County-level analysis of COVID fatality rates in Midwestern states

Walker Moskop

Read in and prepare the data

```
data_dir = 'data/'
df.raw = read.csv(paste0(data_dir, 'county-all-data.csv'))

### should be zero missing vals
sum(is.na(df.raw))
```

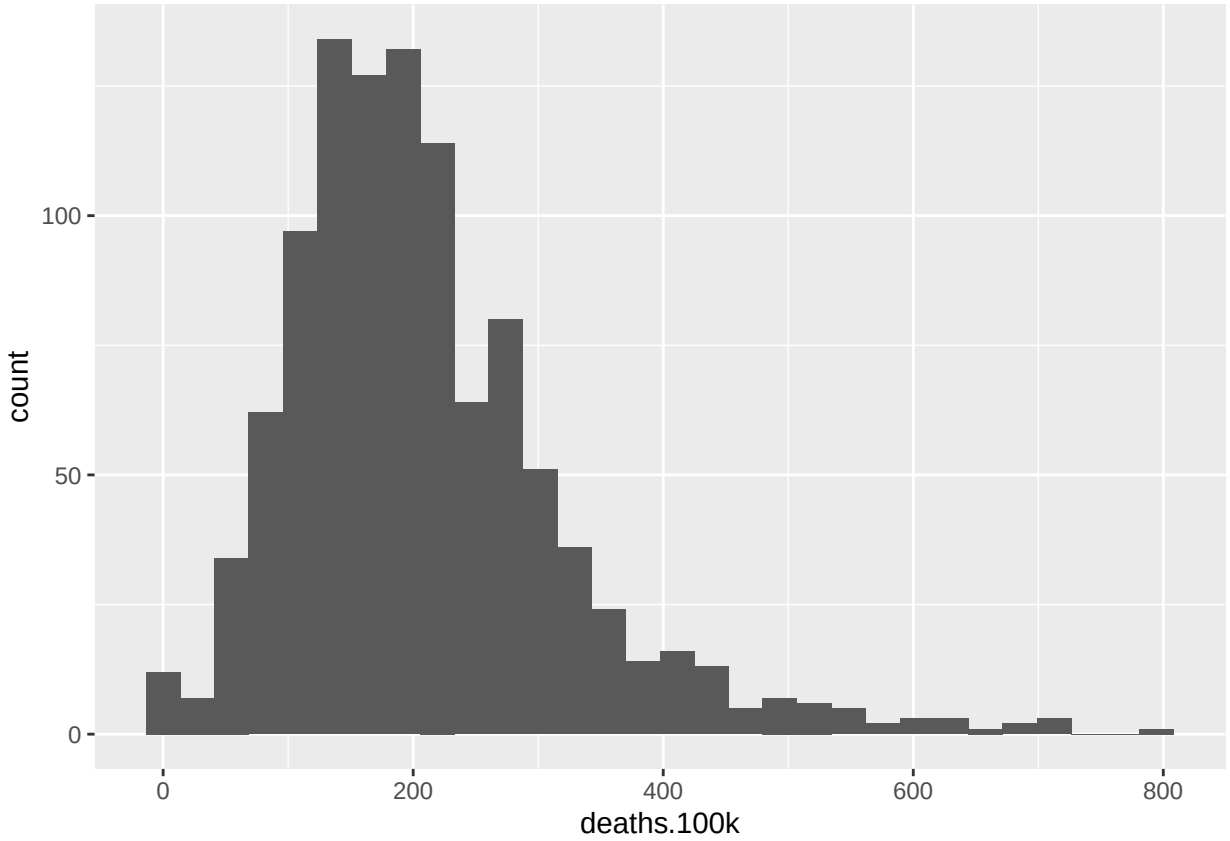
```
## [1] 0
```

```
### narrow to only midwestern states
mw.states = as.data.frame(c('IA', 'IL', 'IN', 'KS', 'MI', 'MN', 'MO',
                            'NE', 'ND', 'OH', 'SD', 'WI'))
colnames(mw.states) = c('STATE')

df = merge(df.raw, mw.states, by='STATE') #inner joing to filter data
## reset state and county factor levels since some are no longer used
df$STATE = factor(df$STATE)
df$COUNTY = factor(df$COUNTY)
df$county.class = factor(df$county.class)
```

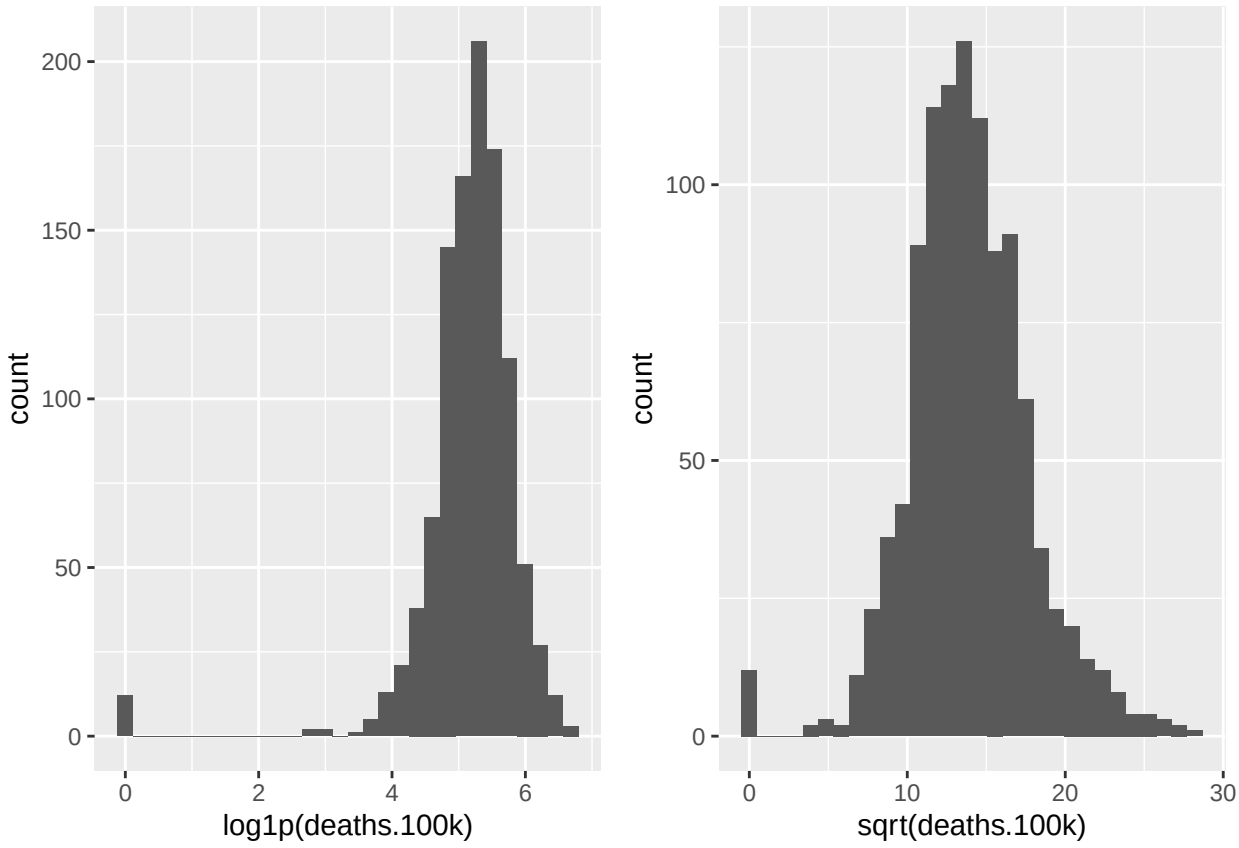
Exploratory Analysis

Start by looking at the distribution of the response variable, deaths per 100,000 population. It clearly has a long right tail and might benefit from a transformation.

```
ggplot(df, aes(x=deaths.100k)) + geom_histogram()
```

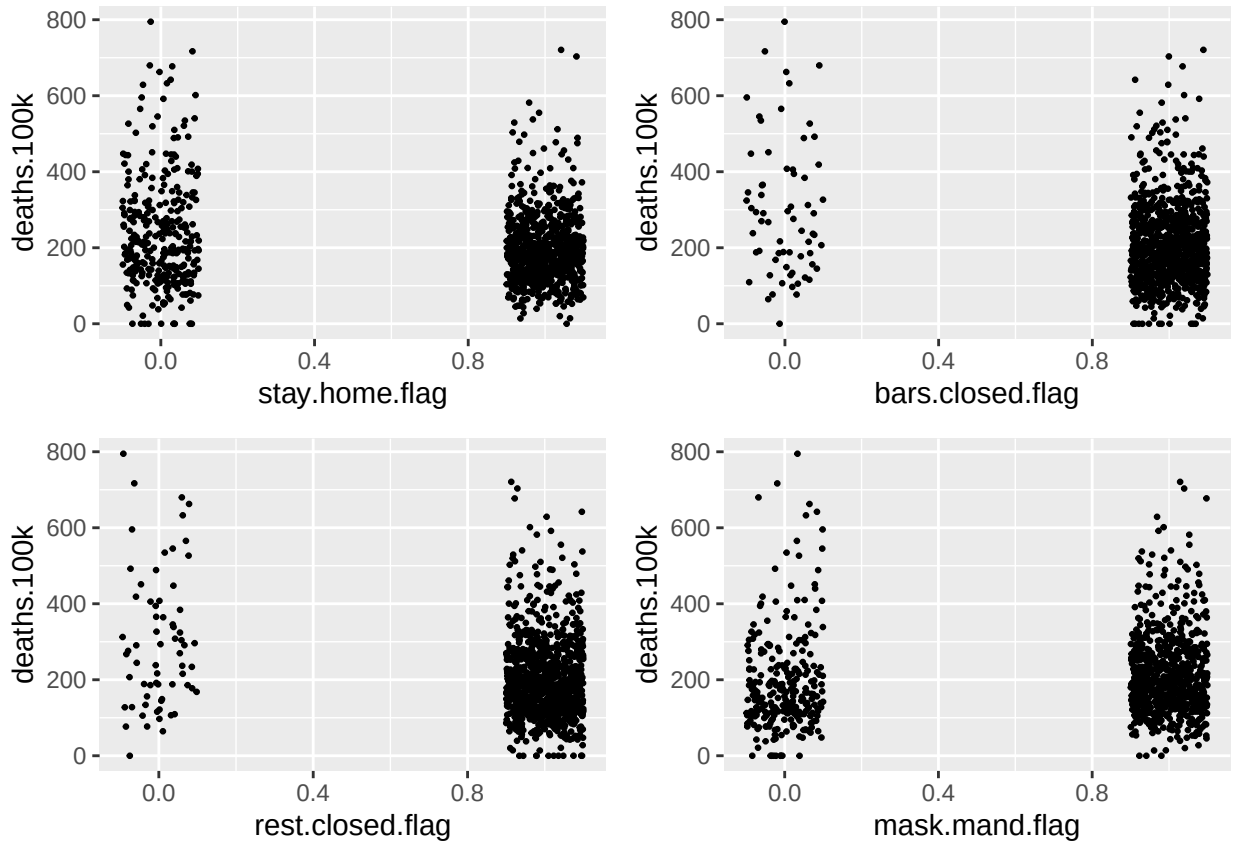



```
### What do log or sq. root tranformations look like? Square root looks better
### though the zeroes in the left tail might cause some problems.
p1 = ggplot(df, aes(x=log1p(deaths.100k))) + geom_histogram()
p2 = ggplot(df, aes(x=sqrt(deaths.100k))) + geom_histogram()
grid.arrange(p1, p2, nrow=1, ncol=2)
```



Compare the relationship between the response and categorical predictors.

```
predictors.cat = c('stay.home.flag', 'rest.closed.flag',
                   'bars.closed.flag', 'mask.mand.flag', 'county.class')
p1 = ggplot(df, aes(x=stay.home.flag, y=deaths.100k)) +
  geom_point(position = position_jitter(width=0.1, height=0.1),
             size=0.5)
p2 = ggplot(df, aes(x=bars.closed.flag, y=deaths.100k)) +
  geom_point(position = position_jitter(width=0.1, height=0.1),
             size=0.5)
p3 = ggplot(df, aes(x=rest.closed.flag, y=deaths.100k)) +
  geom_point(position = position_jitter(width=0.1, height=0.1),
             size=0.5)
p4 = ggplot(df, aes(x=mask.mand.flag, y=deaths.100k)) +
  geom_point(position = position_jitter(width=0.1, height=0.1),
             size=0.5)
grid.arrange(p1, p2, p3, p4, nrow=2, ncol=2)
```



A bit hard to tell, but there does appear to be a bit more variance in most categories among counties with no mandates. Also, the average deaths do appear to be slightly higher in counties with no stay-home orders or bar/restaurant mandates. And, appears the data for bars closed/restaurant closed flags might perfectly overlap, so one might need to be removed.

How widespread were the mandates?

```
### Bar/restaraunt order flags overlap perfectly, so I'll drop bars.closed
mean((df$bars.closed.flag==df$rest.closed.flag))
```

```
## [1] 1
```

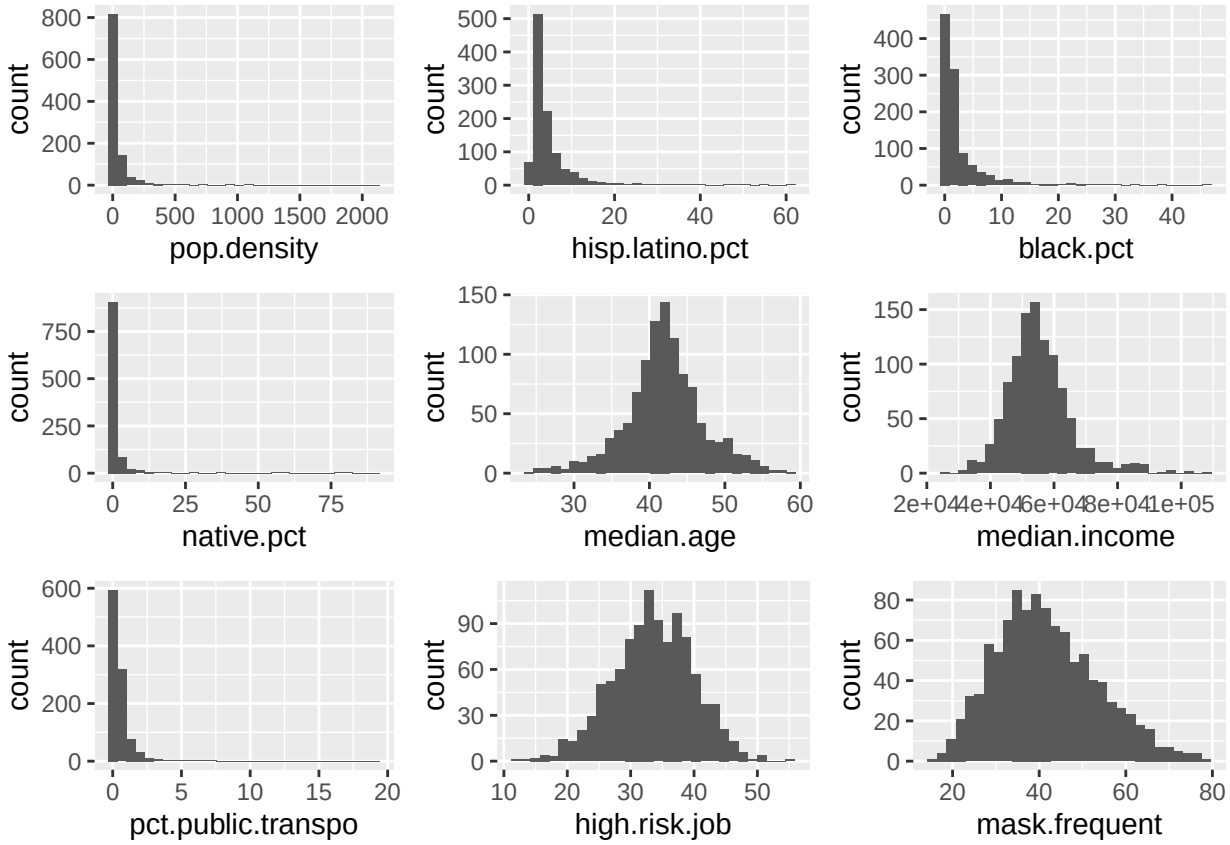
```
df = df %>% select(-c(bars.closed.flag))
print(c(mean(df$mask.mand.flag),
      mean(df$stay.home.flag),
      mean(df$rest.closed.flag)))
```

```
## [1] 0.7402844 0.7052133 0.9374408
```

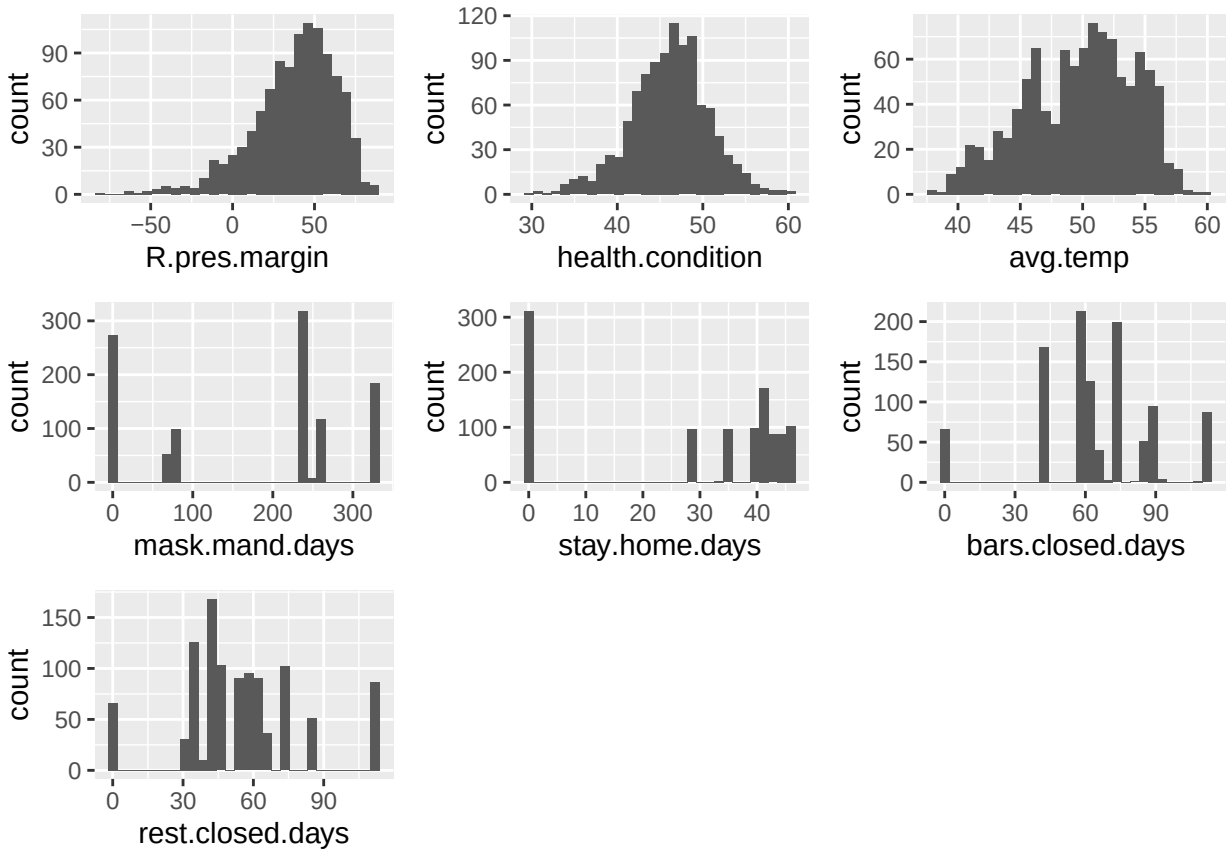
Let's take a closer look at the distribution of each continuous predictor.

```
predictors.cont = c('pop.density', 'hisp.latino.pct', 'black.pct',
                    'native.pct', 'median.age', 'median.income', 'pct.public.transpo',
                    'high.risk.job', 'mask.frequent', 'R.pres.margin', 'health.condition',
                    'avg.temp', 'mask.mand.days', 'stay.home.days',
```

```
                    'bars.closed.days', 'rest.closed.days')

plots = vector('list', length(predictors.cont))
i=0
for (pred in predictors.cont){
  i = i+1
  plots[[i]] = ggplot(df, aes_string(x=pred)) + geom_histogram()
}
grid.arrange(grobs=plots[1:9], nrow=3, ncol=3)
```

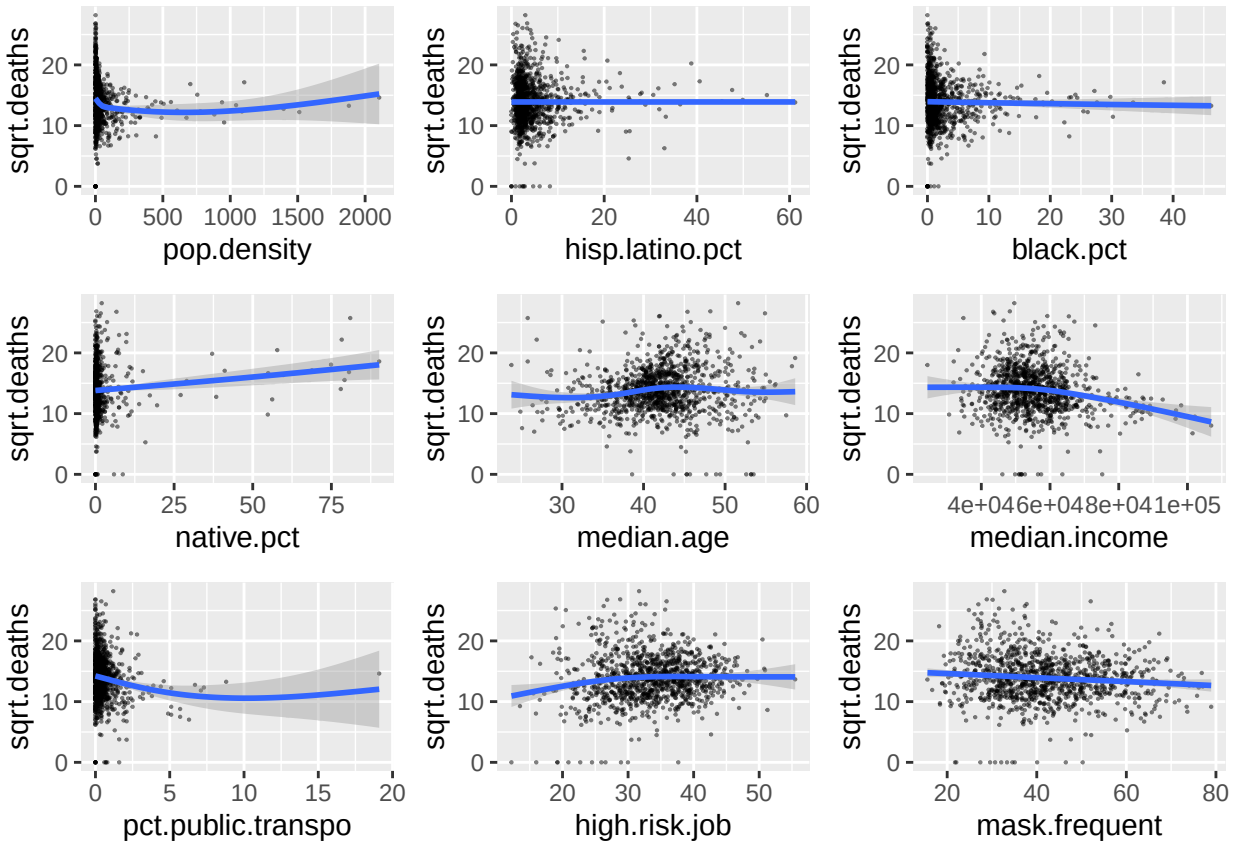



```
grid.arrange(grobs=plots[10:16], nrow=3, ncol=3)
```

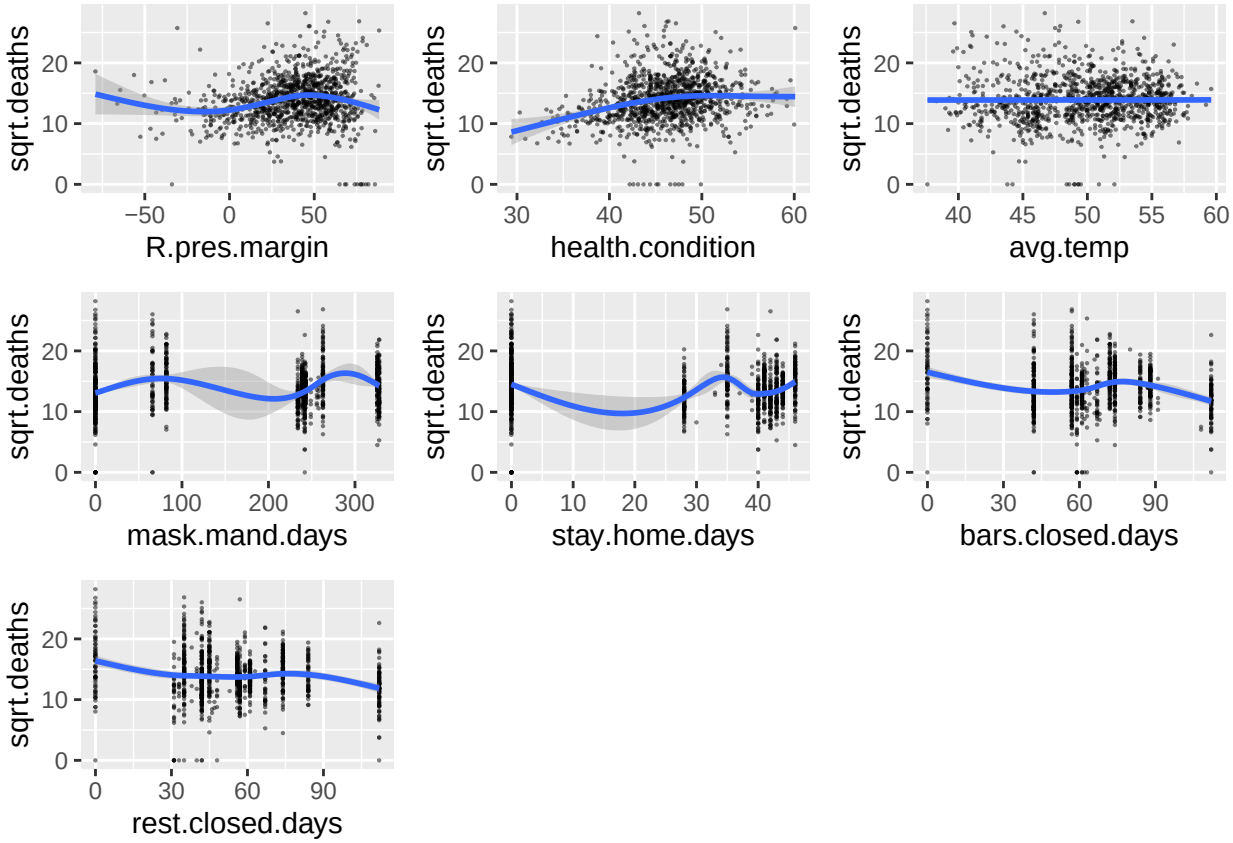


Some skewed distributions here (and also some discontinuity within some distributions). Before making transformations, let's look at the relationships between the (square-root transformed) response and continuous predictors.

```
df$sqrt.deaths = sqrt(df$deaths.100k)
plots = vector('list', length(predictors.cont))
i=0
for (pred in predictors.cont){
  i = i+1
  plots[[i]] = ggplot(df, aes_string(x=pred, y='sqrt.deaths')) +
                  geom_point(size=0.1, alpha=0.5) + geom_smooth()
}
grid.arrange(grobs=plots[1:9], nrow=3, ncol=3)
```




```
grid.arrange(grobs=plots[10:16], nrow=3, ncol=3)
```



Based on all the plots seen so far, it seems like many variables should at least be evaluated for transformations.

```
bc = powerTransform(cbind(deaths.100k, pop.density, hisp.latino.pct,
        black.pct, native.pct, median.age, median.income, high.risk.job,
        mask.frequent, pct.public.transpo, R.pres.margin, avg.temp,
        health.condition)~1, data=df,
        family='yjPower') ### using yj since some vars not strictly positive
summary(bc)
```

```
## yjPower Transformations to Multinormality
##                      Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## deaths.100k            0.5308        0.50       0.4738       0.5878
## pop.density           -0.1658       -0.17      -0.1979      -0.1338
## hisp.latino.pct       -0.4124       -0.41      -0.4849      -0.3399
## black.pct             -0.6520       -0.65      -0.7367      -0.5673
## native.pct            -1.3846       -1.38      -1.5087      -1.2606
## median.age             1.4390        1.44       1.1158       1.7622
## median.income         -0.2784       -0.33      -0.4763      -0.0805
## high.risk.job          1.1791        1.00       0.9453       1.4130
## mask.frequent          0.5003        0.50       0.3245       0.6761
## pct.public.transpo    -1.6514       -1.65      -1.8356      -1.4671
## R.pres.margin          1.1204        1.12       1.0952       1.1457
## avg.temp               1.6013        2.00       1.0160       2.1866
## health.condition       1.1787        1.00       0.7789       1.5785
##
##  Likelihood ratio test that all transformation parameters are equal to 0
```

```
##                                                    LRT df       pval
## LR test, lambda = (0 0 0 0 0 0 0 0 0 0 0 0 0) 14648.06 13 < 2.22e-16
```

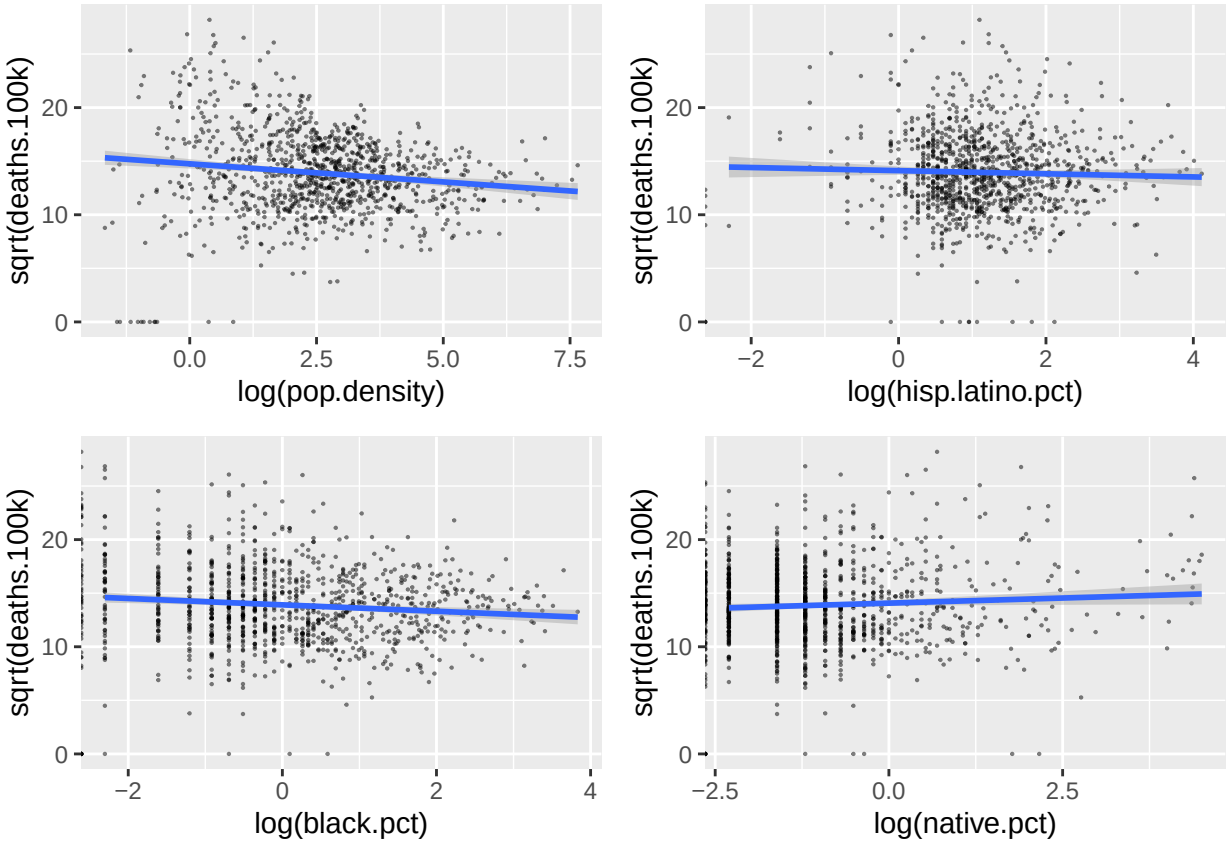
The above summary has some straightforward recommendations, but also several others that would produce unintuitive, confusing interpretations (in some cases reversing the direction of a relationship, such as with pct.public transpo). I'll include the following transformations:

- square-root transformation of the response (deaths.100k)
- log transformations of population density, Hispanic/Latino population pct, Black pct., and median income.

I'm hesitant to apply unintuitive transformations to some of the variables that I'd like to test later on (i.e. bars closed days).

To take a look again at scatter plots, but with transformed predictors:

```
p1 = ggplot(df, aes(x=log(pop.density), y=sqrt(deaths.100k))) +
  geom_point(size=0.1, alpha=0.5) + geom_smooth(method='lm')
p2 = ggplot(df, aes(x=log(hisp.latino.pct), y=sqrt(deaths.100k))) +
  geom_point(size=0.1, alpha=0.5) + geom_smooth(method='lm')
p3 = ggplot(df, aes(x=log(black.pct), y=sqrt(deaths.100k))) +
  geom_point(size=0.1, alpha=0.5) + geom_smooth(method='lm')
p4 = ggplot(df, aes(x=log(native.pct), y=sqrt(deaths.100k))) +
  geom_point(size=0.1, alpha=0.5) + geom_smooth(method='lm')
grid.arrange(p1,p2,p3,p4, nrow=2, ncol=2)
```

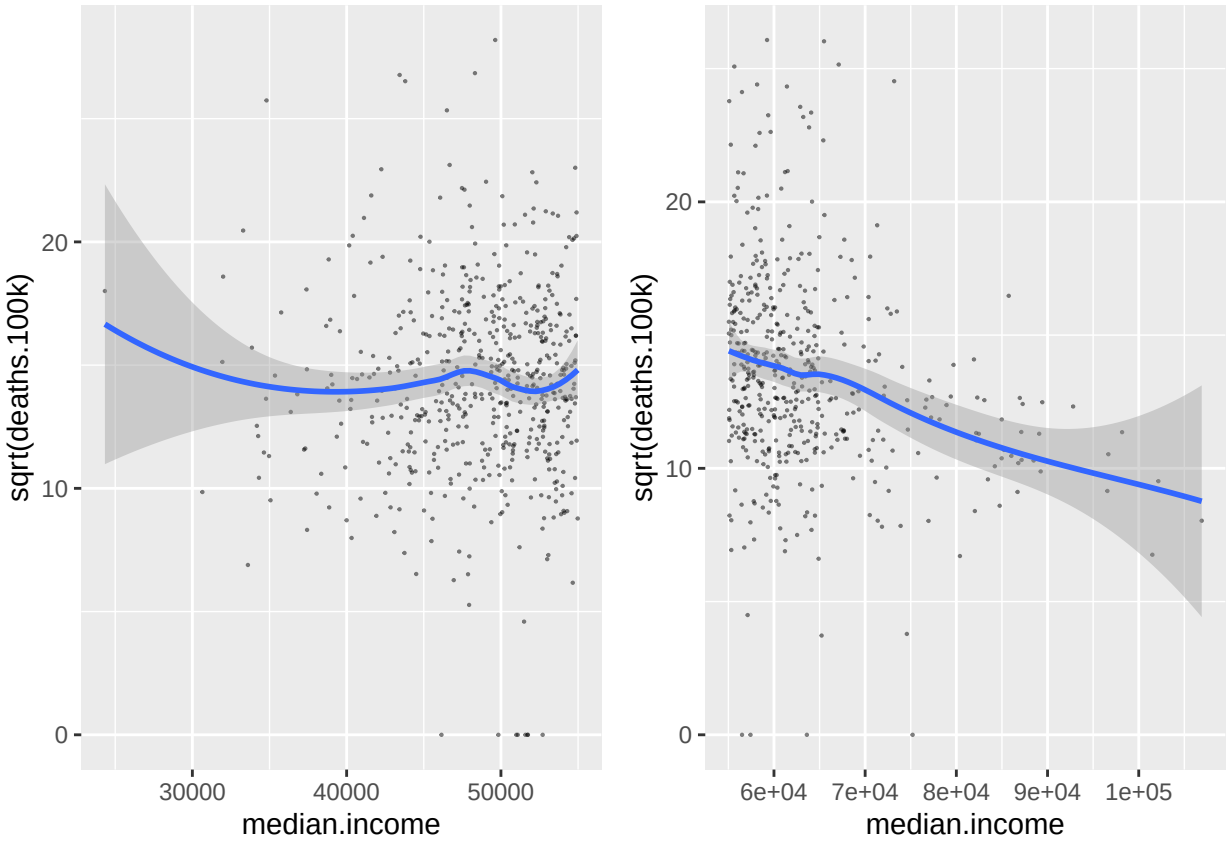


A concern with the last plot, for the Native American share of the population, is that while there isn't much

overall trend here, in counties with very high Native American populations, nearly all death rates values are above the linear trend line.

An additional concern I had with the prior median income plot was that while there appeared to be no trend below a certain income level, but at higher incomes, the slope clearly became negative. If we repeat the original plot using the untransformed incomes split at $55k, we can see there are completely different slopes

```
p1 = ggplot(df %>% filter(median.income <55000), aes(x=median.income,
      y=sqrt(deaths.100k))) +geom_point(size=0.1, alpha=0.5) +
  geom_smooth()
p2 = ggplot(df %>% filter((median.income > 55000)), aes(x=median.income,
      y=sqrt(deaths.100k))) +geom_point(size=0.1, alpha=0.5) +
  geom_smooth()
grid.arrange(p1,p2, nrow=1, ncol=2)
```




```
### Therefore, we will test using a "high income" flag
df$higher.income = ifelse(df$median.income >= 55000, 1,0)
```

One very unevenly distributed variable is pct. public transportation usage. Because the vast majority of counties have zero or a negligible rate of public transportation use, it doesn't really make sense to use it as a continuous variable. But if the variable is then binned by ranges of use, the numbers would suggest that lower rates of usage are associated with higher COVID death rates than all but one county with high usage. Because this doesn't make much intuitive sense and there are likely many other variables influencing higher rates in areas without public transportation, I'm going to exclude this variable moving forward.

```
df %>% mutate(pct.public.transpo.bin = cut_interval(pct.public.transpo, n=7)) %>%
  group_by(pct.public.transpo.bin) %>%
  summarize(mean(deaths.100k), median(deaths.100k), n())
```

```
## # A tibble: 5 x 4
##   pct.public.transpo.bin `mean(deaths.100k)` `median(deaths.100k)` `n()`
## * <fct>                                <dbl>                 <dbl> <int>
## 1 [0,2.73]                              209.                  190.  1030
## 2 (2.73,5.46]                           165.                  150.    17
## 3 (5.46,8.19]                           106.                  109.     6
## 4 (8.19,10.9]                           176.                  176.     1
## 5 (16.4,19.1]                           214.                  214.     1
```

Pct. Native American has a similar issue. Very few counties have sizable populations, but of the handful that do, deaths are markedly higher. Instead of using it as a continuous variable, I'll bin it into a factor.

```
df %>% mutate(pct.native.bin = cut_interval(native.pct, 3)) %>%
  group_by(pct.native.bin) %>%
  summarize(mean(deaths.100k), median(deaths.100k), n())
```

```
## # A tibble: 3 x 4
##   pct.native.bin `mean(deaths.100k)` `median(deaths.100k)` `n()`
## * <fct>                        <dbl>                 <dbl> <int>
## 1 [0,30.1]                      207.                  187.  1038
## 2 (30.1,60.1]                   258.                  273.     8
## 3 (60.1,90.2]                   352.                  324.     9
```

```
### add the above bin into the df
df = df %>% mutate(pct.native.bin = cut_interval(native.pct, 3))
```

Establishing a baseline mixed effects model

I'm interested in first tying to appropriately specify a linear mixed effects model that used state (and possibly county class) as a random effect and variables uninfluenced by COVID as fixed effects. I will then conduct a series of tests to determine significance of additional COVID-related variables, such as mandates, duration of mandates, and partisan political preference.

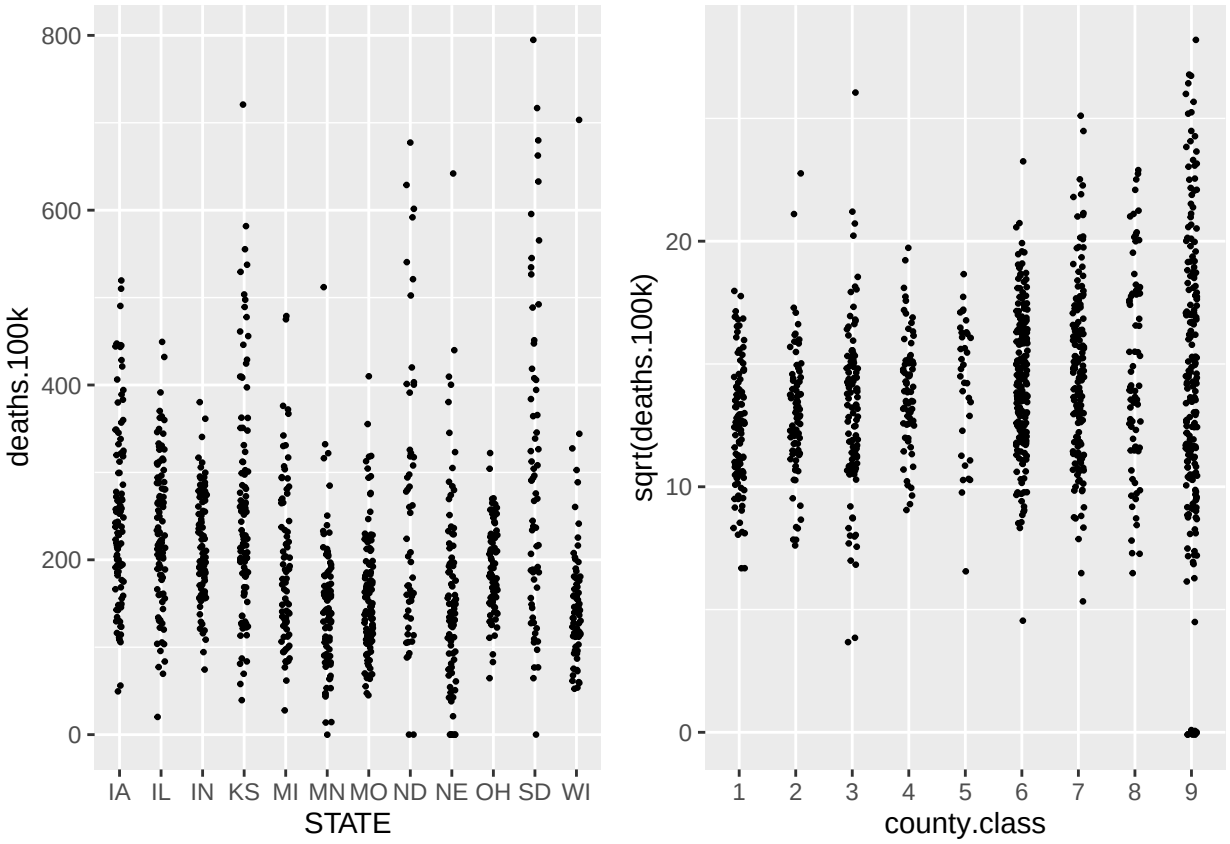
(NOTE: I Initially proposed to also use a Poisson rate model – however, the large range of the offset parameter, population, created challenges that prevented the model from being able to converge. Similar problems were encountered when trying to fit a Poisson mixed effects model (using glmer), so both approaches are not implemented here.

Plot potential random effects

Deaths clearly vary by state, so it seems this will be a useful effect to consider. I'm not sure there's quite as much apparent variance in county classes, but it's still worth considering as an effect.

```
p1 = ggplot(df, aes(x=STATE, y=deaths.100k)) +
  geom_point(position = position_jitter(width=0.1, height=0.1),
             size=0.5)
```

```
p2 = ggplot(df, aes(x=county.class, y=sqrt(deaths.100k))) +
  geom_point(position = position_jitter(width=0.1, height=0.1), size=0.5)
grid.arrange(p1, p2, nrow=1, ncol=2)
```



In a null model with only an intercept, let's test some of the random effects to get a better sense of whether they're worth considering in a larger model.

```
m.state = lmer(sqrt(deaths.100k)~1+(1|STATE),data=df)
m.county = lmer(sqrt(deaths.100k)~1+(1|county.class),data=df)
m.state.county = update(m.county, .~. + (1|STATE))

### first, test for state
exactRLRT(m.state, m.state.county, m.county)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 161.06, p-value < 2.2e-16
```

```
### then for county class
exactRLRT(m.county, m.state.county, m.state)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 17.667, p-value < 2.2e-16
```

When added to a model containing only an intercept, both random effects appear significant.

Establish baseline model

The primary variables whose significance I'm interested in testing are:

- mask mandates
- length of mask mandates
- partisan political preference
- stay at home orders
- length of stay at home orders
- bar/restaurant closings
- length of bar/restaurant closings
- self-reported mask usage

To test these variables I'll begin by comparing their influence to a baseline model that considers variables that would not have been influenced by COVID:

- population density
- race and ethnicity
- median age
- median income
- high risk job
- share of population with underlying health conditions
- average temperature

Because we're using grouped data and the populations of counties vary widely, I at first tried to use population as weights. However, as can be seen below, the huge range for population seems to create issues and the model either won't converge or produces a singular fit (depending on which variables are added/removed. I couldn't find a combination that eliminated this issue.) Shrinking the scale of the weights by applying the same response transformation (square root) eliminated the error, but I couldn't locate a theoretical basis for doing this and was not confident it was an appropriate choice, so I decided not to proceed with it. Needless to say, not applying weights in what is essentially an ecological regression situation is problematic in itself, but I could find a way to appropriately incorporate them.

```
m.base = lmer(sqrt(deaths.100k)~avg.temp+
                health.condition+
                high.risk.job+
                log(median.income)*higher.income+
                log1p(black.pct)+
                pop.density+
                log1p(hisp.latino.pct)+
                pct.native.bin+
                median.age+
```

```
                (1|STATE)+(1|county.class),
                weights=population,
              data=df)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
```

Same model as above, but with no weights:

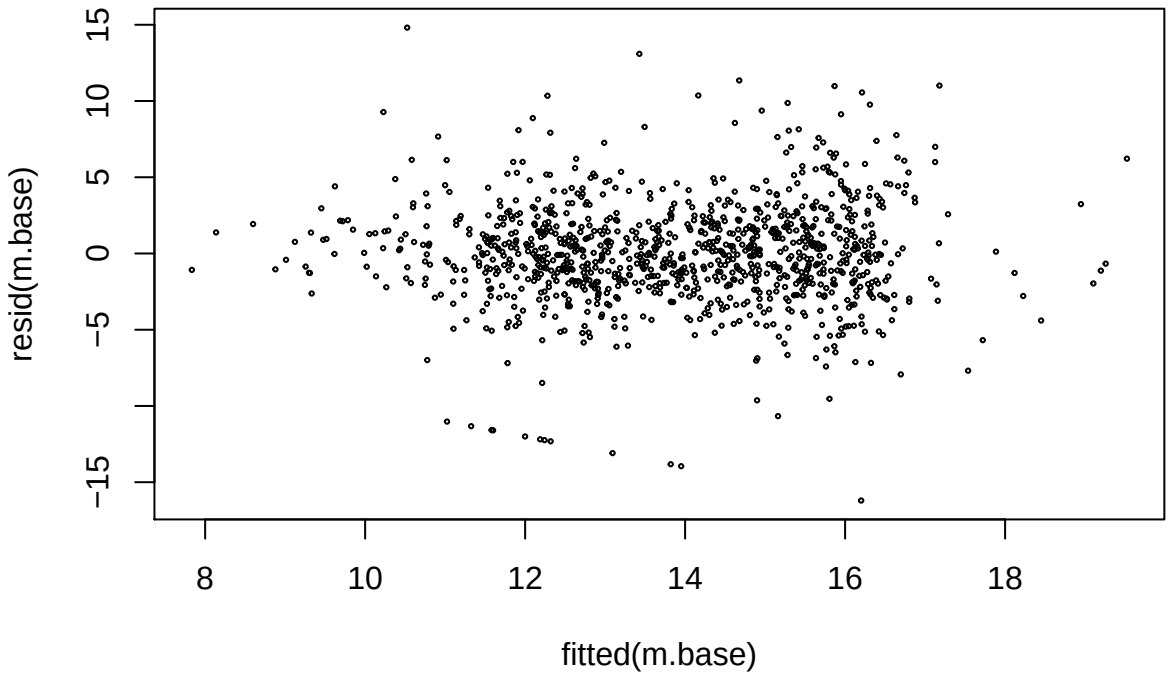
```
m.base = lmer(sqrt(deaths.100k)~avg.temp+
                health.condition+
                high.risk.job+
                log(median.income)*higher.income+
                log1p(black.pct)+
                pop.density+
                log1p(hisp.latino.pct)+
                pct.native.bin+
                median.age+
                (1|STATE)+(1|county.class),
              data=df)
sumary(m.base)
```

```
## Fixed Effects:
##                                  coef.est coef.se
## (Intercept)                      -2.55    18.28
## avg.temp                         -0.04     0.05
## health.condition                  0.09     0.04
## high.risk.job                     0.03     0.02
## log(median.income)                0.89     1.60
## higher.income                    64.64    21.92
## log1p(black.pct)                 -0.17     0.20
## pop.density                       0.00     0.00
## log1p(hisp.latino.pct)            0.45     0.21
## pct.native.bin(30.1,60.1]         0.88     1.38
## pct.native.bin(60.1,90.2]         3.44     1.50
## median.age                        0.09     0.03
## log(median.income):higher.income -5.90     2.01
##
## Random Effects:
##  Groups       Name        Std.Dev.
##  STATE        (Intercept) 1.72
##  county.class (Intercept) 0.21
##  Residual                 3.41
## ---
## number of obs: 1055, groups: STATE, 12; county.class, 9
## AIC = 5666.3, DIC = 5580.9
## deviance = 5607.6
```

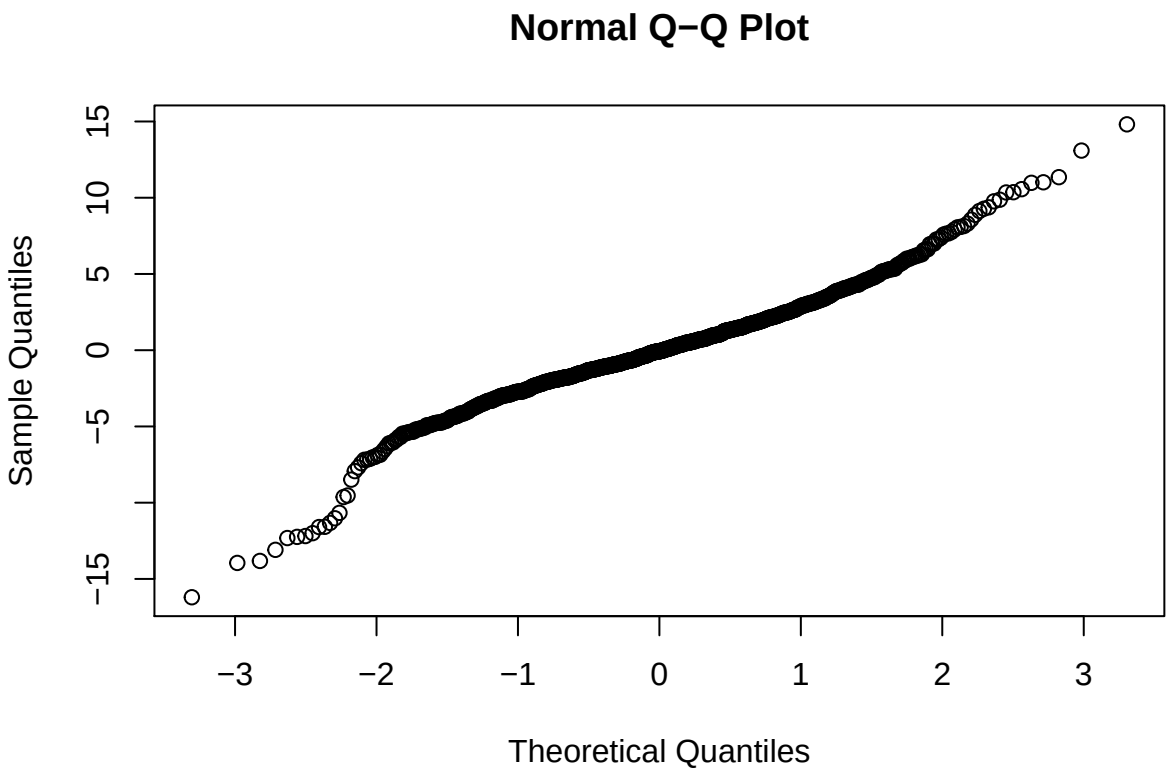
The below plots shows that residuals are normally distributed, for the most part, though they have some mild heteroscedasticity. However, there is an odd diagonal at the bottom of the plot, all of boundary points

in which counties had zero deaths. If I take a look closer, 11 of 13 zero-death points had fewer than 1200 people, and were among the 15 least populous counties in the data set. The other two still had fewer than 6,000 people. If we look closer at diagnostics (below, there aren't any overly influential points, based on Cook's distances, but there are some outliers). Because random effects models are particularly sensitive to outliers (Faraway, p. 211), I'd prefer not to leave these unadressed.

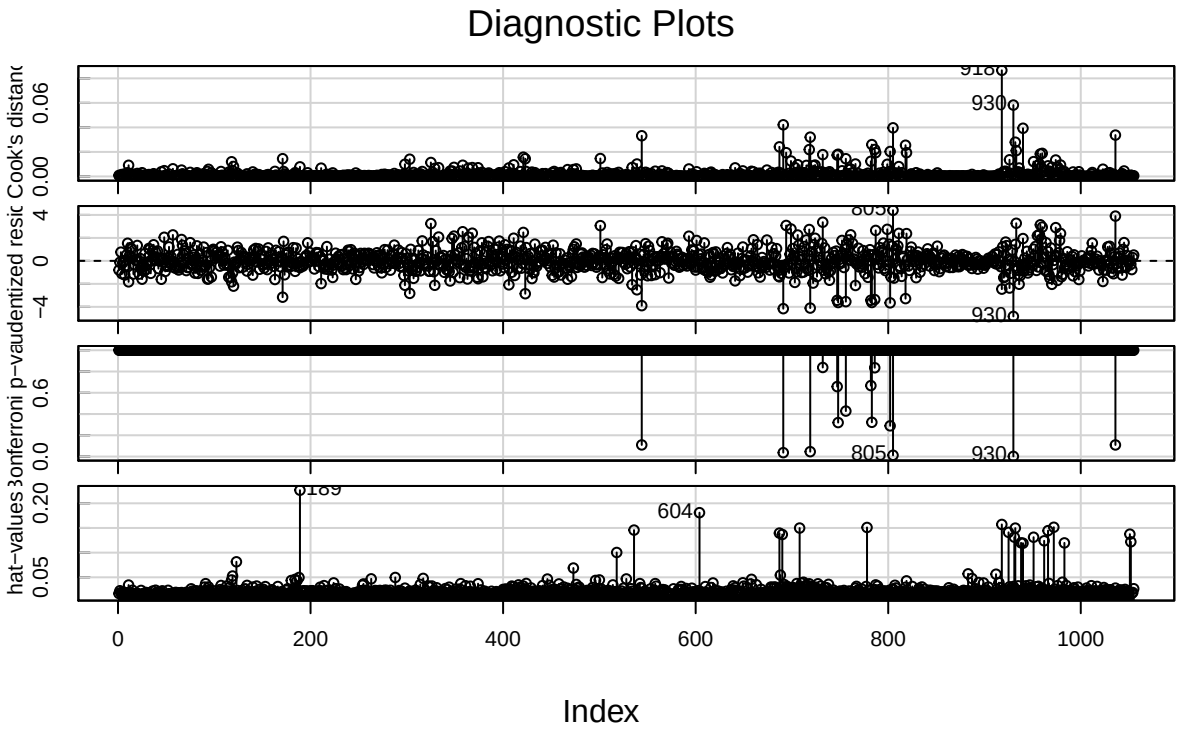
```
plot(fitted(m.base), resid(m.base), cex=0.3)
```




```
qqnorm(resid(m.base))
```




```
influenceIndexPlot(m.base)
```




```
outlierTest(m.base)
```

```
##      rstudent unadjusted p-value Bonferroni p
## 930 -4.831600         1.5583e-06     0.001644
## 805  4.405024         1.1674e-05     0.012316
## 691 -4.158841         3.4629e-05     0.036533
## 719 -4.106072         4.3406e-05     0.045793
```

Two of the above outliers have zero deaths, and one has four, though its very small population causes it's deaths/100k rate to be extreme.

```
df %>% slice(930, 805, 691) %>%
  dplyr::select(STATE, COUNTY, population, deaths, deaths.100k) %>%
  arrange(population)
```

```
##   STATE COUNTY population deaths deaths.100k
## 1    NE     75        623      4    642.0546
## 2    ND     87        750      0      0.0000
## 3    SD     75        903      0      0.0000
```

If we update the above model to exclude the outliers (above), it reveals two new ones, including another very small county with zero deaths. I'll spare you the tedium, but if I were to continue down this path, each time an additional one is removed, another 1-3 outliers is revealed – nearly all of them small counties with zero deaths.

```
m.base.noOutliers = update(m.base, subset=-c(930, 805, 691))
outlierTest(m.base.noOutliers)
```

```
##      rstudent unadjusted p-value Bonferroni p
## 719 -4.324291         1.6781e-05     0.017654
```

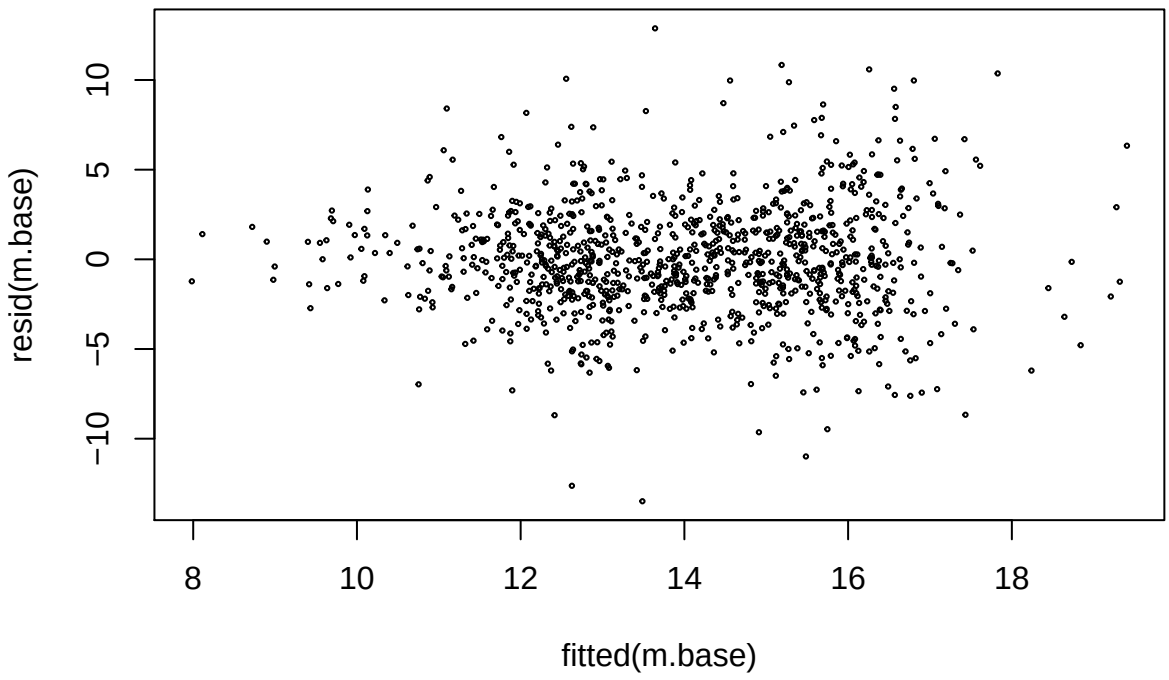
If we take a closer look at 16 counties under 1,200 people, which only represent a little over 1 percent of the data set, 12 are boundary points or outliers (11 zeroes and one four).

```
df %>% filter(population <=1200) %>%
  dplyr::select(STATE, COUNTY, deaths, population, deaths.100k)
```

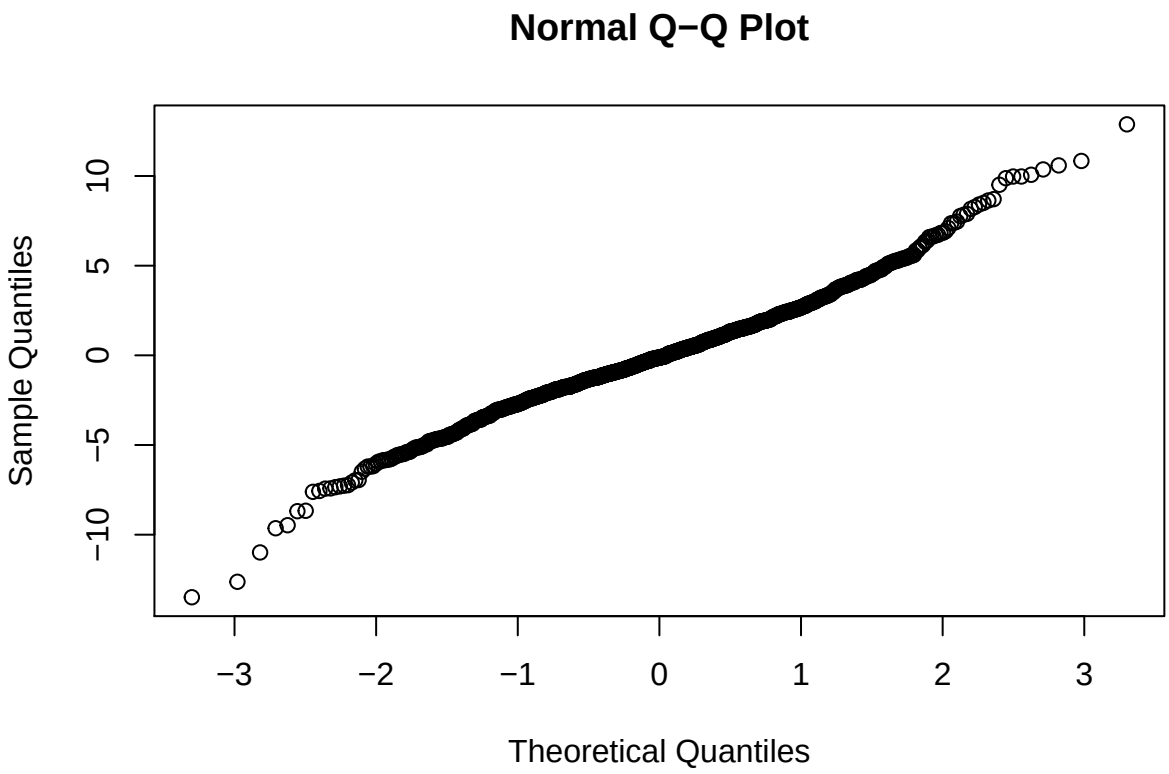
```
##    STATE COUNTY deaths population deaths.100k
## 1     ND     87      0        750     0.00000
## 2     ND      7      0        928     0.00000
## 3     NE    113      0        748     0.00000
## 4     NE    115      0        664     0.00000
## 5     NE    117      1        494   202.42915
## 6     NE    183      0        783     0.00000
## 7     NE    165      1       1166    85.76329
## 8     NE    171      1        722   138.50416
## 9     NE     85      0        922     0.00000
## 10    NE      9      0        465     0.00000
## 11    NE     91      3        682   439.88270
## 12    NE      5      1        463   215.98272
## 13    NE      7      0        745     0.00000
## 14    NE     75      4        623   642.05457
## 15    NE    103      0        806     0.00000
## 16    SD     75      0        903     0.00000
```

Rather than cherry pick outliers, I'll experiment with limiting the volatility caused by the least populous counties by fitting the model without observations with fewer than 1200 people, which still preserves nearly 99 percent of the data.

```
m.base = update(m.base, data=df %>% filter(population >=1200))
plot(fitted(m.base), resid(m.base), cex=0.3)
```



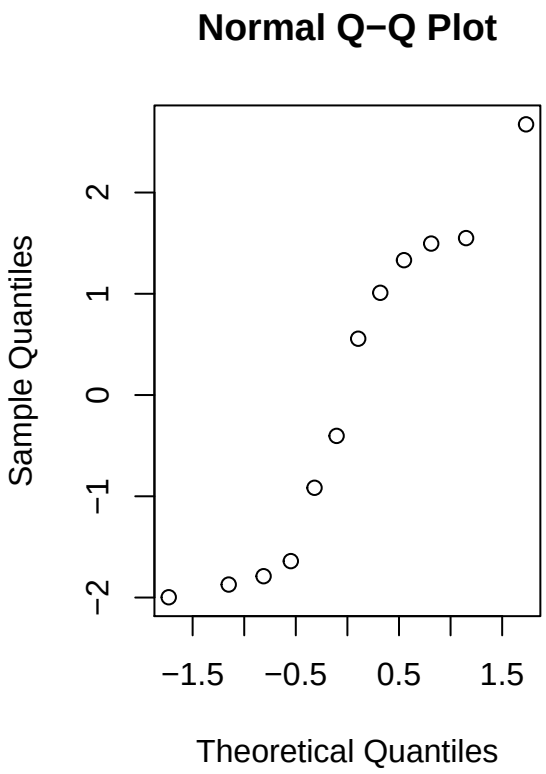

```
qqnorm(resid(m.base))
```

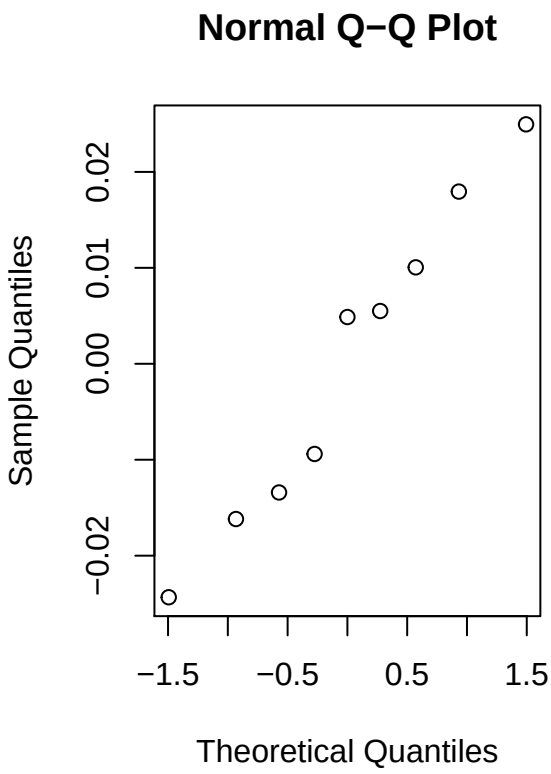


The residual and QQ-norm plots are both improved.

Take a closer look at the random effects in the baseline model.

```
## The random effects aren't perfectly normally distributed,
## but they're not overly concerning
par(mfrow=c(1,2))
qqnorm(ranef(m.base)$STATE[[1]])
qqnorm(ranef(m.base)$county.class[[1]])
```

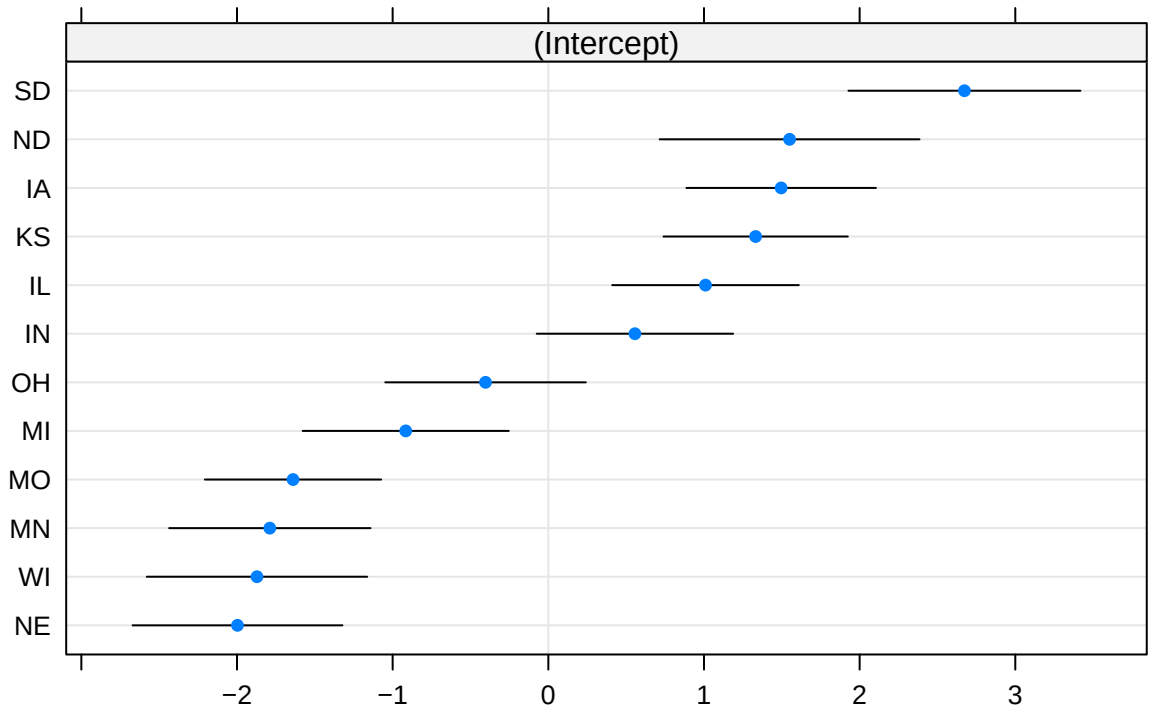




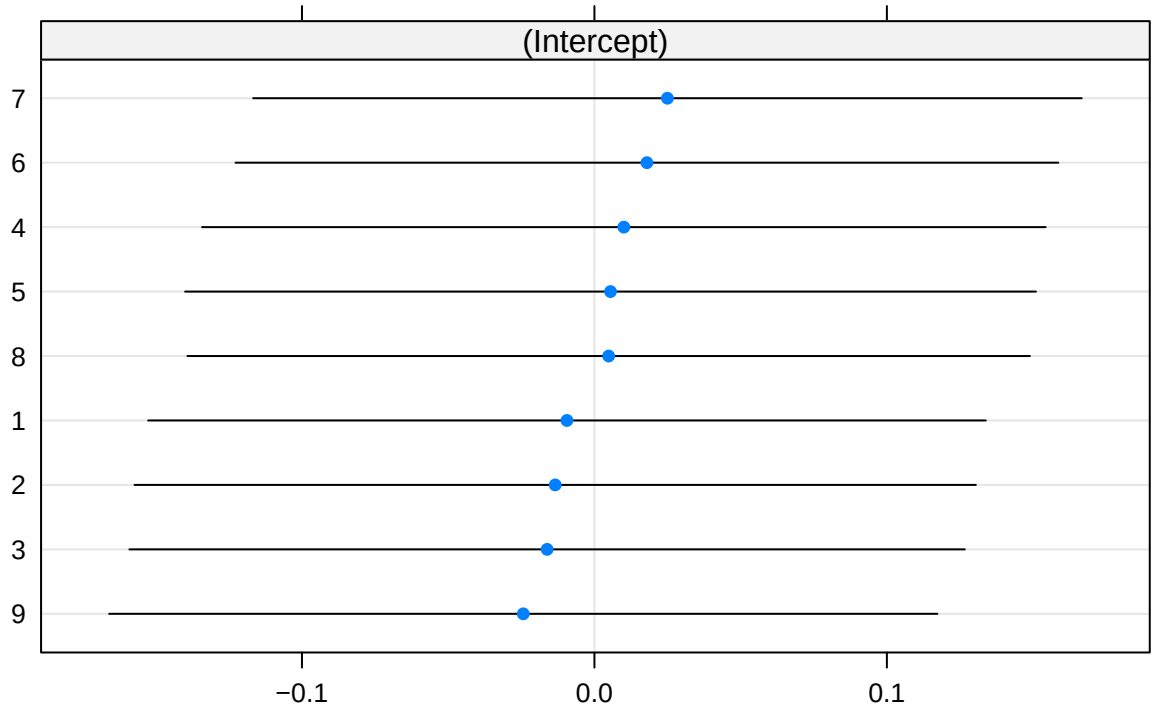
If we look at the confidence intervals (below), the effects for state clearly still seem significant. Not the case for county class, whose effects are quite small. Because I'm curious whether the effects are more relevant when new variables are added to the model, I'll opt to keep it in models moving forward.

```
dotplot(ranef(m.base), condVar=TRUE)
```

```
## $STATE
```




```
##
## $county.class
```



When the confidence intervals for fixed and random effects are bootstrapped, it is again apparent that the county class effect (sig02) is on the boundary and likely not significant. In this base model, several of the fixed effects do not appear to be significant (i.e. average temperature and the log-transformed pct. Hispanic/latino predictor), but I'd still like to include them since excluding them could affect the estimates for the variables I'm interested in testing. The interaction between income and the higher income factor appears to be influential.

```
confint(m.base, method='boot')
```

```
##                                       2.5 %       97.5 %
## .sig01                          9.428960e-01   2.511408257
## .sig02                          0.000000e+00   0.371508693
## .sigma                          2.991607e+00   3.265693317
## (Intercept)                    -4.863264e+01  17.749235237
## avg.temp                       -1.089983e-01   0.086772627
## health.condition                2.537537e-02   0.166555201
## high.risk.job                  -3.448910e-02   0.048753574
## log(median.income)             -9.193420e-01   4.886004491
## higher.income                   3.843102e+01 117.919431764
## log1p(black.pct)               -5.931527e-01   0.139322835
## pop.density                    -5.939095e-04   0.002919595
## log1p(hisp.latino.pct)          2.105737e-02   0.846779520
## pct.native.bin(30.1,60.1]      -1.729912e+00   2.854561107
## pct.native.bin(60.1,90.2]       8.916717e-01   6.078999556
## median.age                      5.091337e-02   0.160904254
## log(median.income):higher.income -1.080340e+01  -3.520361938
```

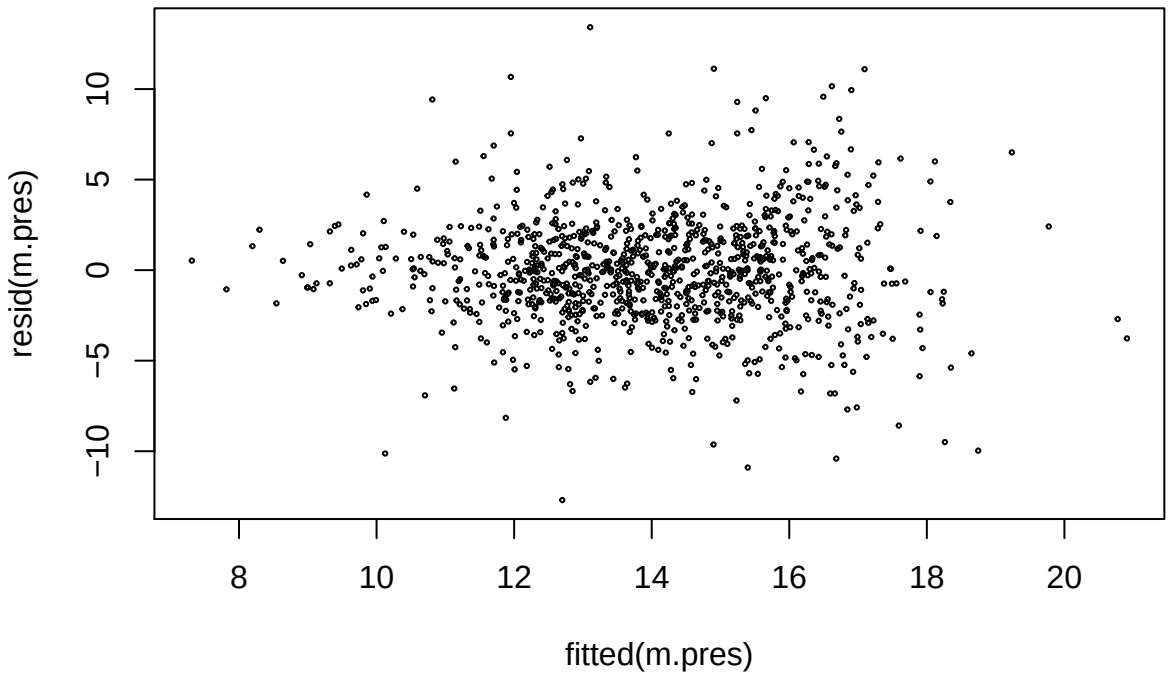
Test predictors of interest

Start by testing presidential vote preference. According to the Kenward Roger adjusted F-test, adding this to the base model is an improvement.

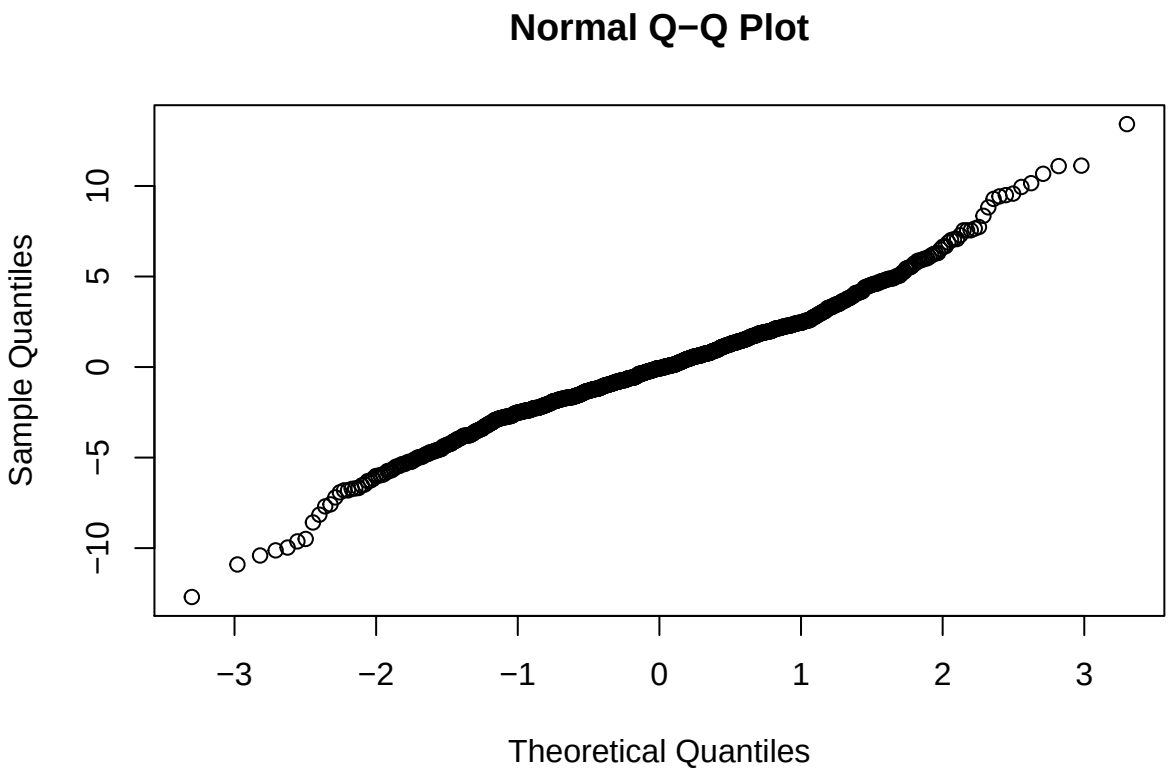
```
m.pres = update(m.base, .~. + R.pres.margin)
KRmodcomp(m.pres, m.base)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + R.pres.margin + log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##          stat     ndf     ddf F.scaling   p.value
## Ftest  47.996   1.000 590.143         1 1.121e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# check residuals. Not much different than before. Nothing too concerning.
plot(fitted(m.pres), resid(m.pres), cex=0.3)
```



```
qqnorm(resid(m.pres))
```

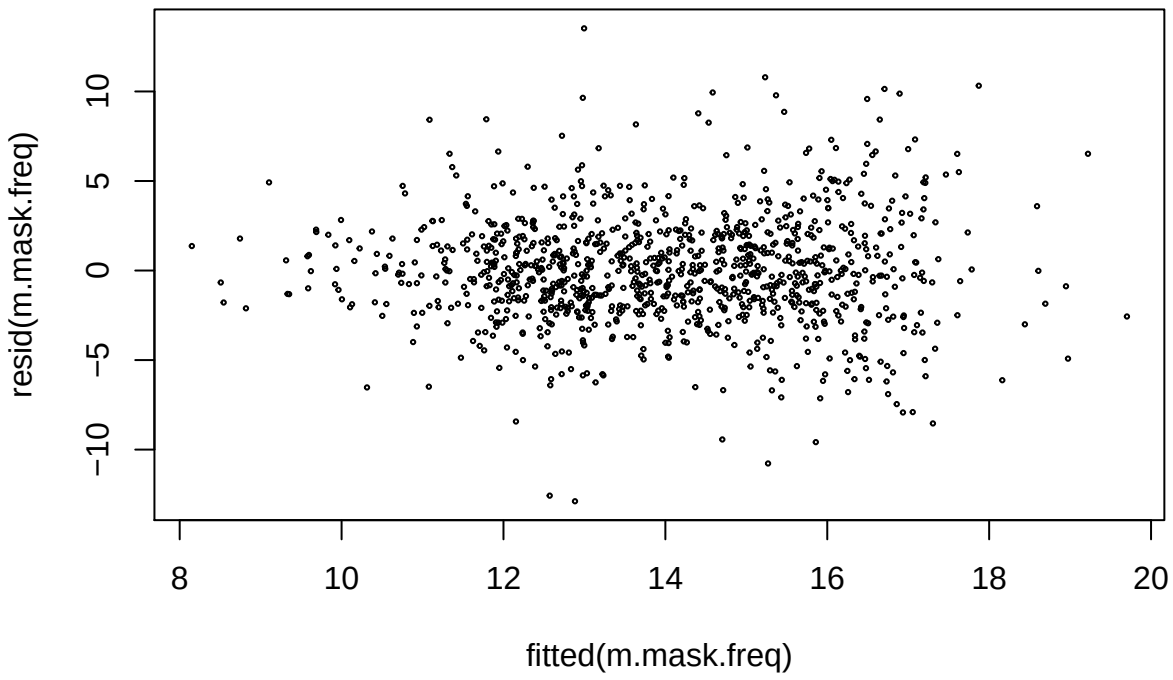


Compared to the base model, does reported mask usage improve the model? According to a Kenward Roger test, it does.

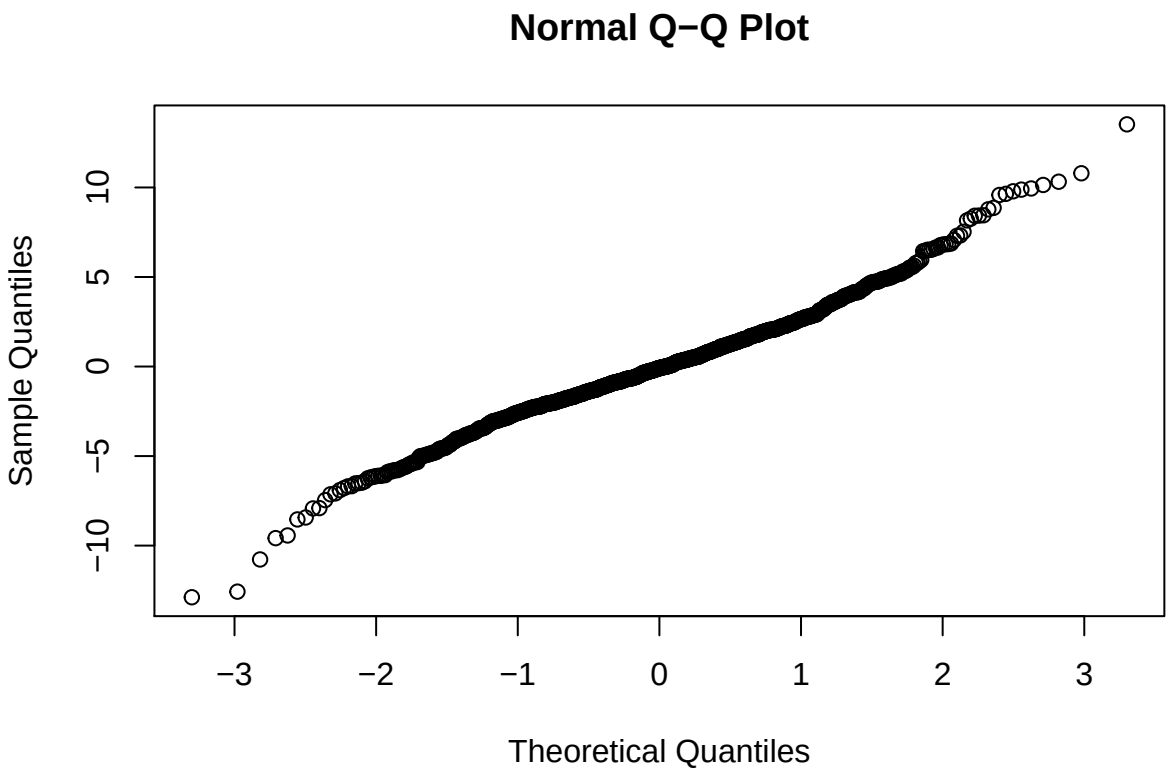
```
m.mask.freq = update(m.base, .~.+mask.frequent)
KRmodcomp(m.mask.freq, m.base)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + mask.frequent + log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##          stat     ndf     ddf F.scaling  p.value
## Ftest  17.236   1.000 856.313         1 3.63e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Residuals look fine
plot(fitted(m.mask.freq), resid(m.mask.freq), cex=.3)
```




```
qqnorm(resid(m.mask.freq))
```



Does including the enactment (and not the duration, which is a separate variable) of bar and restaurant closures improve the model? According to the below Kenward Roger test, it does not. A test using parametric bootstrapping supports this.

```
m.rest.closed = update(m.base, .~.+rest.closed.flag)
KRmodcomp(m.rest.closed, m.base)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + rest.closed.flag + log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##          stat     ndf     ddf F.scaling p.value
## Ftest  3.6756  1.0000 10.2314         1 0.08354 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### since that p-value wasn't large, double-checking with PB test
PBmodcomp(m.rest.closed, m.base)
```

```
## Bootstrap test; time: 63.57 sec; samples: 1000; extremes: 70;
## Requested samples: 1000 Used samples: 654 Extremes: 70
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + rest.closed.flag + log(median.income):higher.income
## sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##          stat df p.value
## LRT    3.6678  1 0.05547 .
## PBtest 3.6678    0.10840
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What about the duration of restaurant closures? Not significant, according to the below Kenward-Rogers and bootstrapping tests.

```
### since I'd argue you can't test this without including the binary
### rest/bars closed flag, I'm updating the previous model and not m.base
m.rest.closed.days = update(m.rest.closed, .~. + rest.closed.days)
KRmodcomp(m.rest.closed.days, m.base)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + rest.closed.flag + rest.closed.days +
##     log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##          stat     ndf     ddf F.scaling p.value
## Ftest  1.6986  2.0000 14.6786   0.99499   0.217
```

What about the duration of bar closures? That doesn't improve the model either, per the below tests.

```
m.bars.closed.days = update(m.rest.closed, .~. + bars.closed.days)
KRmodcomp(m.bars.closed.days, m.base)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + rest.closed.flag + bars.closed.days +
##     log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##          stat     ndf     ddf F.scaling p.value
## Ftest  1.8242  2.0000 14.3541   0.99539  0.1968
```

What about stay-at-home orders (the enactment, not the the duration, which is a separate variable)? That doesn't improve the model either, per the below tests.

```
m.stay.order = update(m.base, .~. + stay.home.flag)
KRmodcomp(m.stay.order, m.base)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + stay.home.flag + log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##          stat     ndf     ddf F.scaling p.value
## Ftest  2.1995  1.0000 10.3501         1  0.1679
```

```
PBmodcomp(m.stay.order, m.base)
```

```
## Bootstrap test; time: 59.86 sec; samples: 1000; extremes: 173;
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + stay.home.flag + log(median.income):higher.income
## sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##          stat df p.value
## LRT    2.2756  1  0.1314
## PBtest 2.2756     0.1738
```

What about the length of the stay-at-home orders? That does not improve the model according to the below test.

```
### since I'd argue you can't test this without including the binary
### stay at home flag, I'm updating the previous model and not m.base
m.stay.order.days = update(m.stay.order, .~. + stay.home.days)
KRmodcomp(m.stay.order.days, m.base)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + stay.home.flag + stay.home.days +
##     log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##          stat     ndf     ddf F.scaling p.value
## Ftest  1.1931  2.0000 20.8938   0.98606  0.3231
```

What about mask mandates (the establishing of them, not the duration?). That did not improve the model, according to the below tests.

```
m.mask.mand = update(m.base, .~. + mask.mand.flag)
KRmodcomp(m.mask.mand, m.base)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + mask.mand.flag + log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##         stat    ndf    ddf F.scaling p.value
## Ftest 0.1410 1.0000 9.9373         1  0.7152
```

What about the duration of the mask mandates? Did not improve the model, per the below tests.

```
m.mask.mand.days = update(m.mask.mand, .~. + mask.mand.days)
KRmodcomp(m.mask.mand.days, m.base)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + mask.mand.flag + mask.mand.days +
##     log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) * higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class)
##         stat    ndf    ddf F.scaling p.value
## Ftest 0.6697 2.0000 9.1607   0.99997  0.5352
```

From the above tests, only two variables, partisan vote preference and frequency of mask use, were significant improvements over a baseline model. Is a model that includes both an improvement over a model that only includes vote preference? Not according to the below tests.

```
m.pres.mask = update(m.pres, .~. + mask.frequent)
KRmodcomp(m.pres.mask, m.pres)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + R.pres.margin + mask.frequent +
##     log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + R.pres.margin + log(median.income):higher.income
##          stat     ndf     ddf F.scaling p.value
## Ftest   2.739   1.000 996.069         1 0.09824 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
PBmodcomp(m.pres.mask, m.pres)
```

```
## Bootstrap test; time: 68.65 sec; samples: 1000; extremes: 99;
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + R.pres.margin + mask.frequent +
##     log(median.income):higher.income
## sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + R.pres.margin + log(median.income):higher.income
##          stat df p.value
## LRT    2.7788  1 0.09552 .
## PBtest 2.7788    0.09990 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What about when compared to a model that only contains mask frequency?

```
KRmodcomp(m.pres.mask, m.mask.freq)
```

```
## large : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + R.pres.margin + mask.frequent +
##     log(median.income):higher.income
## small : sqrt(deaths.100k) ~ avg.temp + health.condition + high.risk.job +
##     log(median.income) + higher.income + log1p(black.pct) + pop.density +
##     log1p(hisp.latino.pct) + pct.native.bin + median.age + (1 |
##     STATE) + (1 | county.class) + mask.frequent + log(median.income):higher.income
##          stat     ndf     ddf F.scaling   p.value
## Ftest  33.797   1.000 767.714         1 8.967e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Adding presidential preference over a model that only contains mask frequency is clearly an improvement. So then if presidential vote preference clearly should be included, is mask frequency still significant?

```
# If we look at AIC, a model with vote preference and mask frequency actually
# has a higher value than if just including vote preference.
print(AIC(m.pres.mask, m.pres, m.mask.freq))
```

```
##             df      AIC
## m.pres.mask 18 5373.762
## m.pres      17 5367.603
## m.mask.freq 17 5397.257
```

```
### Let's bootstrap the confidence intervals for effects to look more closely
### at mask frequency.
confint(m.pres.mask, method='boot')
```

```
##                                         2.5 %       97.5 %
## .sig01                              0.885858793   2.329053136
## .sig02                              0.000000000   0.530195293
## .sigma                              2.901115027   3.184926381
## (Intercept)                       -37.359657048  27.919445321
## avg.temp                           -0.179452308   0.014948524
## health.condition                   -0.035148475   0.108196476
## high.risk.job                      -0.070464831   0.023240601
## log(median.income)                 -1.316510358   4.249809296
## higher.income                      34.053249988 118.076658977
## log1p(black.pct)                   -0.077445211   0.652973786
## pop.density                         0.001401259   0.005010231
## log1p(hisp.latino.pct)              0.193833597   0.917759967
## pct.native.bin(30.1,60.1]           0.107366957   4.892616591
## pct.native.bin(60.1,90.2]           4.798984917  10.797836474
## median.age                          0.043019768   0.147431260
## R.pres.margin                       0.031104905   0.063687532
## mask.frequent                      -0.046008204   0.006311382
## log(median.income):higher.income -10.783098801  -3.110577384
```

Mask frequency does not appear to be significant according to the above 95% interval. Additionally, since the AIC for a model with only voter preference is lower than a model that also contains mask frequency, it seems likely that only presidential vote preference provides a significant improvement over the baseline set of predictors.

Model selection

Another approach to identifying a model would be to use stepwise model selection. I couldn't figure out a way to implement stepwise using a mixed-effects model, but a manual, somewhat tedious equivalent of forward stepwise is implemented below. Since we already know that partisan vote preference was an improvement in AIC over the base model, that will serve as my starting point. (I'm also skipping the next predictor, mask usage since I know the AIC for that was lower than in the model with partisan preference)

```
fm.1 = update(m.pres, .~. + rest.closed.flag)
AIC(m.pres, fm.1) # rest.closed.flag is an improvement
```

```
##        df      AIC
## m.pres 17 5367.603
## fm.1   18 5364.353
```

```
fm.2 = update(fm.1, .~. + rest.closed.days)
AIC(fm.1, fm.2) # not an improvement
```

```
##      df      AIC
## fm.1 18 5364.353
## fm.2 19 5370.708
```

```
fm.3 = update(fm.1, .~. + bars.closed.days)
AIC(fm.1, fm.3) # not an improvement
```

```
##      df      AIC
## fm.1 18 5364.353
## fm.3 19 5370.631
```

```
fm.4 = update(fm.1, .~. + stay.home.flag)
AIC(fm.1, fm.4) # improvment
```

```
##      df      AIC
## fm.1 18 5364.353
## fm.4 19 5364.218
```

```
fm.5 = update(fm.4, .~. + stay.home.days)
AIC(fm.4, fm.5) # not an improvment
```

```
##      df      AIC
## fm.4 19 5364.218
## fm.5 20 5367.476
```

```
fm.6 = update(fm.4, .~. + mask.mand.flag)
AIC(fm.4, fm.6) # improvment
```

```
##      df      AIC
## fm.4 19 5364.218
## fm.6 20 5356.088
```

```
fm.7 = update(fm.6, .~. + mask.mand.days)
AIC(fm.6, fm.7) # not an improvement
```

```
##      df      AIC
## fm.6 20 5356.088
## fm.7 21 5366.180
```

That process added three additional variables: the binary flags for restaurant/bar closures, stay-at-home orders and mask mandates. Interestingly, AIC did not select any of the variables measuring how long those orders lasted. All three of these binary variables are highly correlated since states and counties that issued one type of mandate were likely to issue another. If we look at the below bootstrapped confidence intervals for the effects, one of them, the stay-at-home order, doesn't appear to be significant, and another, mask mandates, presents the unintuitive conclusion that mask mandates are associated with higher death rates. Ultimately, because the results of this conflict a bit with the results of the model comparisons using Kenward Rogers and parametric bootstrapping, I'd be quite hesitant to conclude that the mandates are influential. Doubly so since stepwise model selection didn't find the length of those mandates to be meaningful.

```
confint(fm.6, method='boot')
```

```
##                                2.5 %       97.5 %
## .sig01                 0.334299155   1.420323057
## .sig02                 0.000000000   0.534730201
## .sigma                 2.908430664   3.180773442
## (Intercept)          -33.385582776  29.527164955
## avg.temp              -0.118091316   0.065504874
```

```
## health.condition                   -0.028141934   0.119312379
## high.risk.job                      -0.065538852   0.017546552
## log(median.income)                 -1.526000402   4.019050021
## higher.income                      37.036027519 115.734184918
## log1p(black.pct)                   -0.008311066   0.742170655
## pop.density                         0.001450344   0.004809503
## log1p(hisp.latino.pct)              0.250590102   0.968073629
## pct.native.bin(30.1,60.1]          -0.144817709   4.859134593
## pct.native.bin(60.1,90.2]           5.055031570  10.842148554
## median.age                          0.033617494   0.148916859
## R.pres.margin                       0.038514227   0.067802517
## rest.closed.flag                   -7.371681354  -1.876445445
## stay.home.flag                     -2.778647094   0.198048552
## mask.mand.flag                      1.445399698   4.993802111
## log(median.income):higher.income -10.596741682  -3.373895016
```

Below are the bootstrapped confidence intervals for the effects in the model that only added partisan vote preference to the baseline model:

```
confint(m.pres, method='boot')
```

```
##                                        2.5 %        97.5 %
## .sig01                          0.925333840   2.366645341
## .sig02                          0.000000000   0.547092820
## .sigma                          2.926096470   3.170166763
## (Intercept)                   -36.464185205  30.073298037
## avg.temp                       -0.199988202   0.015280398
## health.condition               -0.030978702   0.120107572
## high.risk.job                  -0.069145255   0.018941824
## log(median.income)             -1.636522380   4.184822028
## higher.income                  34.450251348 113.481204649
## log1p(black.pct)               -0.020062694   0.695306217
## pop.density                     0.001359605   0.004885573
## log1p(hisp.latino.pct)          0.178454038   0.987592066
## pct.native.bin(30.1,60.1]      -0.050960912   5.164718755
## pct.native.bin(60.1,90.2]       4.825254843  10.971003629
## median.age                      0.038823182   0.149299602
## R.pres.margin                   0.036804567   0.067772850
## log(median.income):higher.income -10.369215295  -3.137848053
```

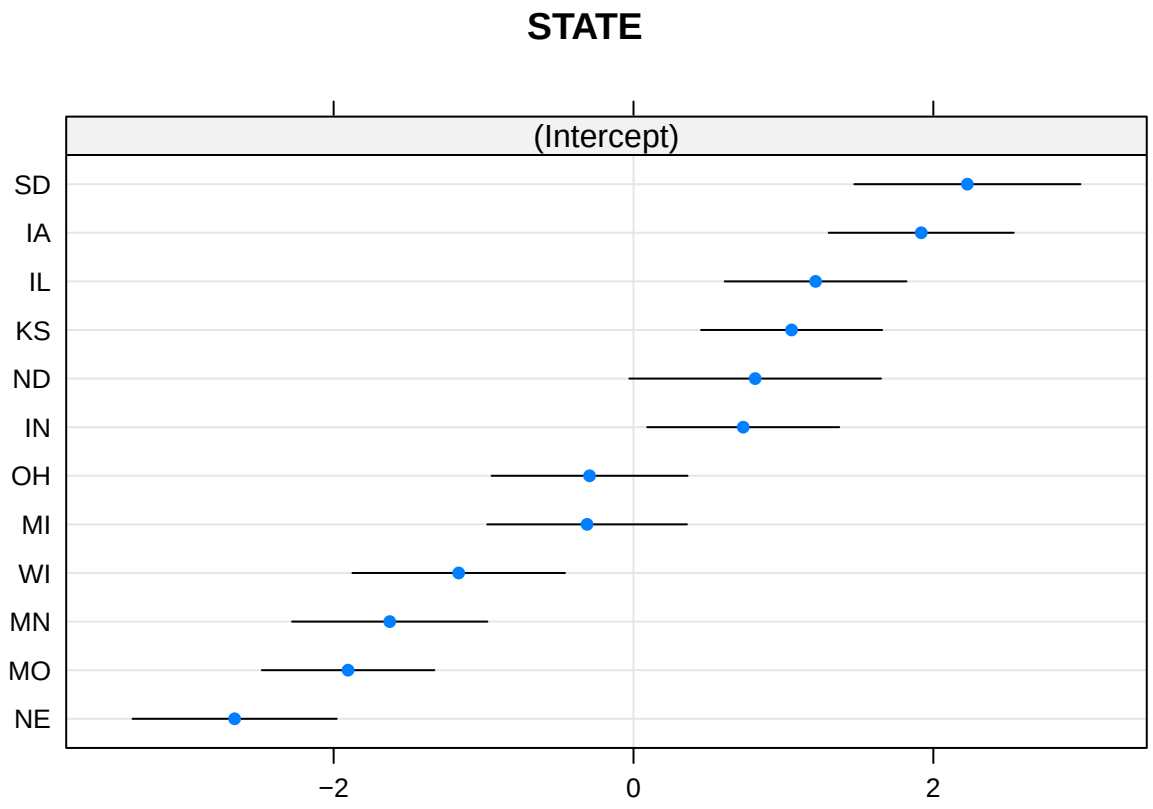
In the above output, the square root transformation of the response makes one-unit changes in the predictors challenging to explain, but the following predictors appear do appear to be significant:

- population density (positively associated with death rate)
- Hispanic/Latino share of the population (positively associated with death rate)
- Native American shares of county populations were binned, but the factors for the 2nd and 3rd levels (corresponding to 30-60% and 60-90%, respectively) were significant, and positively associated with death rate.
- Median age (positively associated with death rate).
- Republican presidential vote preference (positively associated whit death rate).
- The interaction between (log-transformed) median household income and a factor for "higher" income (meaning $57K and up) was negatively associated with death rate.

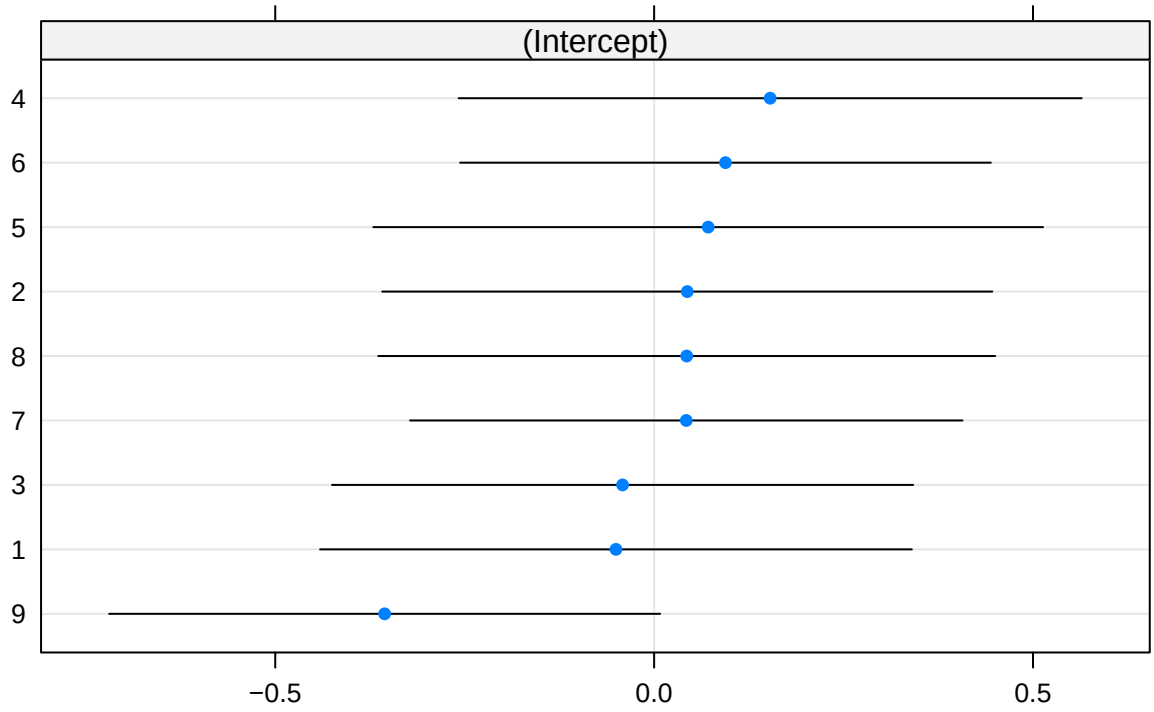
To take a closer look at the random effects, the confidence intervals for each state demonstrate that the state effect is still clearly significant. The effect for South Dakota, whose COVID outbreak was well-documented last year, last the largest positive effect of any state. And in a change from previously, the very smallest class, for rural areas with fewer than 2,500 people, appears to be influential.

```
dotplot(ranef(m.pres))
```

```
## $STATE
```




```
##
## $county.class
```




```
## If we repeat our tests for random effects that were initially
## done on the null model, state is still clearly significant,
## but county class is not (though it's p-value is almost borderline)
m.1.state.only  = update(m.pres, .~. - (1|county.class))
m.1.county.only  = update(m.pres, .~. - (1|STATE))
exactRLRT(m.1.state.only, m.pres, m.1.county.only)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 187.37, p-value < 2.2e-16
```

```
exactRLRT(m.1.county.only, m.pres, m.1.state.only)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 1.2455, p-value = 0.0948
```

Random Forest model

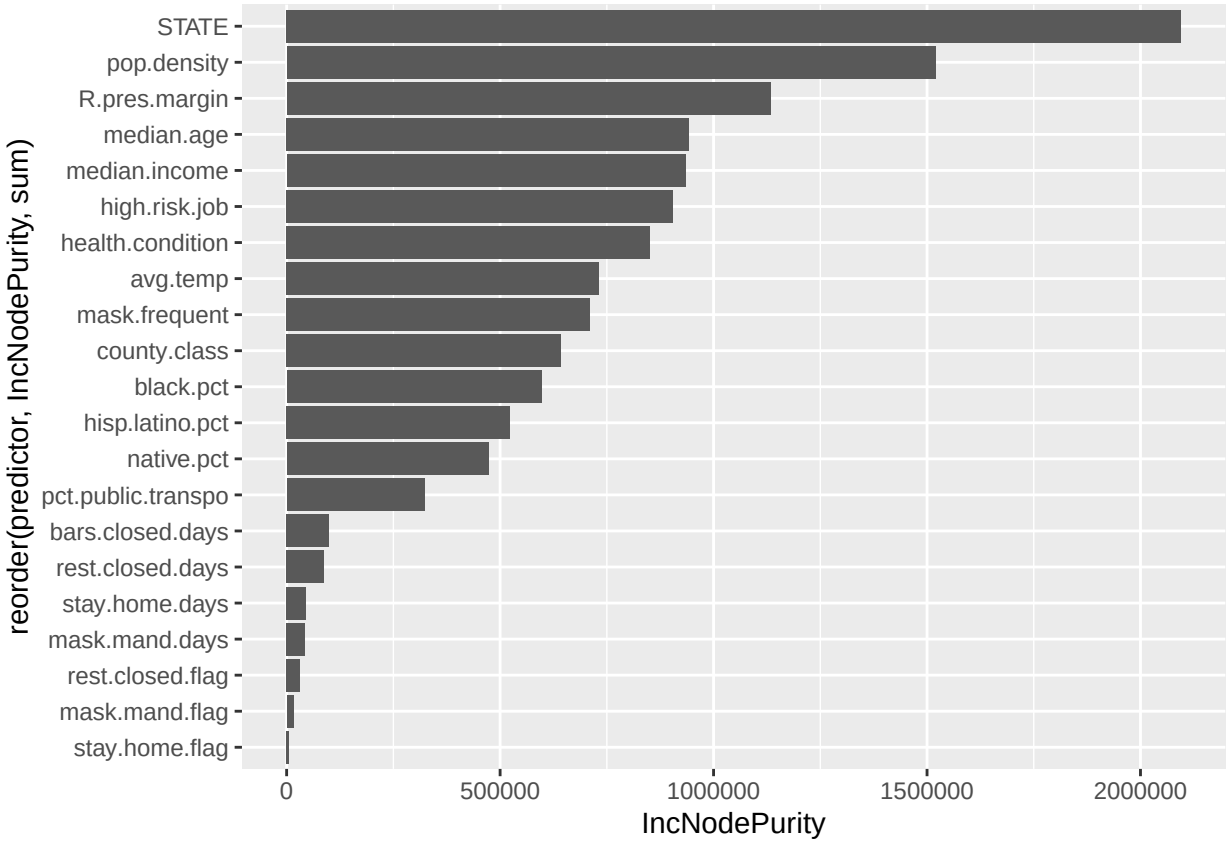
Another approach to modeling this problem that could help account for some of the non-linearity and discontinuities present in the variables is a random forest model. In this scenario, I'm more interested in understanding feature importance rather than being able to make accurate predictions on an unseen test data set. To get a sense of which variables might be influential in explaining COVID death rates, I'm first going to perform cross-validation to determine how many variables should be included in the sub-sample of predictors, and then fit a model on the entire dataset using the optimal parameter chosen by cross-validation

```
### removing unnneeded or artificial vars
tree.df = df %>%
  select(-c(COUNTY, higher.income, sqrt.deaths, population, pct.native.bin, deaths))

X1 = tree.df %>% select(-c(deaths.100k))
y1 = tree.df$deaths.100k
m.rf = rfcv(X1, y1, step = 0.9)
### look at the CV error for different levels of nvars
cbind(nvars=m.rf$n.var,MSE=m.rf$error.cv)
```

```
##    nvars       MSE
## 21    21  9389.958
## 19    19  9394.636
## 17    17  9574.024
## 15    15  9612.576
## 14    14  9552.709
## 12    12  9627.424
## 11    11  9804.998
## 10    10  9857.673
## 9      9  9727.144
## 8      8  9657.514
## 7      7  9722.581
## 6      6  9931.684
## 5      5 10227.815
## 4      4 10467.211
## 3      3 10255.114
## 2      2 10601.966
## 1      1 10561.761
```

NOTE (The number selected for the nvar parameter was not consistent each time this document was compiled (setting the seed worked in my local environment, but not with knitr). I have chosen 17 because it was the most frequent outcome in the handful of times I ran this.)

```
### apply the value with the lowest MSE to mtry and fit on full dataset
fmod = randomForest(X1, y1, mtry=17, importance=TRUE)
import = importance(fmod) %>%
  as.data.frame()
import = import %>% mutate(predictor=rownames(import))

ggplot(import, aes(y=reorder(predictor, IncNodePurity, sum), x=IncNodePurity)) +
  geom_col()
```



The random forest model found that state (which I'd prefer the random effect interpretation of since I'm not so much trying to estimate its effect as account for it's variance.), population density, presidential vote preference, median age, high-risk jobs and underlying health conditions were the most important features. Many of the mandate-related variables were found to be fairly unimportant.

If we re-run the above without state and county class (below), the feature importance changes a bit. While I prefer the random effects interpretation of those variables, with a tree-based model, I'd argue it's important they still be included as predictors and that it's not appropriate to exclude them.

```
X2 = tree.df %>% select(-c(deaths.100k, STATE, county.class))
y2 = tree.df$deaths.100k
## then split into train/test
m.rf2 = rfcv(X2, y2, step = 0.9)

### look at the CV error for different levels of nvars
cbind(nvars=m.rf2$n.var,MSE=m.rf2$error.cv)
```

```
##    nvars       MSE
## 19    19  9453.967
## 17    17  9525.617
## 15    15  9459.217
## 14    14  9605.257
## 12    12  9767.009
## 11    11  9844.966
## 10    10  9870.075
## 9      9  9779.949
```

```
## 8      8  9879.430
## 7      7  9912.273
## 6      6  9866.994
## 5      5 10050.818
## 4      4 10426.834
## 3      3 10849.417
## 2      2 11079.576
## 1      1 12317.460
```

```
### That recommends leaving 17 again for mtry
fmod2 = randomForest(X2, y2, mtry=17, importance=TRUE)
importance(fmod2) %>%
  as.data.frame() %>%
  arrange(desc(IncNodePurity))
```

```
##                     %IncMSE IncNodePurity
## pop.density        17.483447    1553641.51
## health.condition   18.304578    1259020.58
## R.pres.margin      18.043250    1210742.71
## median.income      12.588576     994830.39
## avg.temp           14.945199     905386.11
## high.risk.job       9.062946     899044.99
## mask.frequent       8.321727     846716.47
## median.age          3.396958     830747.01
## bars.closed.days   16.551985     645779.22
## hisp.latino.pct     3.133171     631372.63
## black.pct           8.582944     617120.80
## native.pct         10.647787     548339.27
## pct.public.transpo  5.413405     404887.29
## rest.closed.days   13.707035     391151.09
## mask.mand.days     15.038401     334500.57
## mask.mand.flag     13.162160     232020.66
## stay.home.days      9.940621     223844.52
## rest.closed.flag    8.614966     182850.05
## stay.home.flag      5.471184      29637.33
```

About the data

(Hover over descriptions for links to sources.)

The following county-level datasets were dowloaded from the U.S Census Bureau County-level statistics containing population, income, age, occupation breakdowns, were downloaded from the U.S. Census Bureau:

- Population
- Land area (used to calculated population density)
- Hispanic or Latino population
- Black population
- Native American population
- Median household income
- Median age
- Share of residents riding public transporation
- Share of residents working in high-risk occupations (defined as the below categories, which were highlighted by this UC-San Francisco study of California excess deaths)
  - Food preparation and serving related occupations
  - Production, transportation, and material moving occupations
  - Construction and extraction
  - Installation, maintenane and repair occupations

County classification codes (AKA Rural-Urban Continuum Codes), were downloaded from https://www.ers.usda.gov/data-products/rural-urban-continuum-codes.aspx

The self-relported share of residents who "always" or "frequently" wear a mask was made available by the New York Times.

Covid death statistics were downloaded from the New York Times.

County-lelel election results were scraped from the unofficial NYT elections API using this scraper.

County-level statistics on residents' underlying health conditions came from the CDC

Average temperature data from April 2020-March 2021 came from NOAA

Data for the following statistics was downloaded from the CDC:

- Stay at home orders
- Mask mandates
- Restaraunt closures
- Bar closures