CLM Assumptions

```
library(tidyverse)
library(stargazer)
data <- read_rds("../data/processed/main_state_data.RDS")

## All below done in Python Now ##
##################################
# Colnames: replace spaces with underscore,
# remove parantheses, metacharacters () must be enclosed in []
#colnames(data) <- gsub(" ", "_", colnames(data))
#colnames(data) <- gsub("[()]", "", colnames(data))

head(data)
```

```
lm1 <- lm(Case.Rate.per.100000.in.Last.7.Days ~ SIP, data = data)

lm2 <- lm(Case.Rate.per.100000.in.Last.7.Days ~ SIP + workplaces_2020.10.10, data = data)

lm3 <- lm(Case.Rate.per.100000.in.Last.7.Days ~ SIP + workplaces_2020.10.10 + NoFaceMask, data = data)

lm4 <- lm(Case.Rate.per.100000.in.Last.7.Days ~ SIP + workplaces_2020.10.10 + NoFaceMask + NoFaceMaskEm

summary(lm1)
```

```
##
## Call:
## lm(formula = Case.Rate.per.100000.in.Last.7.Days ~ SIP, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.527 -12.943  -5.127  10.558  70.173
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   52.827      6.403   8.250 7.93e-11 ***
## SIP          -27.685      7.230  -3.829 0.000366 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.24 on 49 degrees of freedom
## Multiple R-squared:  0.2303, Adjusted R-squared:  0.2146
## F-statistic: 14.66 on 1 and 49 DF,  p-value: 0.0003657
```

```
summary(lm2)
```

```
##
```

```
## Call:
## lm(formula = Case.Rate.per.100000.in.Last.7.Days ~ SIP + workplaces_2020.10.10,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.363 -11.148  -5.250   7.967  69.974
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            72.1468    11.4487   6.302 8.66e-08 ***
## SIP                   -24.4592     7.1972  -3.398  0.00137 **
## workplaces_2020.10.10   1.0499     0.5225   2.009  0.05017 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.61 on 48 degrees of freedom
## Multiple R-squared:   0.29,  Adjusted R-squared:  0.2604
## F-statistic: 9.804 on 2 and 48 DF,  p-value: 0.0002691
```

```
summary(lm3)
```

```
##
## Call:
## lm(formula = Case.Rate.per.100000.in.Last.7.Days ~ SIP + workplaces_2020.10.10 +
##     NoFaceMask, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.964 -13.331  -3.882   9.249  65.914
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            65.7892    12.1668   5.407  2.1e-06 ***
## SIP                   -22.2795     7.2813  -3.060  0.00365 **
## workplaces_2020.10.10   0.9701     0.5199   1.866  0.06832 .
## NoFaceMask              8.9653     6.2678   1.430  0.15922
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.39 on 47 degrees of freedom
## Multiple R-squared:  0.3196, Adjusted R-squared:  0.2762
## F-statistic:  7.36 on 3 and 47 DF,  p-value: 0.000384
```

```
summary(lm4)
```

```
##
## Call:
## lm(formula = Case.Rate.per.100000.in.Last.7.Days ~ SIP + workplaces_2020.10.10 +
##     NoFaceMask + NoFaceMaskEmploy, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -31.579 -11.589  -4.967   8.802  69.586
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            63.6783    12.4766   5.104  6.2e-06 ***
## SIP                   -21.5180     7.3652  -2.922  0.00538 **
## workplaces_2020.10.10   0.9019     0.5283   1.707  0.09453 .
## NoFaceMask              6.1619     7.1556   0.861  0.39363
## NoFaceMaskEmploy        8.7451    10.6449   0.822  0.41558
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.46 on 46 degrees of freedom
## Multiple R-squared:  0.3295, Adjusted R-squared:  0.2712
## F-statistic: 5.651 on 4 and 46 DF,  p-value: 0.0008728
```

The 5 CLM Assumptions that we want to evaluate: * IID Data * No perfect collinearity * Linear Conditional Expectation * Homoskedastic errors * Normally distributed errors

Let's evaluate these assumptions for the State COVID Policy dataset.

1. IID Data Each state has their own governors/policymakers, policies, and policy timings which may lead us to think that each state is independent. However, the states are adjacent to each other which violates the independence assumption. States that are closer to each other (like Georgia and Alabama) are more likely to be more similar to each other as compared to states that are further (like Georgia and California). Additionally, each state's population density is different, which means that some states are more likely to have higher COVID case rates and different policies as compared to others. We do not have independent or identically distributed state COVID policy data so this assumption is not met. We will want to be careful as our results may not be consistent and we may have larger errors because of our non-IID data. We will definitely keep this in mind when looking at our analysis and determining the practical significance.
2. No perfect collinearity

```
summary(lm3)
```

```
##
## Call:
## lm(formula = Case.Rate.per.100000.in.Last.7.Days ~ SIP + workplaces_2020.10.10 +
##     NoFaceMask, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.964 -13.331  -3.882   9.249  65.914
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            65.7892    12.1668   5.407  2.1e-06 ***
## SIP                   -22.2795     7.2813  -3.060  0.00365 **
## workplaces_2020.10.10   0.9701     0.5199   1.866  0.06832 .
## NoFaceMask              8.9653     6.2678   1.430  0.15922
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 20.39 on 47 degrees of freedom
## Multiple R-squared:  0.3196, Adjusted R-squared:  0.2762
## F-statistic:  7.36 on 3 and 47 DF,  p-value: 0.000384
```

```
cor(data$SIP, data$workplaces_2020.10.10)
```

```
## [1] -0.2230676
```

```
cor(data$SIP, data$NoFaceMask)
```

```
## [1] -0.23597
```

```
cor(data$NoFaceMask, data$workplaces_2020.10.10)
```

```
## [1] 0.1542048
```

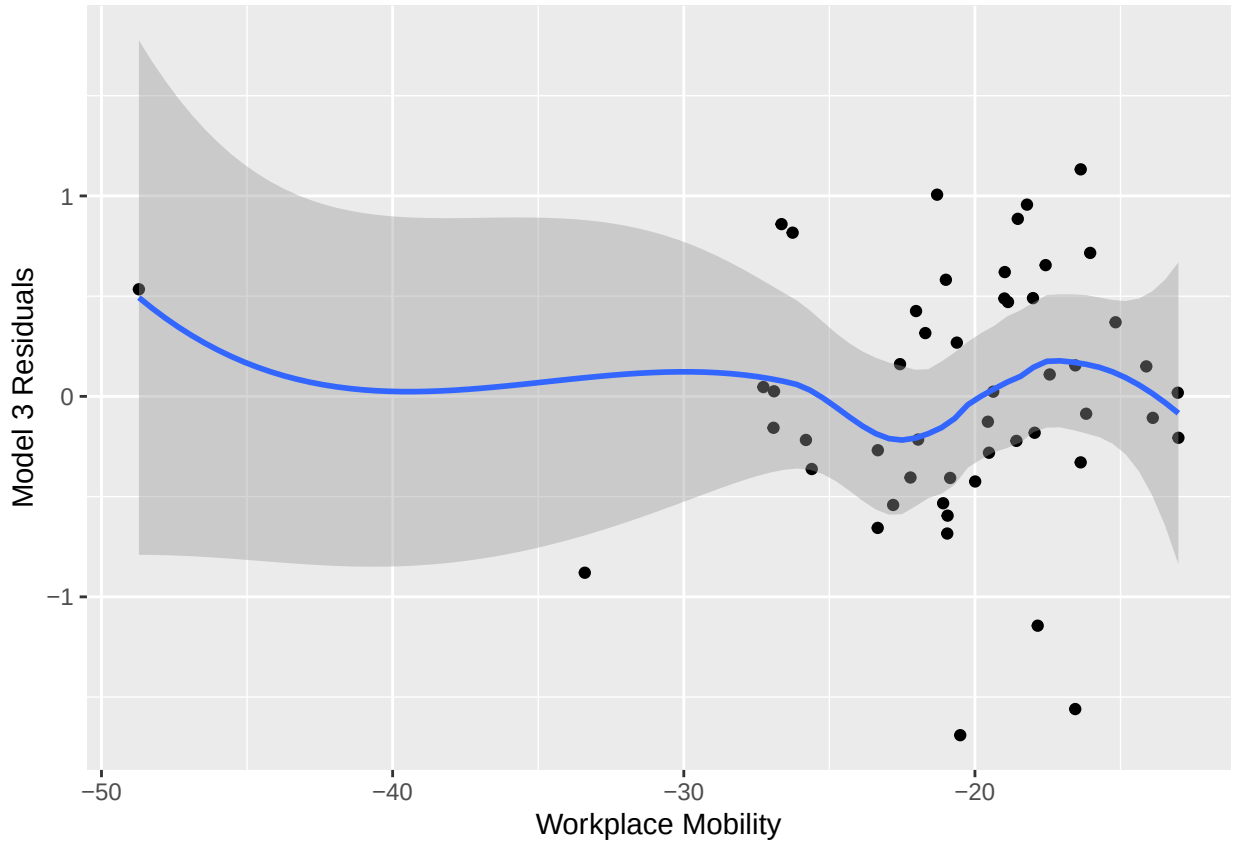
When looking at our main model, we want to check that the regressors are not collinear or near collinear. If we had near perfect collinearity, we would have large standard errors on collinear features. Looking at our standard errors, none of them are huge, so we lean towards the idea of not having near perfect collinearity. We can also check the correlation between each of the regressors. We get values around 0.25 and -0.25, which aren't too high. If we had near perfect collinearity between different variables, the correlation between them would be much higher. We do not see this so we can say that the no perfect collinearity condition is met - we will not have to drop any of our variables If our data was perfectly collinear, the variables would not have a unique solution and we would not be able to generate estimates of the regression coefficients without dropping some of the variables.

3. Linear Conditional Expectation

```
lm3 <- lm(log(Case.Rate.per.100000.in.Last.7.Days) ~ SIP + workplaces_2020.10.10 + NoFaceMask, data = da
model_3_predictions <- predict(lm3)
model_3_residuals <- resid(lm3)


plot_model_3a <- data %>%
  ggplot(aes(x = SIP, y = model_3_residuals)) +
  geom_point() + stat_smooth()

plot_model_3b <- data %>%
  ggplot(aes(x = workplaces_2020.10.10, y = model_3_residuals)) +
  geom_point() + stat_smooth() +
  xlab('Workplace Mobility') +
  ylab("Model 3 Residuals")

plot_model_3c <- data %>%
  ggplot(aes(x = NoFaceMask, y = model_3_residuals)) +
  geom_point() + stat_smooth()

plot_model_3d <- data %>%
  ggplot(aes(x = model_3_predictions, y = model_3_residuals)) +
  geom_point() + stat_smooth() +
  xlab('Model 3 Predictions') +
```

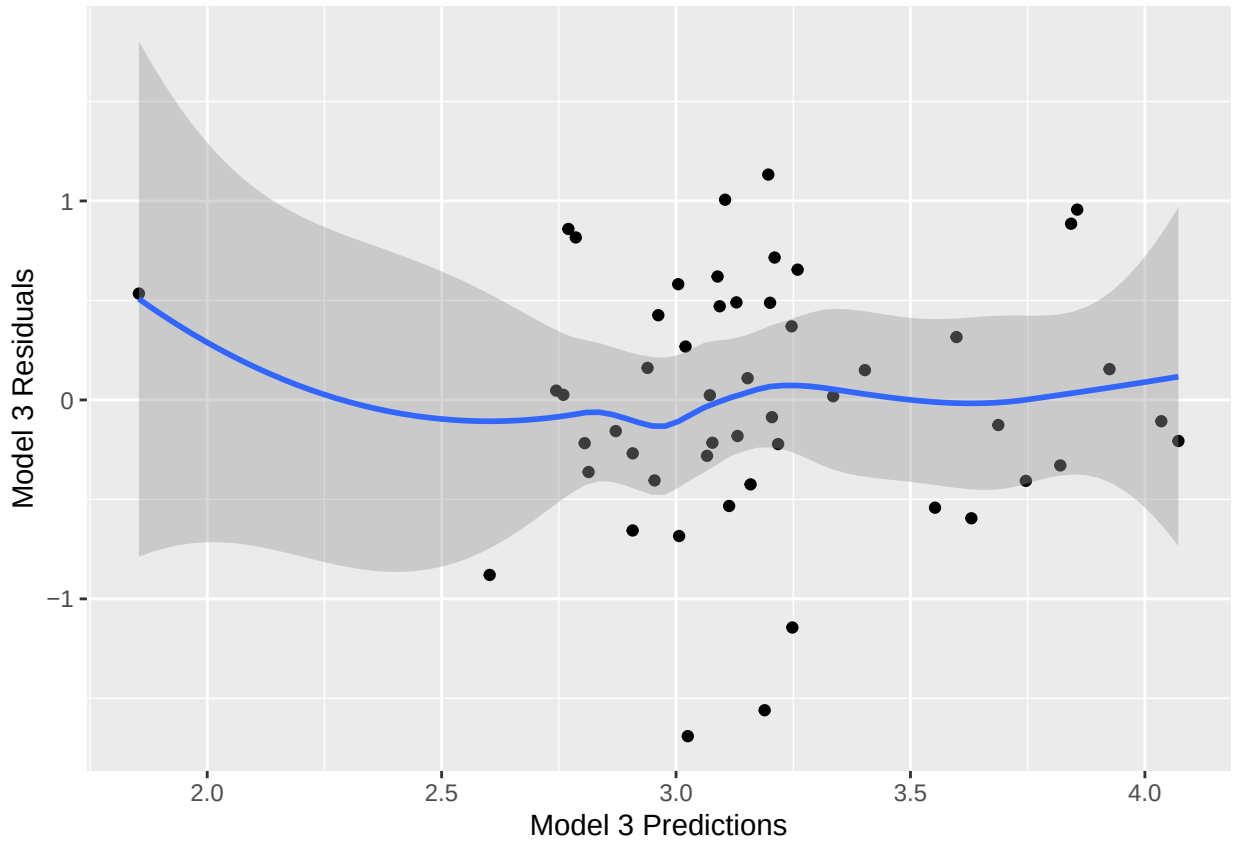
```
  ylab('Model 3 Residuals')

#plot_model_3a
plot_model_3b
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
#plot_model_3c
plot_model_3d
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggsave(plot = plot_model_3b, filename = "../reports/figures/LCE_mob_resid.png")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
ggsave(plot = plot_model_3b, filename = "../reports/figures/LCE_pred_resid.png")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

Ideally if this condition is satisfied, we will see a linear relationship between the predictions and residuals, represented through the plots. Two of our variables (SIP and NoFaceMask) are indicator variables so the relationship described related to these two does not make sense (there won't be a good linear relationship from these variables). When looking at the first plot above (workplace mobility), we see a non-linear relationship - the plot curves around -25 to -15 values of mobility. Looking at the overall model predictions, the relationship is curved as well. We have a few areas where it is linear but the blue line looks more cubic. As a result, this assumption is not met and we do not have linear conditional expectation. The model that we have fitted assumes that the data is linear but the estimated coefficient does not match the relationship in the data. It is possible that a linear relationship/model is not the best way to represent the data. In the future when we model with this data, we may want to consider transforming the data so there is a linear conditional expectation.

4. Homoskedastic errors

To evaluate homoskedastic errors, we will conduct the Ocular test (to look for fanning out of the data acros the predicted values) and the Breusch-Pagan test (a statistical test with a p-value to evaluate the errors).

```
# Ocular Test
# use the same plot from lce
#ggplot(aes(x = predict(lm3), y = resid(lm3)), data = data) +
  #geom_point() +
  #xlab('Model 3 Predictions')


#plot(lm3, which=3)
```

```
# Breusch-Pagan test
lmtest::bptest(lm3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm3
## BP = 0.73457, df = 3, p-value = 0.865
```

When data has different conditional variance we say that it is heteroskedastic. We will conduct the Ocular/Eye test by plotting the predictions versus the residuals to see the errors and conditional variance, thus we can inspect the second plot in We want to check if this data fans out or not. Based on the plot above, our data does seem to fan out. As we move from left to right on the plot, the spread of the points increases. We also don't see the errors evenly distributed. Instead there are a lot of points in the center area of the plot and fewer points as we get to the sides of the plot. When we conduct the Breusch-Pagan test, we get a p-value of 0.02974. This is a small p-value (less than 0.05 which is our typical cutoff), so we can reject the null hypothesis that the error variances are all equal. Both of these results point towards the same result: that our data does not have homoskedastic errors, we have heteroskedastic errors. Since the data is heteroskedastic, the error variance is different for the different parts of the x range. We will have bias in the standard errors of our model and our model may miss things in certain areas because it cannot detect the bias effect. We may get a p-value that is smaller than it should be. As we move forward with this data and our model, we should consider the effect of heteroskedastic errors. To account for this, we will proceed with using robust standard errors.

5. Normally distributed errors

```
norm_error1 <- ggplot(data = data) +
  geom_histogram(mapping = aes(x = resid(lm3))) +
  ggtitle("Distribution of Residuals for Model 3") +
  xlab('Model Residuals') +
  ylab('Count')


norm_error2 <- ggplot(data = data, aes(sample = resid(lm3))) + stat_qq() + stat_qq_line()

ggsave(plot = norm_error1, filename = "../reports/figures/norm_error1.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
ggsave(plot = norm_error2, filename = "../reports/figures/norm_error2.png")
```

```
## Saving 6.5 x 4.5 in image
```

To evaluate this assumption, we will want to plot a histogram of the errors and see what the distribution looks like. When plotting our residuals for this model, we get a distribution that is slightly skewed (with a right tail) but is tending towards a normal distribution. As an extra check, we can also plot a qqplot to see how normal the residuals are. We can see the points and the line. The points are not linear and do not follow the given line on either end of the data. As a result, we may want to be careful because this assumption isn't perfectly satisfied. If our random errors are not from a normal distribution, our model can make incorrect decisions more or less frequently than what our inferences show. We will want to keep this assumption in mind, especially when determining the practical significance of our model results.