +  
  geom_point(aes(age_cat, mean, color = age_cat),size=2) +  
  geom_line(aes(age_cat, value, group = age_cat)) +  
  labs(title="Diamond Princess: Risk of Death", color="", y="Fatality Rate", x="Age") +  
  theme_bw()
```

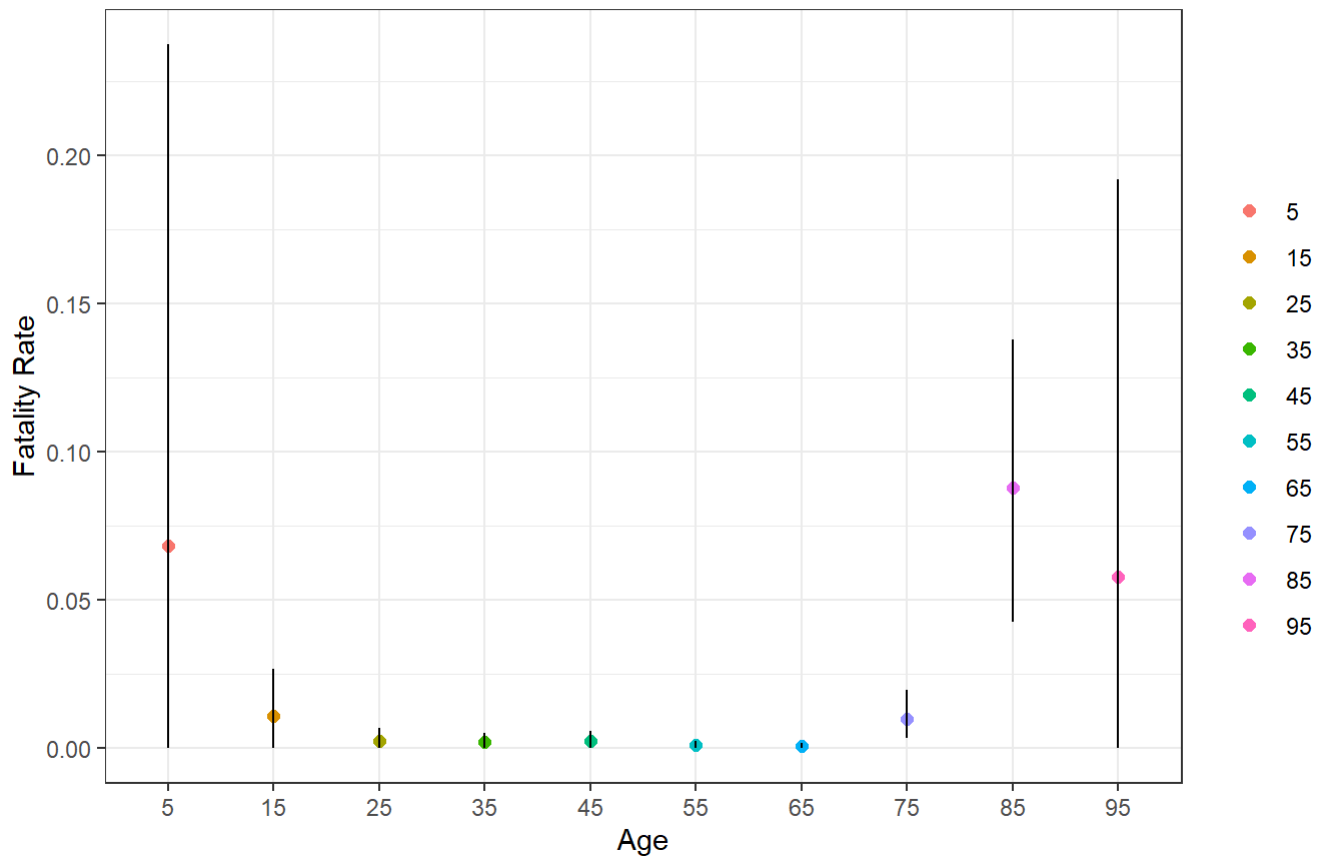
Diamond Princess: Risk of Death



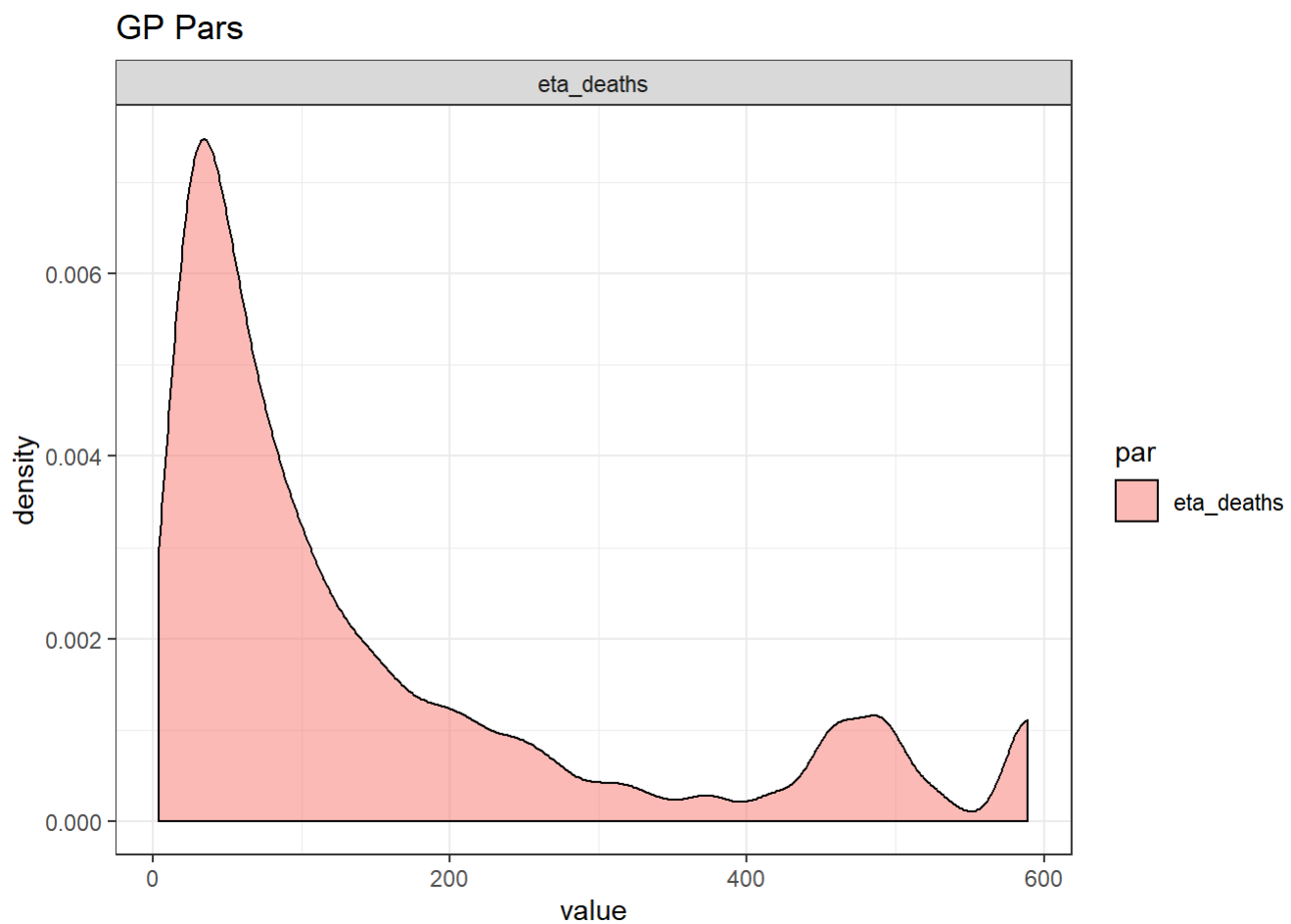
```
# ggplot(theta_post_gp ) +
#   geom_violin(aes(age_cat, value, fill = age_cat), alpha=0.5, draw_quantiles = c(0.1, 0.5, 0.9), scale = "width") +
#   labs(title="Diamond Princess: Risk of Death", color="", y="Fatality Rate", x="Age") +
#   ylim(0,.1)+
#   theme_bw()

ggplot(theta_post_gp_sum) +
  geom_point(aes(age_cat, mean, color = age_cat),size=2) +
  geom_line(aes(age_cat, value, group = age_cat)) +
  labs(title="Diamond Princess: Risk of Death\n with 80% Credible Interval", color="", y="Fatality Rate", x="Age") +
  theme_bw()
```

Diamond Princess: Risk of Death with 80% Credible Interval



```
gp_par =  
  draws_gp %>%  
  reduce(rbind) %>%  
  as_tibble() %>%  
  select(eta_deaths) %>%  
  gather(par, value)  
  
ggplot(gp_par ) +  
  geom_density(aes(value, fill=par), alpha=0.5) +  
  labs(title="GP Pars") +  
  facet_wrap(~par, scales = "free")+  
  theme_bw()
```



Summary Statistics

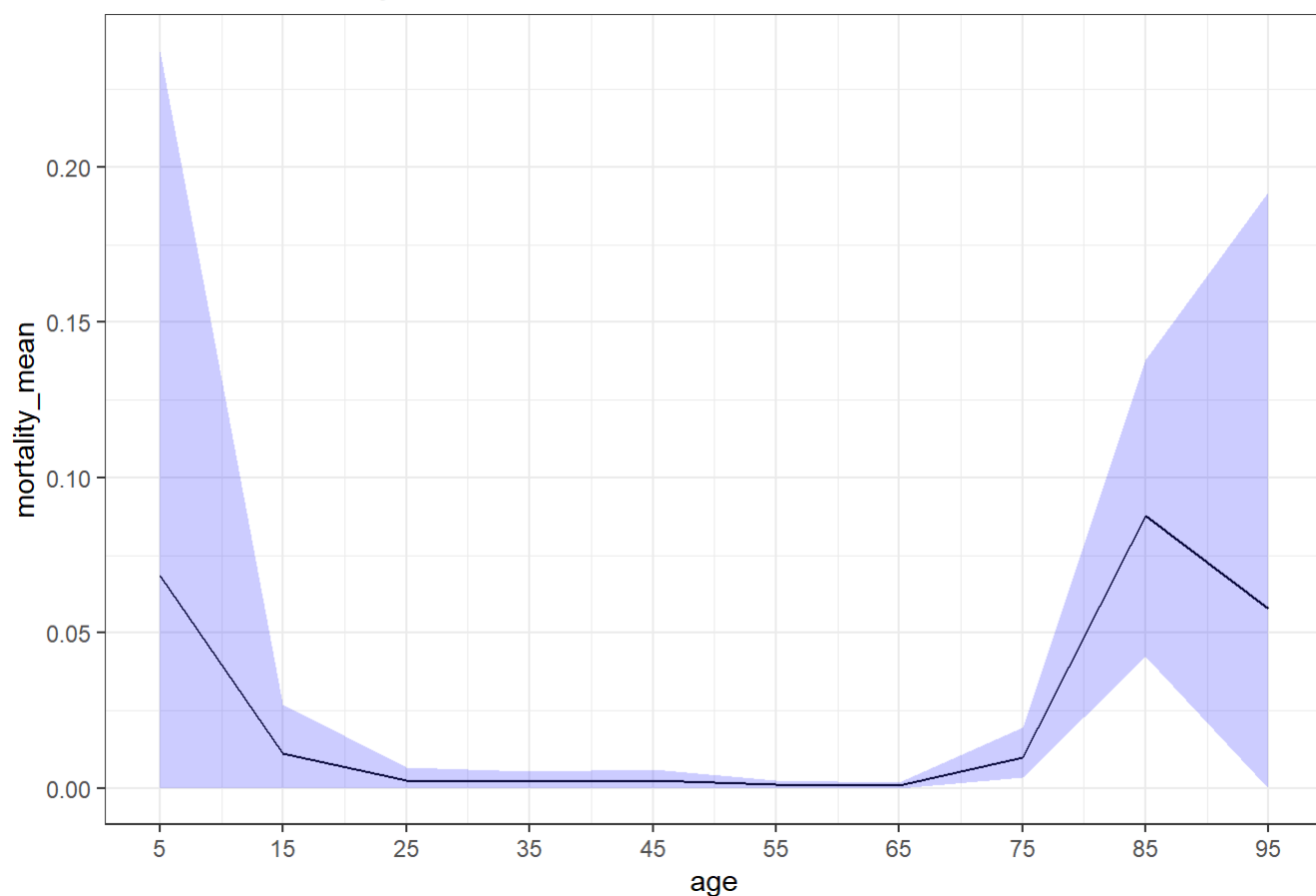
```

risk_band_dependent = theta_post_gp %>%
  group_by(age_cat) %>%
  summarize(mortality_mean = mean(value),
            mortality_low = quantile(value, probs = .1),
            mortality_high = quantile(value, probs = .9))

gp_mortality_fixed_rho = ggplot(risk_band_dependent %>% mutate(age= as.numeric(as.character(age_cat)))) +
  geom_line(aes(age, mortality_mean)) +
  geom_ribbon(aes(age, ymin=mortality_low, ymax=mortality_high), fill= "blue", alpha=.2)+
  theme_bw() +
  labs(title="GP Model - Fatality Rate: 80% Credible Interval")+
  scale_x_continuous(breaks = diamond$age)
gp_mortality_fixed_rho

```

GP Model - Fatality Rate: 80% Credible Interval



risk_band_dependent

| age_cat <fctr> | mortality_mean <dbl> | mortality_low <dbl> | mortality_high <dbl> |
|-------------------|-------------------------|------------------------|-------------------------|
| 5 | 0.0682484932 | 2.035502e-10 | 0.237491090 |
| 15 | 0.0109584918 | 2.951104e-12 | 0.026879345 |
| 25 | 0.0025447035 | 2.774312e-11 | 0.006724129 |

| age_cat <fctr> | mortality_mean <dbl> | mortality_low <dbl> | mortality_high <dbl> |
|--------------------------|--------------------------------|-------------------------------|--------------------------------|
| 35 | 0.0020249264 | 1.029842e-11 | 0.005187103 |
| 45 | 0.0024715985 | 9.915176e-13 | 0.005939558 |
| 55 | 0.0010805332 | 3.362539e-11 | 0.002585711 |
| 65 | 0.0006635048 | 1.393904e-09 | 0.001810135 |
| 75 | 0.0098225998 | 3.356994e-03 | 0.019580672 |
| 85 | 0.0877590146 | 4.249387e-02 | 0.137910546 |
| 95 | 0.0579274139 | 4.040642e-08 | 0.192044955 |
| 1-10 of 10 rows | | | |

Gaussian Process Model - Variable Rho


```

# Identify the data input for Greta using the as_data function

categories = nrow(diamond)
exposure = as_data(diamond$exposure)
ages = as_data(diamond$age)

infected = as_data(diamond$confirmed)
symptomatic= as_data(diamond$symptomatic)
asymptomatic= as_data(diamond$asymptomatic)
deaths = as_data(diamond$deaths)

# Define the prior for the theta parameter
#theta_infected = beta(shape1 = 1, shape2 = 1, dim = categories)
#theta_symptomatic = beta(shape1 = 1, shape2 = 1, dim = categories)
#theta_asymptomatic = beta(shape1 = 1, shape2 = 1, dim = categories)

eta_deaths = lognormal(0, 20)

rho_mean = 20; rho_sd = 5
rho_deaths = lognormal(log(rho_mean/sqrt(1+rho_sd^2/rho_mean^2)), sdlog = sqrt(log(1+rho_sd^2/rho_mean^2)))

# kernel & GP
kernel = rbf(rho_deaths, eta_deaths)
f = gp(ages, kernel)
theta_deaths = ilogit(f) # convert to unit interval

# Define the Likelihood for the model
#distribution(infected) = binomial(size = exposure, prob = theta_infected, dim = categories)
#distribution(symptomatic) = binomial(size = exposure, prob = theta_symptomatic, dim = categories)
#distribution(asymptomatic) = binomial(size = exposure, prob = theta_asymptomatic, dim = categories)
distribution(deaths) = binomial(size = infected, prob = theta_deaths, dim = categories)

```

We can now establish and compile the model:

```

# Establish the model
#m_gp= model(theta_deaths, eta_deaths, rho_deaths)
m_gp_var= model(theta_deaths, eta_deaths, rho_deaths)

n_samples = n_samples_base; chains=chains_base; warmup=warmup_base*4
S = n_samples * chains # Total number of simulations
draws_gp_var=mcmc(m_gp_var, n_samples = n_samples, warmup = warmup,chains = chains)

```

```

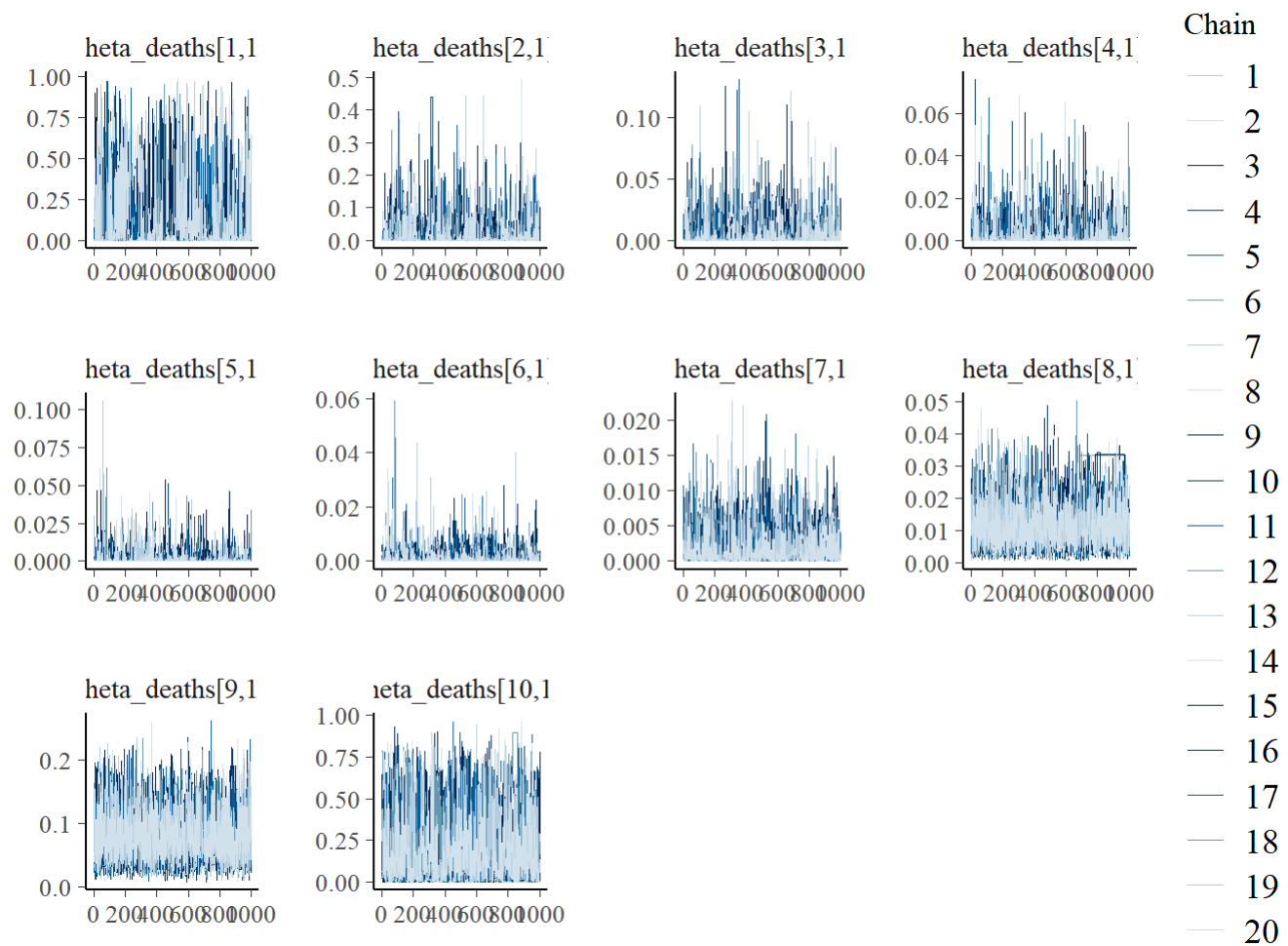
##
## running 20 chains simultaneously on up to 12 cores

```

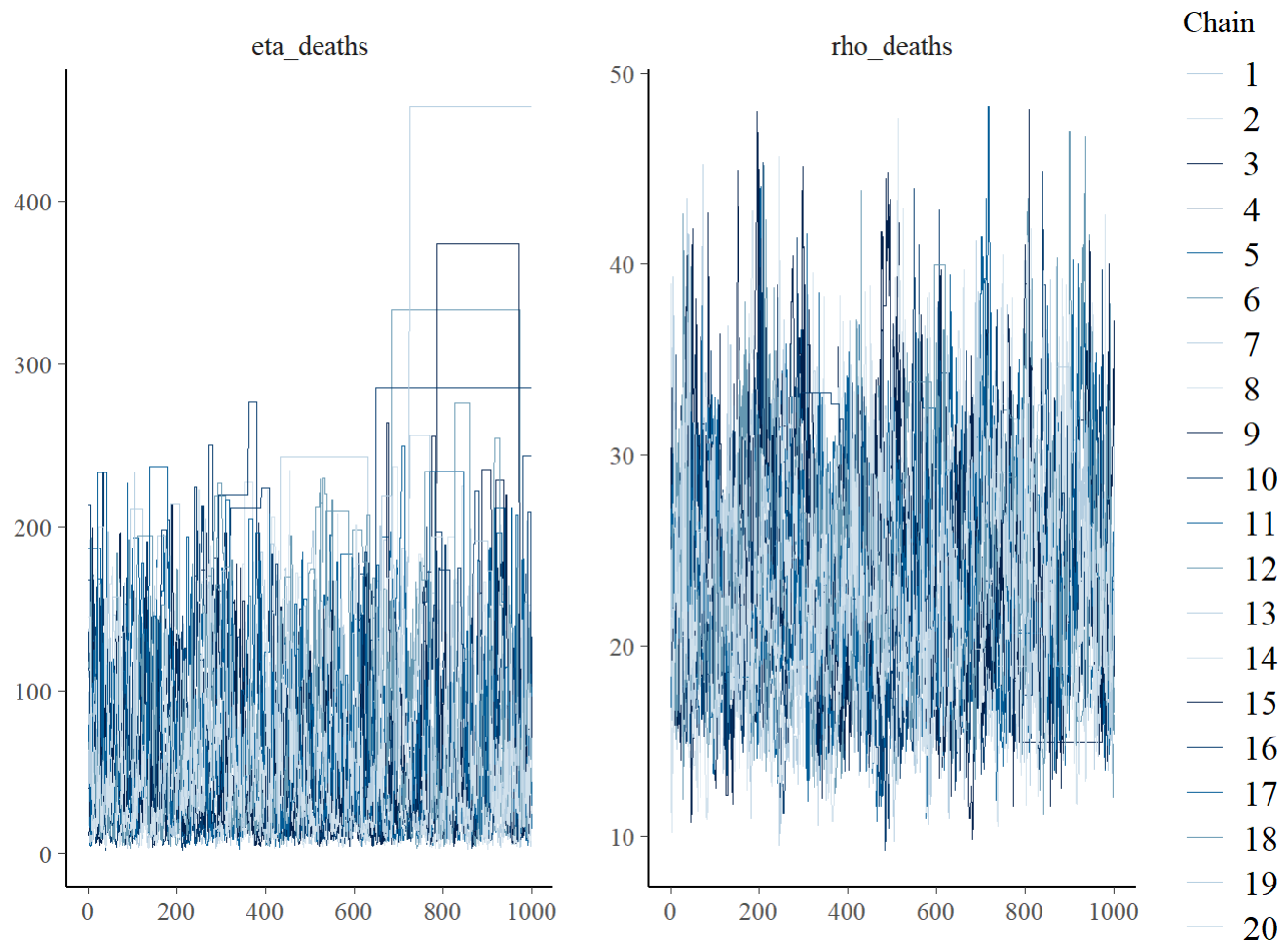
```
##
warmup                                0/1000 | eta:  ?s
warmup ==                             50/1000 | eta:  1m
warmup ====                           100/1000 | eta:  1m
warmup =====                        150/1000 | eta:  1m
warmup =====                       200/1000 | eta:  1m
warmup =====                      250/1000 | eta:  1m
warmup =====                     300/1000 | eta:  1m
warmup =====                    350/1000 | eta: 47s
warmup =====                   400/1000 | eta: 43s
warmup =====                  450/1000 | eta: 40s
warmup =====                 500/1000 | eta: 36s
warmup =====                550/1000 | eta: 32s
warmup =====               600/1000 | eta: 28s
warmup =====              650/1000 | eta: 25s
warmup =====             700/1000 | eta: 21s
warmup =====            750/1000 | eta: 18s
warmup =====           800/1000 | eta: 14s
warmup =====          850/1000 | eta: 11s
warmup =====         900/1000 | eta:  7s
warmup =====        950/1000 | eta:  4s
warmup =====       1000/1000 | eta:  0s
```

```
##
sampling                                0/1000 | eta:  ?s
sampling ==                             50/1000 | eta:  1m
sampling ====                           100/1000 | eta:  1m
sampling =====                        150/1000 | eta:  1m
sampling =====                      200/1000 | eta:  1m
sampling =====                     250/1000 | eta:  1m
sampling =====                    300/1000 | eta: 47s
sampling =====                   350/1000 | eta: 46s
sampling =====                  400/1000 | eta: 41s
sampling =====                 450/1000 | eta: 39s
sampling =====                500/1000 | eta: 35s
sampling =====               550/1000 | eta: 31s
sampling =====              600/1000 | eta: 28s
sampling =====             650/1000 | eta: 25s
sampling =====            700/1000 | eta: 21s
sampling =====           750/1000 | eta: 17s
sampling =====          800/1000 | eta: 14s
sampling =====         850/1000 | eta: 11s
sampling =====        900/1000 | eta:  7s
sampling =====       950/1000 | eta:  4s
sampling =====      1000/1000 | eta:  0s
```

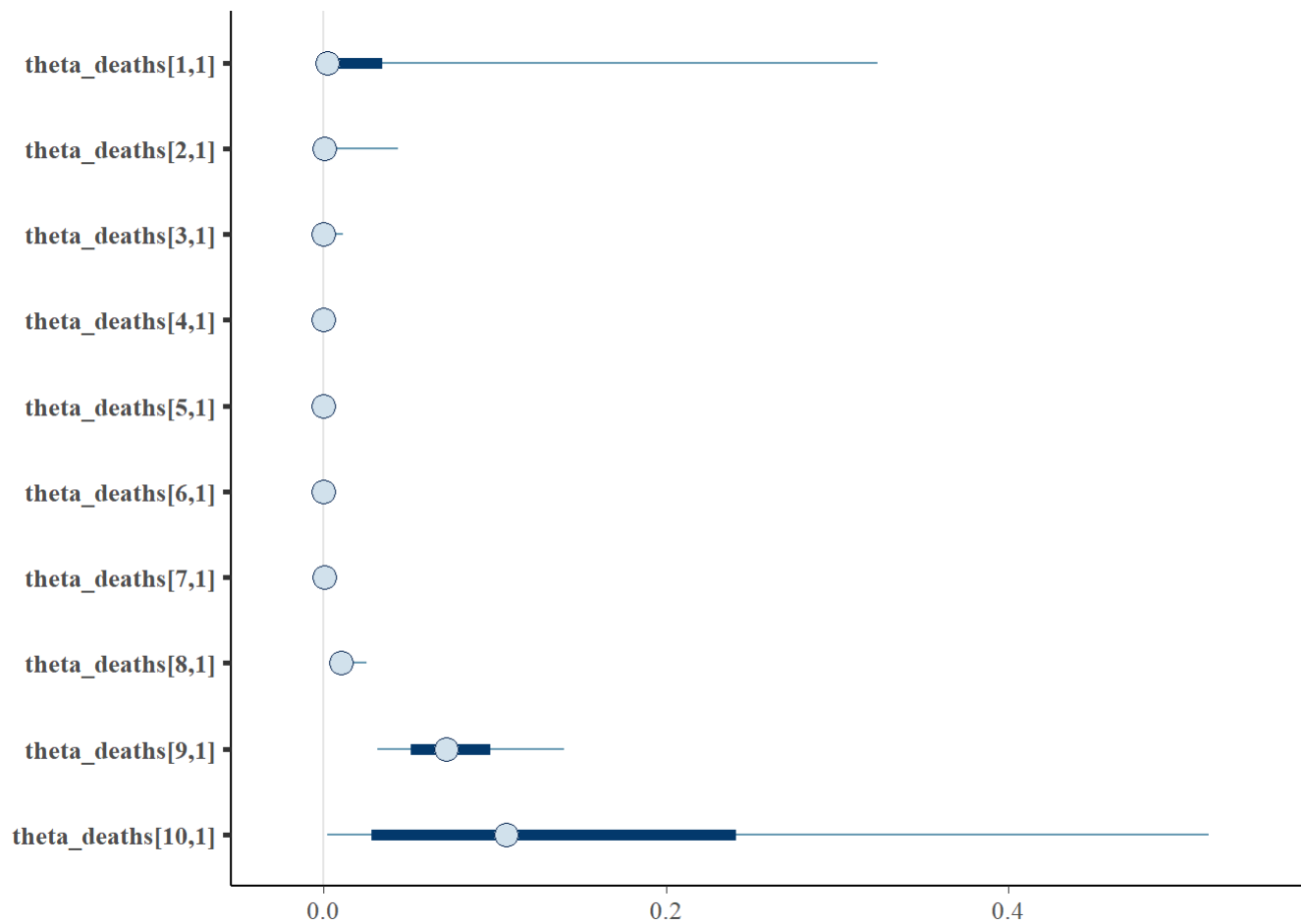
```
# diagnostics
theta_names = colnames(draws_gp_var[[1]])
mcmc_trace(draws_gp_var, pars = theta_names[1:10])
```



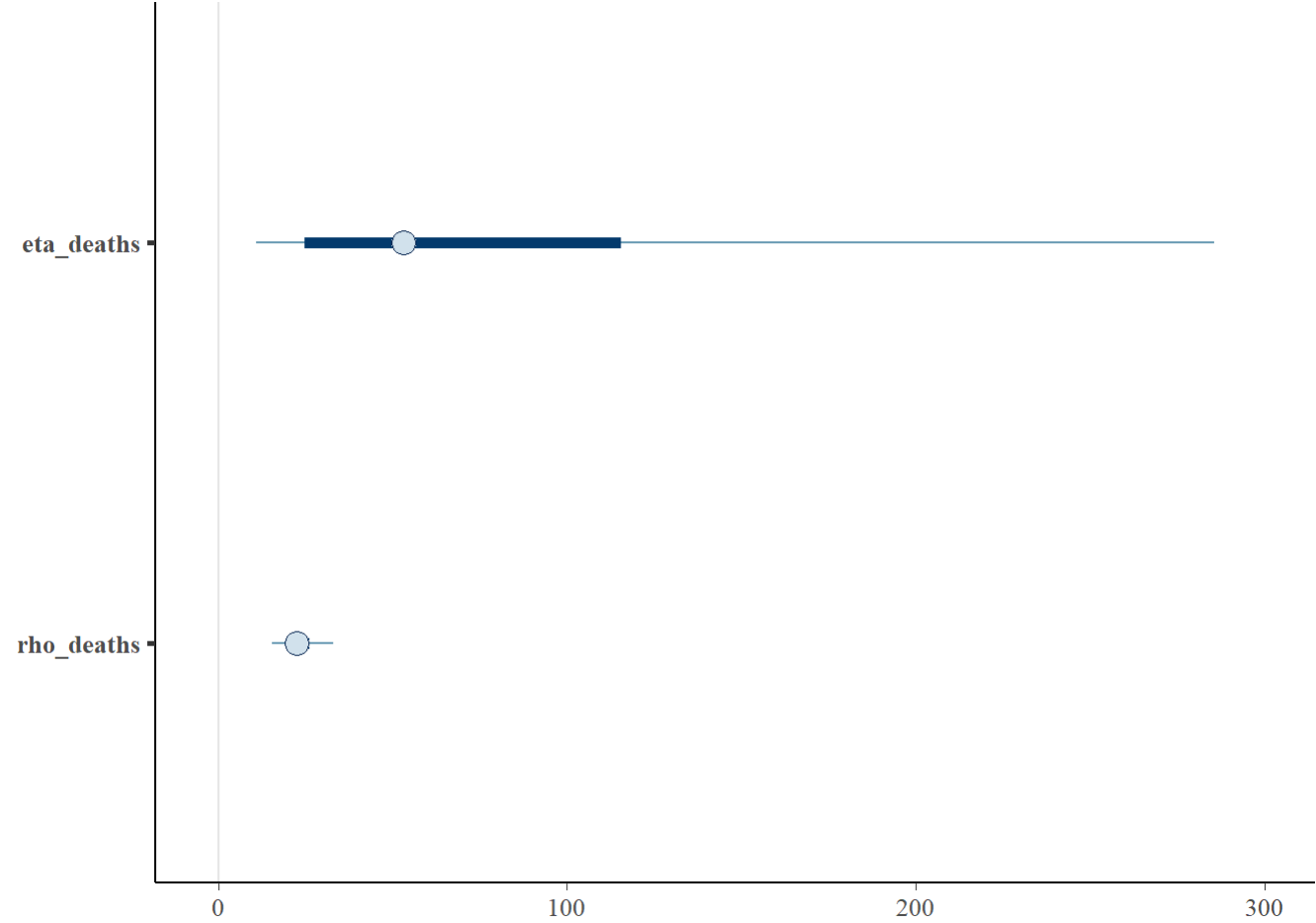
```
mcmc_trace(draws_gp_var, pars = theta_names[11:12])
```



```
mcmc_intervals(draws_gp_var, pars = theta_names[1:10])
```



```
mcmc_intervals(draws_gp_var, theta_names[11:12])
```



```
summary(draws_gp_var)
```

```
##
## Iterations = 1:1000
## Thinning interval = 1
## Number of chains = 20
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##
```

| | Mean | SD | Naive SE | Time-series SE |
|-----------------------|-----------|-----------|-----------|----------------|
| ## theta_deaths[1,1] | 5.470e-02 | 0.130900 | 9.256e-04 | 2.675e-03 |
| ## theta_deaths[2,1] | 8.346e-03 | 0.026294 | 1.859e-04 | 4.078e-04 |
| ## theta_deaths[3,1] | 2.108e-03 | 0.006231 | 4.406e-05 | 7.919e-05 |
| ## theta_deaths[4,1] | 1.159e-03 | 0.003728 | 2.636e-05 | 5.035e-05 |
| ## theta_deaths[5,1] | 7.998e-04 | 0.002996 | 2.119e-05 | 4.396e-05 |
| ## theta_deaths[6,1] | 5.778e-04 | 0.001708 | 1.208e-05 | 2.152e-05 |
| ## theta_deaths[7,1] | 1.201e-03 | 0.001741 | 1.231e-05 | 2.995e-05 |
| ## theta_deaths[8,1] | 1.138e-02 | 0.006919 | 4.892e-05 | 3.289e-04 |
| ## theta_deaths[9,1] | 7.674e-02 | 0.034285 | 2.424e-04 | 6.961e-04 |
| ## theta_deaths[10,1] | 1.615e-01 | 0.169033 | 1.195e-03 | 3.748e-03 |
| ## eta_deaths | 8.677e+01 | 90.028199 | 6.366e-01 | 6.281e+00 |
| ## rho_deaths | 2.309e+01 | 5.284245 | 3.737e-02 | 1.603e-01 |

```
##
## 2. Quantiles for each variable:
##
##
```

| | 2.5% | 25% | 50% | 75% | 97.5% |
|-----------------------|-----------|-----------|-----------|-----------|-----------|
| ## theta_deaths[1,1] | 3.493e-14 | 2.508e-05 | 2.357e-03 | 3.439e-02 | 4.926e-01 |
| ## theta_deaths[2,1] | 1.223e-13 | 2.935e-06 | 2.688e-04 | 4.367e-03 | 7.391e-02 |
| ## theta_deaths[3,1] | 3.791e-13 | 8.878e-07 | 7.115e-05 | 1.235e-03 | 1.890e-02 |
| ## theta_deaths[4,1] | 4.358e-12 | 3.938e-07 | 4.082e-05 | 5.418e-04 | 1.076e-02 |
| ## theta_deaths[5,1] | 4.072e-12 | 6.240e-07 | 2.864e-05 | 3.460e-04 | 7.171e-03 |
| ## theta_deaths[6,1] | 4.540e-10 | 3.850e-06 | 6.201e-05 | 4.128e-04 | 4.355e-03 |
| ## theta_deaths[7,1] | 1.103e-06 | 1.422e-04 | 5.120e-04 | 1.601e-03 | 6.034e-03 |
| ## theta_deaths[8,1] | 2.508e-03 | 6.493e-03 | 9.931e-03 | 1.452e-02 | 3.332e-02 |
| ## theta_deaths[9,1] | 2.641e-02 | 5.091e-02 | 7.157e-02 | 9.748e-02 | 1.559e-01 |
| ## theta_deaths[10,1] | 4.288e-04 | 2.766e-02 | 1.065e-01 | 2.410e-01 | 6.082e-01 |
| ## eta_deaths | 8.375e+00 | 2.486e+01 | 5.308e+01 | 1.155e+02 | 3.332e+02 |
| ## rho_deaths | 1.467e+01 | 1.930e+01 | 2.249e+01 | 2.604e+01 | 3.489e+01 |

```

theta_post_gp_var =
  draws_gp_var %>%
  reduce(rbind) %>%
  as_tibble() %>%
  select(-eta_deaths, -rho_deaths)

colnames(theta_post_gp_var) =
  paste0(rep(c( "theta_deaths_"),
              each=categories) , 1:categories)

theta_post_gp_var = as_tibble(theta_post_gp_var) %>%
  mutate(S=1:S) %>%
  gather(posterior, value, -S) %>%
  mutate(risk_cat = as.factor(gsub(pattern = "_([_]*)$","",(posterior))),
         age_cat = as.factor((as.numeric(extract_numeric(posterior))*10-5))

```

extract_numeric() is deprecated: please use readr::parse_number() instead

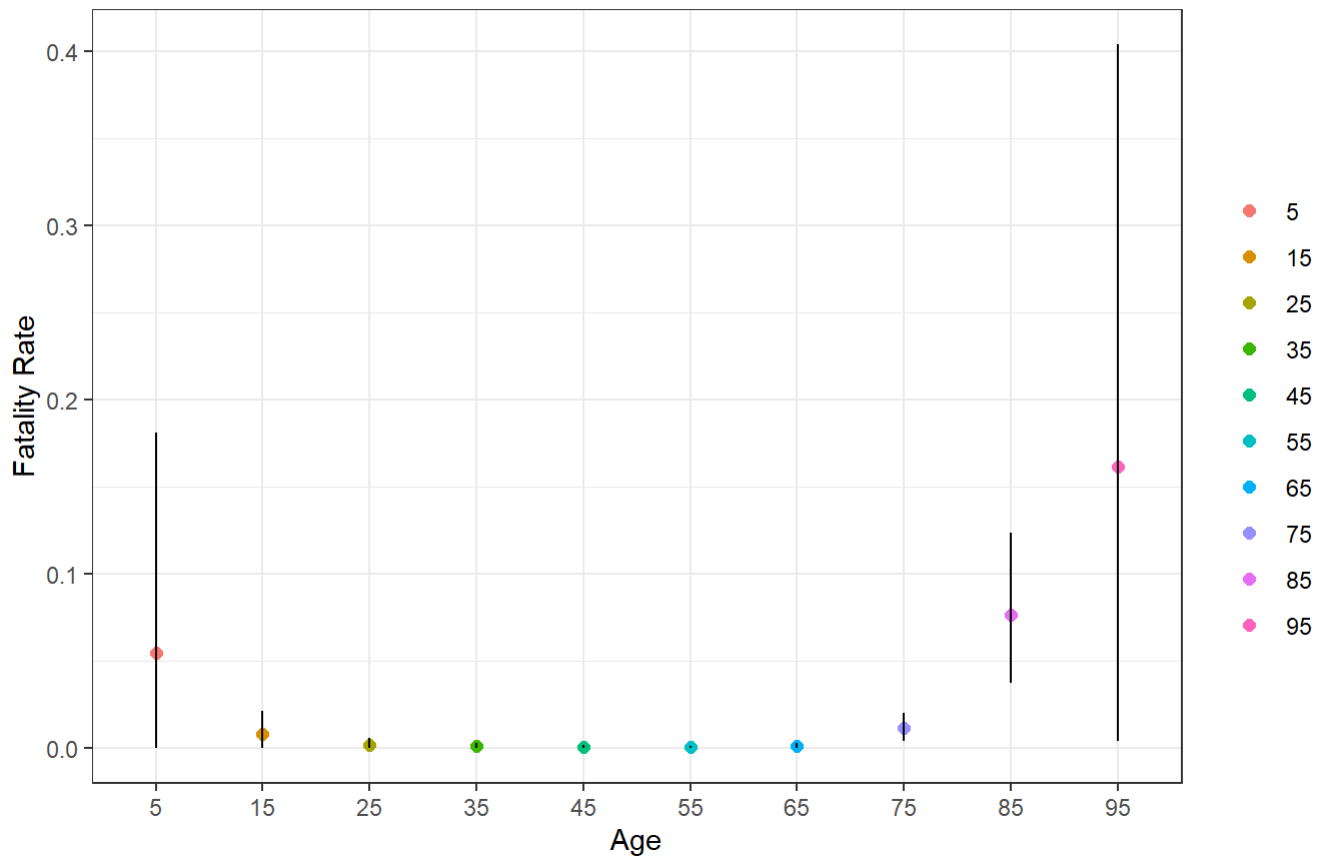
```

theta_post_gp_var_sum = theta_post_gp_var %>%
  group_by(age_cat) %>%
  summarize(mean = mean(value),
            low = quantile(value, 0.1),
            high = quantile(value, 0.9)
            ) %>%
  gather(range, value, -mean, -age_cat)

ggplot(theta_post_gp_var_sum) +
  geom_point(aes(age_cat, mean, color = age_cat),size=2) +
  geom_line(aes(age_cat, value, group = age_cat)) +
  labs(title="Diamond Princess: Risk of Death\n with 80% Credible Interval", color="", y="Fatali
ty Rate", x="Age") +
  theme_bw()

```


Diamond Princess: Risk of Death with 80% Credible Interval

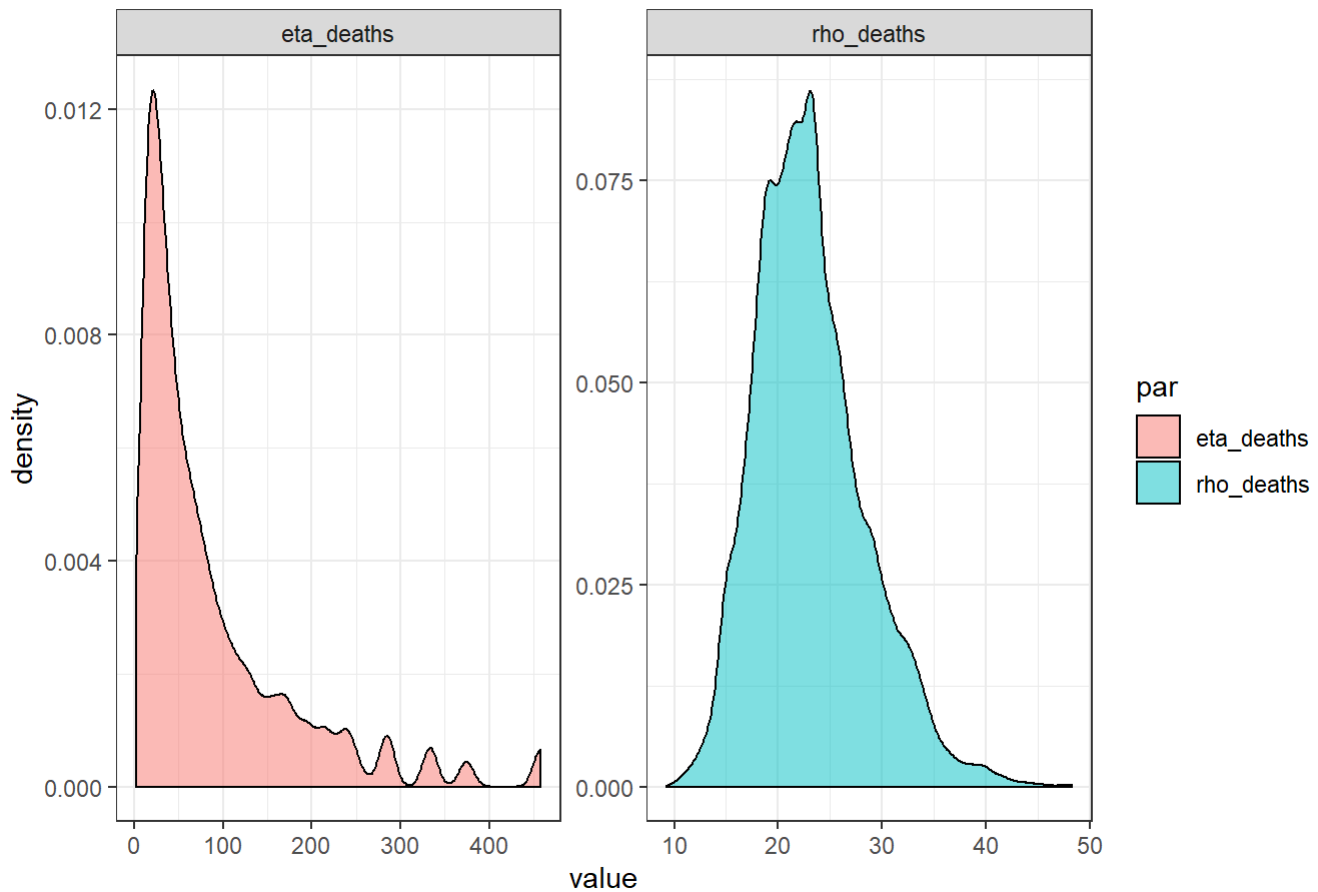


```
# ggplot(theta_post_gp_var) +
#   geom_violin(aes(age_cat, value, fill = age_cat), alpha=0.5, draw_quantiles = c(0.1, 0.5, 0.9), scale = "width") +
#   labs(title="Diamond Princess: Risk of Death", color="", y="Fatality Rate", x="Age") +
#   ylim(0,.2)+
#   theme_bw()

gp_par =
  draws_gp_var %>%
  reduce(rbind) %>%
  as_tibble() %>%
  select(eta_deaths, rho_deaths) %>%
  gather(par, value)

gp_par_plot = ggplot(gp_par) +
  geom_density(aes(value, fill=par), alpha=0.5) +
  labs(title="GP Pars") +
  facet_wrap(~par, scales = "free")+
  theme_bw()
gp_par_plot
```

GP Pars



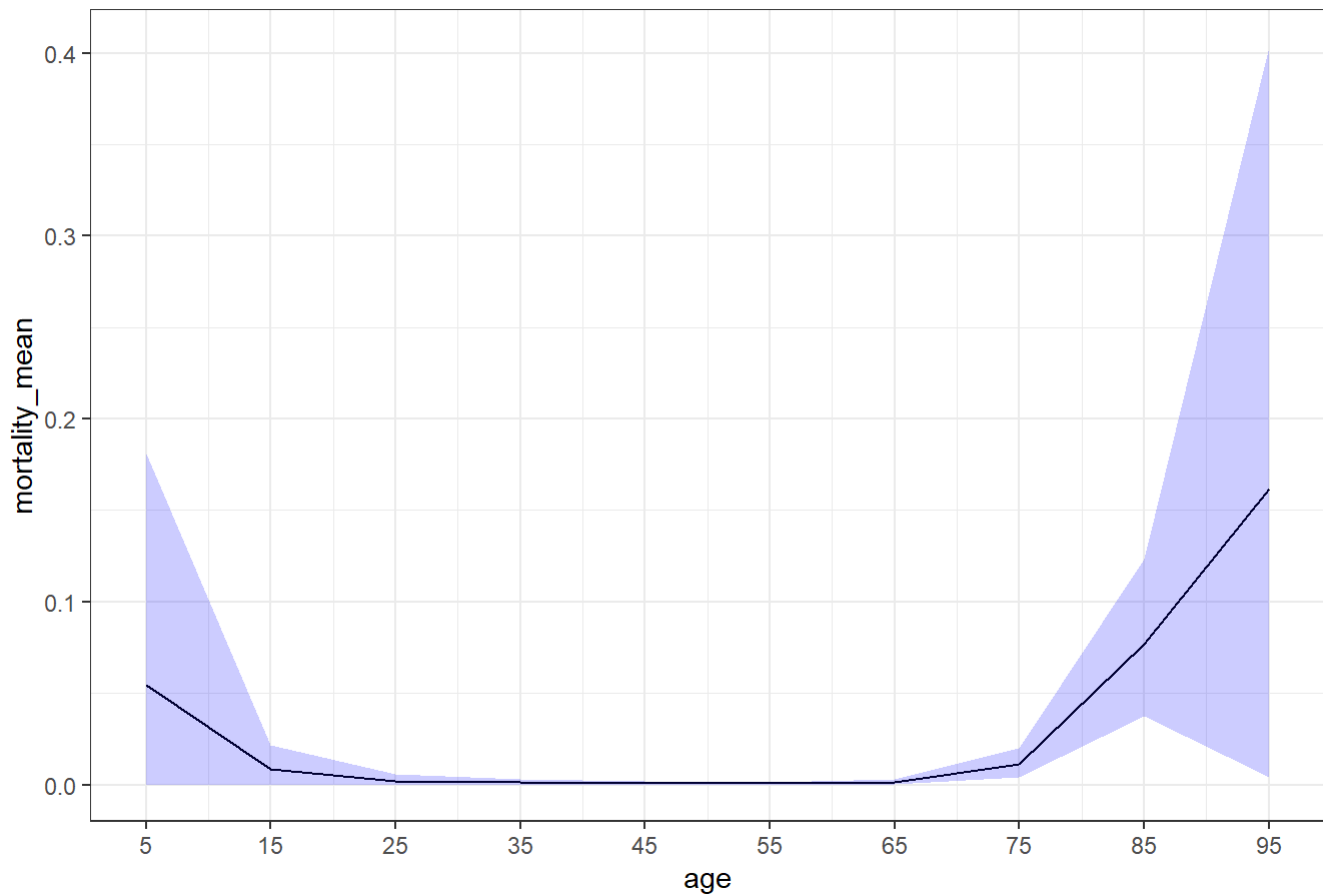
```

risk_band = theta_post_gp_var %>%
  group_by(age_cat) %>%
  summarize(mortality_mean = mean(value),
            mortality_low = quantile(value, probs = .1),
            mortality_high = quantile(value, probs = .9))

gp_mortality = ggplot(risk_band %>% mutate(age= as.numeric(as.character(age_cat)))) +
  geom_line(aes(age, mortality_mean)) +
  geom_ribbon(aes(age, ymin=mortality_low, ymax=mortality_high), fill= "blue", alpha=.2)+
  theme_bw() +
  labs(title="GP Model - Fatality Rate: 80% Credible Interval")+
  scale_x_continuous(breaks = diamond$age)
gp_mortality

```

GP Model - Fatality Rate: 80% Credible Interval



risk_band

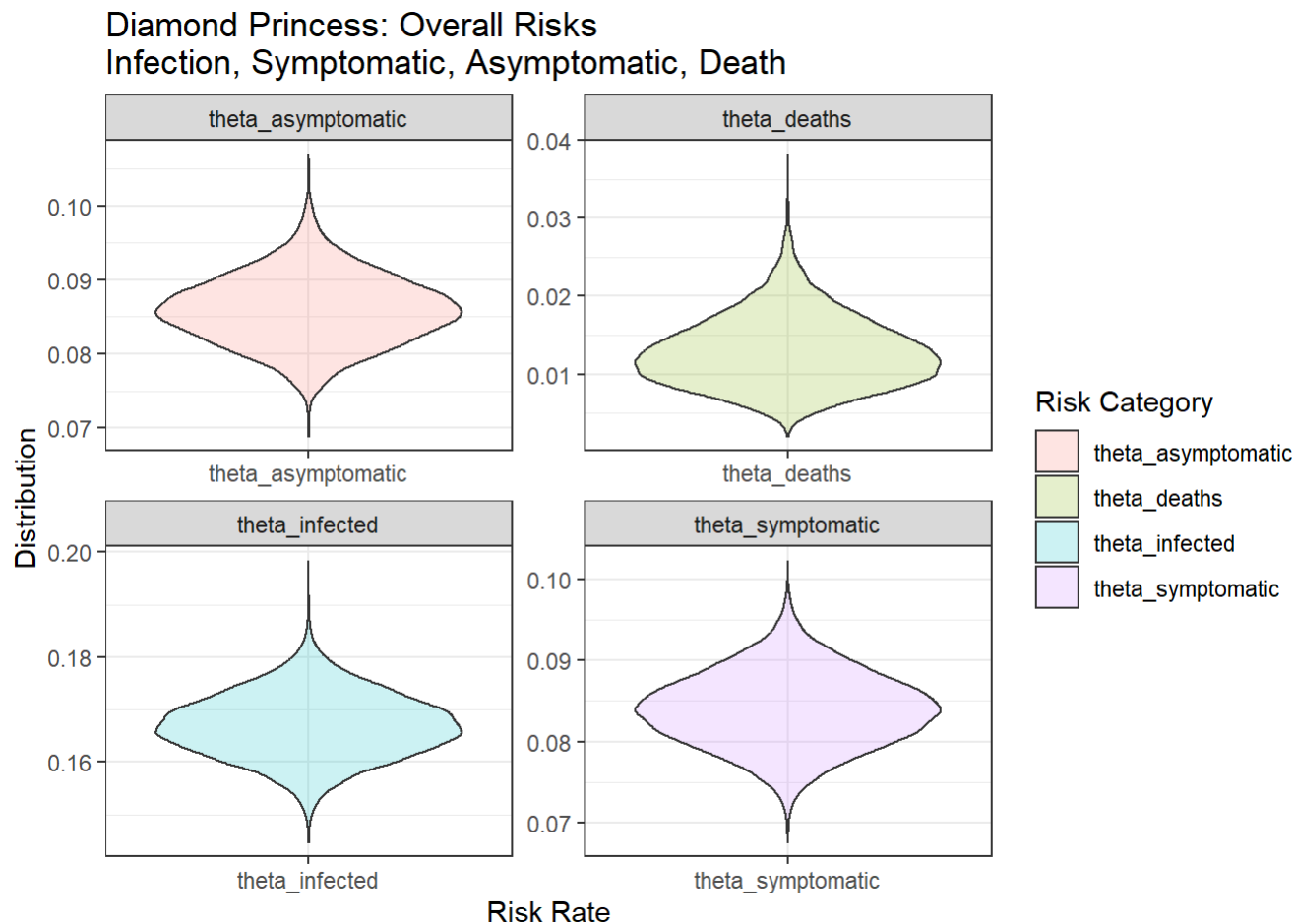
| age_cat <fctr> | mortality_mean <dbl> | mortality_low <dbl> | mortality_high <dbl> |
|-------------------|-------------------------|------------------------|-------------------------|
| 5 | 0.0546982197 | 8.439248e-09 | 0.180739050 |
| 15 | 0.0083460557 | 1.376237e-08 | 0.021632144 |
| 25 | 0.0021077779 | 2.393764e-09 | 0.005748091 |
| 35 | 0.0011586973 | 6.564968e-10 | 0.002884149 |
| 45 | 0.0007997615 | 2.731522e-09 | 0.001720694 |
| 55 | 0.0005778407 | 7.258730e-08 | 0.001504043 |
| 65 | 0.0012010006 | 2.173927e-05 | 0.003182883 |
| 75 | 0.0113772272 | 4.049245e-03 | 0.020159243 |
| 85 | 0.0767387421 | 3.733834e-02 | 0.123468348 |
| 95 | 0.1615174052 | 4.239444e-03 | 0.403819468 |

1-10 of 10 rows

Summary of Results

A. Combined Age Category Model

Risk Rates

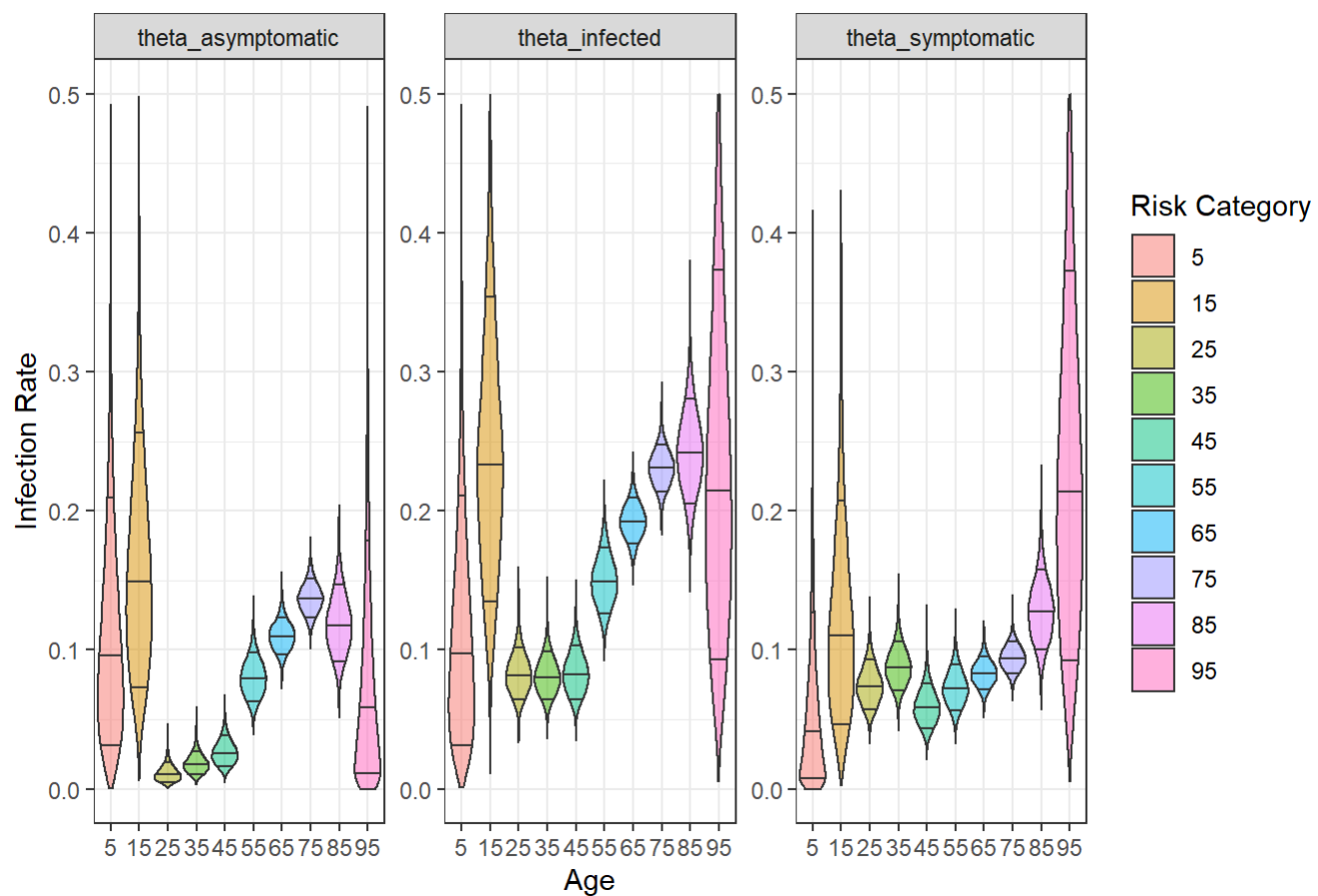


B. Independent Age Category Model

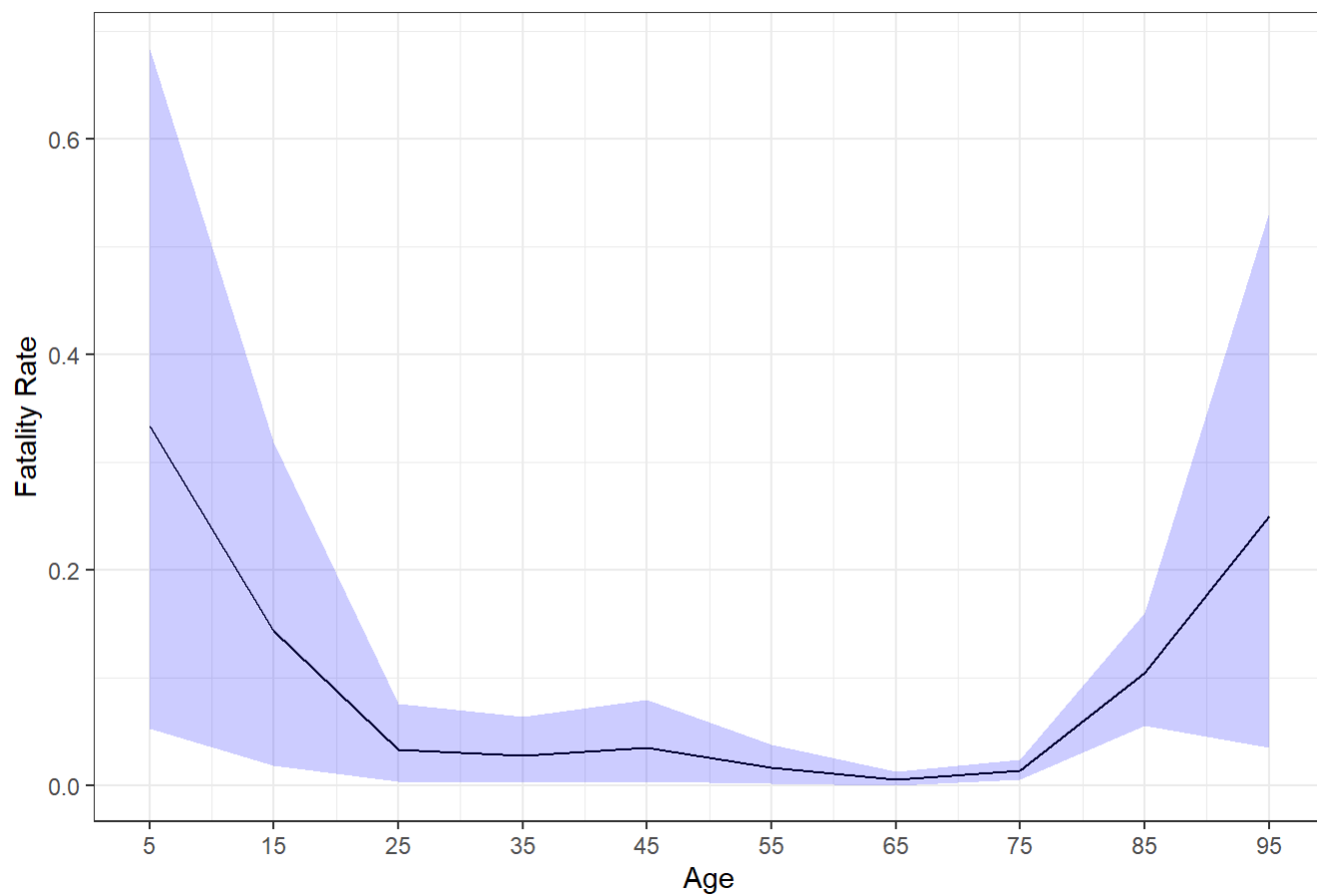
Risk Rates

```
## Warning: Removed 896 rows containing non-finite values (stat_ydensity).
```

Diamond Princess: Risk of Infection



Independent Model - Fatality Rate: 80% Credible Interval



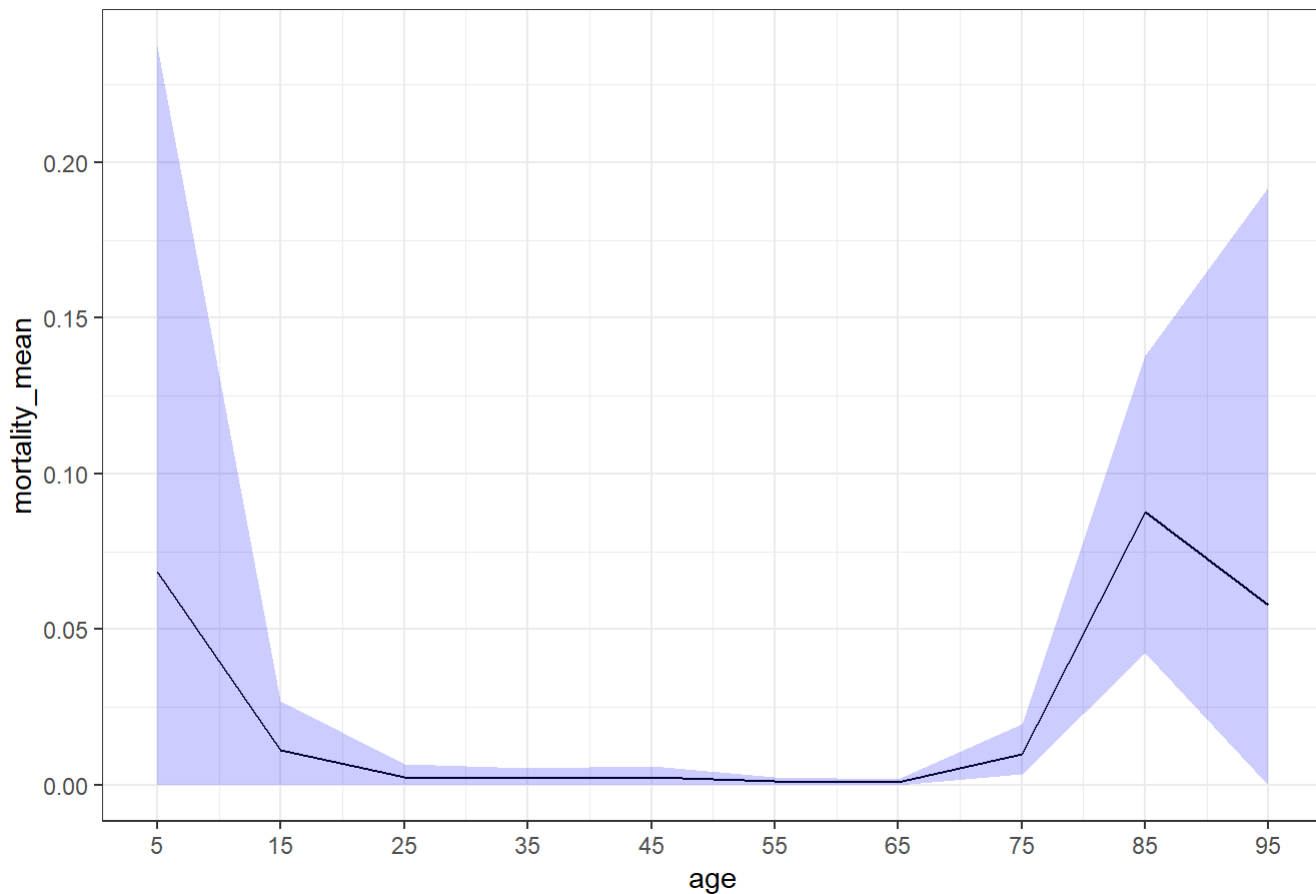
Notes: There are very limited data for the age categories 5, 15 and 95. Here the uniform plays an outsize role, the resultant posterior values for these age buckets suggests a uniform prior might not be the most optimal prior, and a prior skewed towards lower risk rates might be more appropriate.

C. Gaussian Process Model: Fixed Rho

Fixed $\rho = 10$

Fatality Rates

GP Model - Fatality Rate: 80% Credible Interval

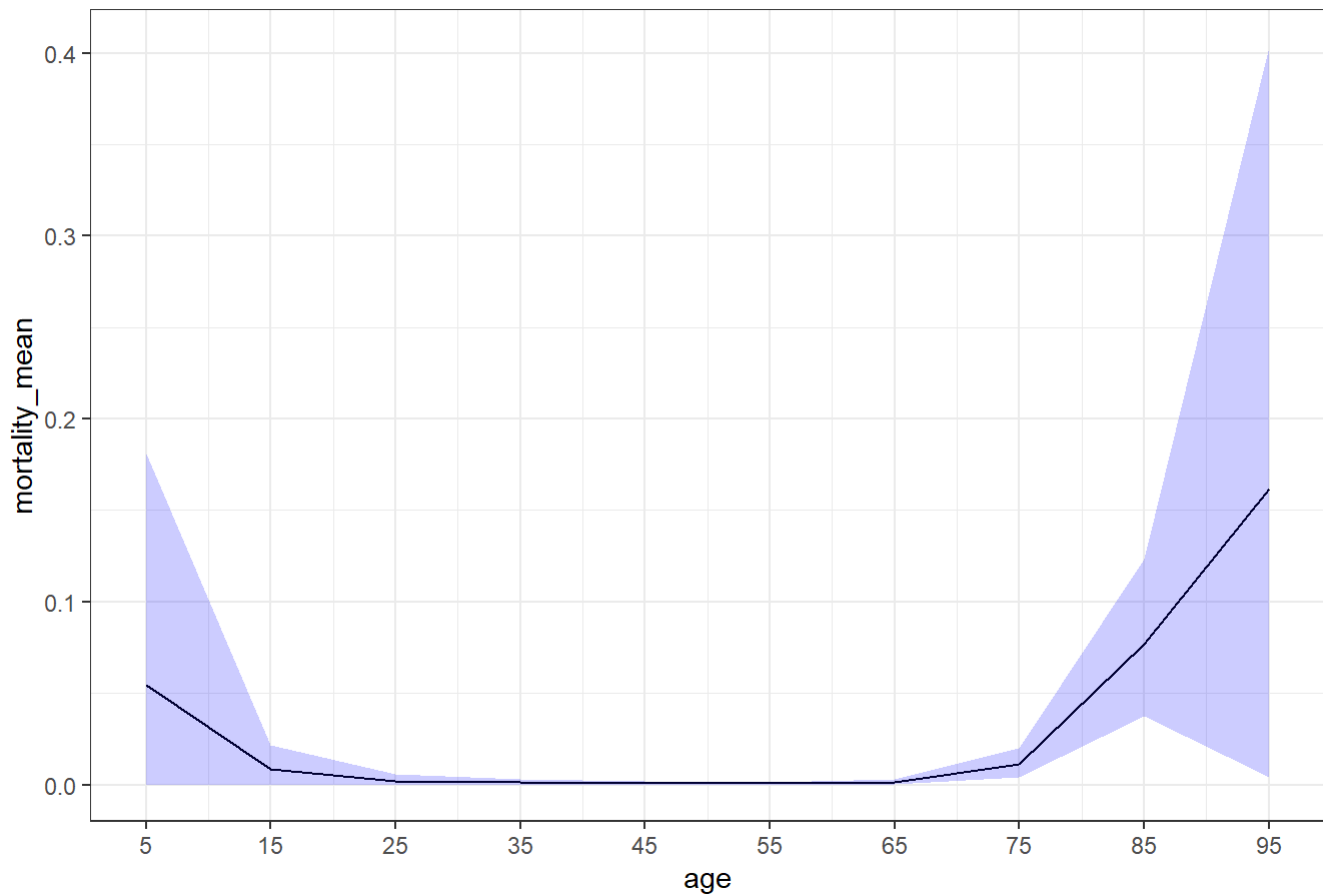


C. Gaussian Process Model: Variable Rho

Fixed $\rho = 10$

Fatality Rates

GP Model - Fatality Rate: 80% Credible Interval



Model Parameters

GP Pars

