Ceaseless Cumulative COVID Cases

Executive Summary

We are interested in the relationships between the cumulative number of COVID cases for each zip code in San Diego and demographic information such as race, sex, etc. Our goal is to create a model to estimate the number of cases in San Diego County, based on a few independent variables. A linear regression model is created to explore this relationship since we have numeric data. Four variables were significant in estimating the square root of the cumulative number of cases; the number of Blacks, total Hispanics, total Hispanics^2, and non-Hispanic whites variables captured much of the variation in COVID cases. All hypothesis tests in this analysis will be conducted using a significance level of 0.05.

Introduction

Our primary interest is seeing which variables have the highest correlation with COVID case counts, with the goal to create a regression model for estimation. There are a total of 16 variables containing demographic information for 102 zip codes for San Diego residents, along with another file that has 113 cumulative case counts per day from 4/1/2020 to 6/29/2021. For our analysis, we merged these data sets together. Unfortunately, this means we have no demographic information for the 12 zip codes that are not in both files. With this merge, there are now 102 observations in our data set. Using this new combined data set, we are able to look into our question of interest: How are cumulative COVID case counts related to demographic information? We had the following hypotheses to address the relationship between our explanatory variables and our response variable: H_0: There is no linear relationship between any explanatory variables and our response variable vs. H_A: At least one explanatory variable has a linear relationship with the cumulative number of cases in a county.

Exploratory Data Analysis

We want to see the distribution of our intended response variable, cumulative COIVD-19 cases, in order to understand what we want to estimate. Hence, we visualized it with a histogram:

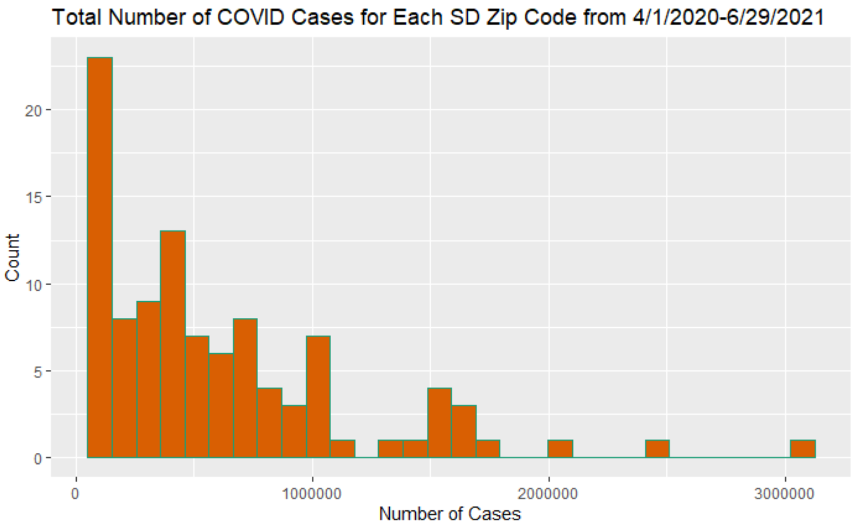

Figure 1: Distribution of Cases for Zip Codes in San Diego County

We notice the data are heavily right-skewed, meaning few of our observations, which are zip code-level data, have over 100k cumulative cases. Hence, we elect to transform it into something that has a less skewed distribution. Rather than using the response variable as provided, we will be using the square root of it instead, as depicted here, in our analysis:

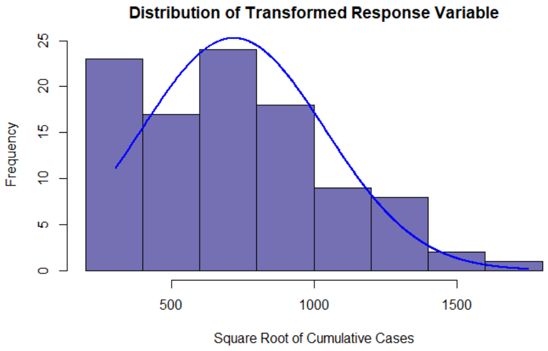

Figure 2: Distribution of Transformed Response

Although there is still some right skew, it is approximately normal and should work better as a response variable due to the reduction in skew. Now that we have transformed our response variable, we want to see how it is related with demographic data, which will serve as our explanatory variables. Further, no outliers are detected in our response variable, so we do not make any other changes to it. Below, we visualize the relationships for our data, in pairs. Note that we dropped the women variable since it is very redundant with the men variable, thus making it inappropriate to keep as part of our data.

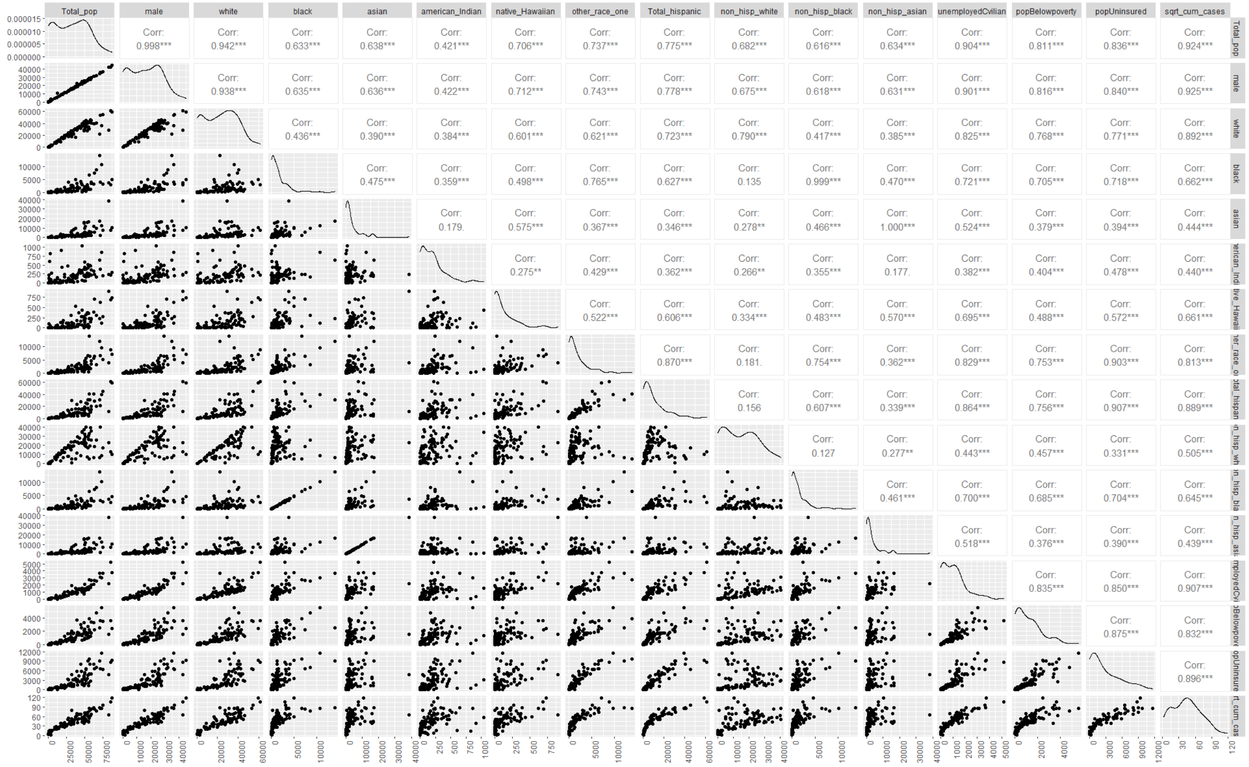

Figure 3: Pairs Plots for Demographics & Cumulative Cases Data

We are particularly interested in the relationships that the square root of cumulative cases (sqrt_cum_cases) has with all the explanatory variables. We notice some correlation values with sqrt_cum_cases are +0.8 or higher, indicating a strong positive relationship between those variables. Although these seem like strong explanatory variables, some explanatory variables have high correlations with each other, which would lead to a model suffering from multicollinearity. This is an issue since it essentially means some information in some variables is repeated in others; information is redundant. As a result, we must be wary about how variables relate not only to our dependent variable, but also with each other. Furthermore, some relationships with the square root of cumulative cases are linear, while others like race are quadratic.

As part of our study, we decided to check the difference between case counts per day, regardless of zip code, defined as current date minus yesterday's date. In total, we have 451 observations left after losing the first date using this grouped daily data.

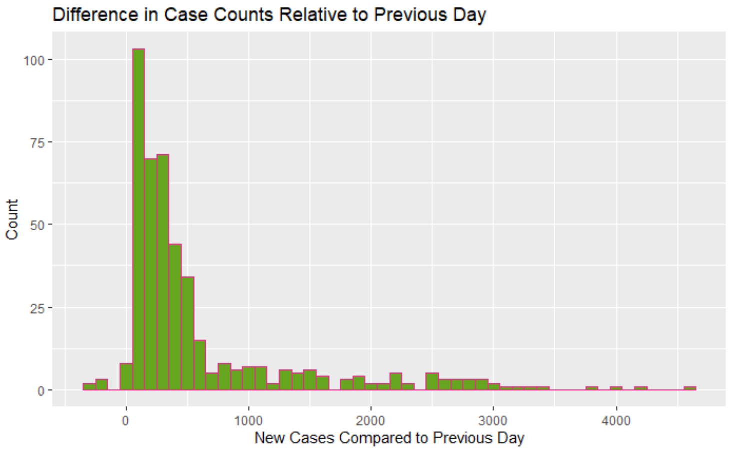

Figure 4: Change in Cases Per Day for all San Diego Zip Codes

Interestingly, there are a few days that have fewer cases than the previous day, suggesting an error in the data entry process. Ideally, we would look into why, but we do not have access to such resources, so we note that our inference will not be perfect due to a collection mistake(s).

Statistical Analyses

Model Assumptions & Diagnostics

We have developed a regression model since we have a continuous response variable with numeric independent variables, and observed linear and quadratic relationships with our response variable in our pairs plot. Linear regression has assumptions we have to verify are met for reliable inference. Namely, that there is a polynomial relationship between the dependent and independent variables, constant variance for all observations, independent observations, and normally distributed errors resulting from the model. Since some variables appear quadratic, we begin by narrowing down candidates for independent variables and will see how they with the model.

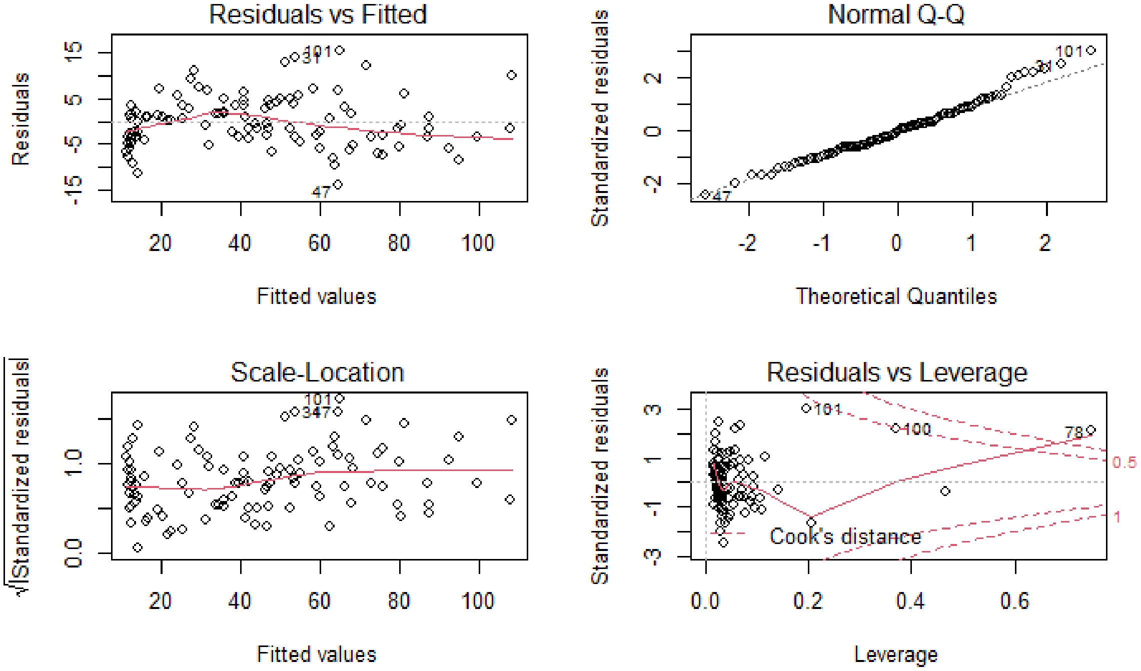

Figure 5: Regression Model Diagnostics

The lack of a clear pattern for a majority of the residuals vs. fitted plot suggests our errors follow a normal distribution, and we see the errors are centered around 0, suggesting the data are independent and identically distributed. Next, the Q-Q plot following shows our normal assumption is met. Since the scale location is mostly a line, it suggests our constant variance plot is met. Lastly, the final plot indicates two observations that are influential in the model, which are contained in the 78th and 100th rows of our data set. These observations contain data for the 92114 and 92154 zip codes, respectfully, which have a much lower value for the square root of cumulative cases than the model estimated, making them influential in our model. Overall, our model assumptions are met, allowing us to proceed with statistical inference.

Regression Model

We were able to create an additive regression model using only four variables. Originally, we started with a loaded model that contained all our explanatory variables. However, we wanted a model that has fewer terms to make it easier to understand, which would reduce the redundant information our exploratory data analysis found due to high multicollinearity with the data set. After trying to perform stepwise regression and noticing a multicollinearity problem with the model it generated, we chose to recursively remove variables that had high variable inflation factors (VIFs) to create a model with less repetitive information. A cutoff of VIF > 4 is used for our analysis. Once this recursion is finished, we take out the insignificant variables, leaving us with number of Black, total Hispanic, and non-Hispanic white as our explanatory variables for the square root of cumulative coronavirus case counts. Since we observed a quadratic relationship between Black and total Hispanic from our pairs plot, we consider a model that includes the first and second degree versions for those variables.

We must assess if adding the quadratic terms to our model was worth it. After all, our goal is to have a model as simple and informative as possible. Hence, we must look for significance of our explanatory variables. When we had a model that included quadratic terms for both the total Hispanic and Black variables, we found that Black^2 had a p-value of 0.055, making it insignificant at our significance level of 0.05 since the p-value is greater than our significance level. On the other hand, all other variables have p-values below 0.05, making them significant. As a result, this suggests dropping the Black^2 term. Another way we assessed if keeping Black^2 in our model was worth it was to do an ANOVA test comparing the model with Black^2 in it with a model without Black^2, while keeping the other aforementioned explanatory variables as part of these models. This test's hypotheses are that the simpler model is preferred as the null, versus the more complicated model is preferred as the alternative. In other words, adding the Black^2 term has to provide a significant improvement to our model to have a significant p-value. Conducting the ANOVA test comparing these two models yielded a p-value of 0.0556, which supports the claim that the simpler model is preferred. Hence, we use a model containing Black, total Hispanic, total Hispanic^2, and non-Hispanic white as our best regression model. Our regression results follow below.

Variable	Estimate	Standard Error	Test Statistic, t	P-value
Intercept	11.2819	1.0442	10.804	< 0.001
Black	0.0012	0.0003	3.392	0.001
Total Hispanic	0.0022	0.0002	14.183	< 0.001
$(\text{Total Hispanic})^2$	~0	~0	-4.888	< 0.001
Non-Hispanic White	0.0007	0.0001	12.64	< 0.01

Our model has the following form: $\{\hat{Y}\}$ = 11.2819 + 0.0012*Black + 0.0022*Total Hispanic + ~0*$(\text{Total Hispanic})^2$ + 0.0007*Non-Hispanic White, where $\{\hat{Y}\}$ is the estimated square root of cumulative cases. Since all coefficients are positive, an increase in any of our independent variables means the estimated square root of cases will increase. If we add one Hispanic person, we can use the regression coefficient to interpret how the response would be affected. For one additional Hispanic person, we would expect the square root of cumulative cases to increase by 0.0022, holding other variables constant. Additionally, we have an adjusted R^2 value of 95.13%, which means 95.13% of variation in the square root of cumulative cases can be explained by our four explanatory variables; almost all variation in our response is captured with only four variables, making our model is successful for estimation.

Due to having two influential points, we created another model with those observations removed to see how the model would change. Our diagnostics looked fairly similar, with the regression coefficients changing slightly. Overall, the influential points have an impact, but removing them does not make any significant changes.

Variable	Estimate	Standard Error	Test Statistic, t	P-value
Intercept	11.1488	1.0240	10.887	< 0.001
Black	0.0011	0.0042	2.729	0.0076
Total Hispanic	0.0024	0.0002	14.342	< 0.001
(Total Hispanic)^2	~0	~0	-5.54	< 0.001
Non-Hispanic White	0.0007	0.0001	12.369	< 0.01

Conclusions

Our study successfully created a model to estimate the number of the square root of COVID-19 cases using four variables containing demographic information for San Diego County residents. Three of these variables are first order terms, while one of them needed a second order term to capture the quadratic relationship between itself and the response variable.

However, our study did have some limitations. Since the demographic information data set had less zip codes than the case counts data set, we lost data when performing our merge. Ideally, we would get the demographic information from these zip codes that had to be dropped as well - either through another data set found online, or by sending researchers to collect these data. This would provide more observations to work with, leading to a more accurate data set to work with. Another issue was that even though the counts should be cumulative, we still noticed that when all zip codes were aggregated by day, there were some days that had negative case counts. Since case counts can only remain stable or go up, it indicates an issue in the data collection process has occurred, and not all data are reliable.

Other studies can be done using our data sets, especially for the case counts file. Since we have information on a daily basis, we can create a time series model using dependencies from previous days to estimate how the case count changes based on our demographic data. This analysis can be done using aggregated data for all San Diego zip codes like in Figure 4, or can be done at the zip code level instead.

References

ggpairs Rotation: https://stackoverflow.com/questions/46864196/ggpairs-rotate-axis-label
Quadratic Model: https://datascienceplus.com/fitting-polynomial-regression-r/
Diagnostics & Leverage:
https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression7.html
https://data.library.virginia.edu/diagnostic-plots/
Reshaping Data from Wide to Long Format for COVID Data:
https://stackoverflow.com/questions/2185252/reshaping-data-frame-from-wide-to-long-format
Renaming Variables:
https://www.sharpsightlabs.com/blog/rename-columns-in-r/

Alice, Michy. "Fitting Polynomial Regression in R." *DataScience+*, 10 Sept. 2015, https://datascienceplus.com/fitting-polynomial-regression-r/.

Bommae, Written by. "University of Virginia Library Research Data Services + Sciences." *Research Data Services + Sciences*, 21 Sept. 2015, https://data.library.virginia.edu/diagnostic-plots/.

Ebner, Joshua. "How to Rename Columns in R." *Sharp Sight*, 24 July 2021, https://www.sharpsightlabs.com/blog/rename-columns-in-r/.

Jaap. "Reshaping Data.frame from Wide to Long Format." *Stack Overflow*, 15 Sept. 2014, https://stackoverflow.com/questions/2185252/reshaping-data-frame-from-wide-to-long-format.

Public School of Health, Boston University. "Regression Diagnostics." *Regression Diagnostics*, 6 Jan. 2016, https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression7.html.

S J Cowtan. "GGPAIRS Rotate Axis Label." *Stack Overflow*, 21 Oct. 2017, https://stackoverflow.com/questions/46864196/ggpairs-rotate-axis-label.

Appendix

~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r Libraries, message=FALSE}
library(tidyverse)
library(lubridate)
library(data.table)
library(RColorBrewer)
library(corrplot)
library(betareg)
library(leaps)
library(MASS)
library(FactoMineR)
library(sandwich)
library(msm)
```

```{r Formatting}
# color palette for data viz
color_pal <- RColorBrewer::brewer.pal(n = 5, name = "Dark2")
# change scientific notation to standard form
options(scipen = 100)
```



```{r Load in Data}
# change read.csv to read_csv to fix the data reading in issue for dates as columns
covid <- read_csv("C:/Users/razmi/OneDrive/Desktop/Data Analysis Exam
SDSU/SD_Zipcode_COVID_4_DAE_F22_f.csv")
demographics <- read_csv("C:/Users/razmi/OneDrive/Desktop/Data Analysis Exam
SDSU/demographic_SD_ZIP_4_DAE_F22.csv")

# notice: one data set has more rows than the other => some zipcodes are not going to be in both
# potential issue!
dim(covid)
dim(demographics)
~~~

~~~
# change capitalization to allow for merge
# https://www.sharpsightlabs.com/blog/rename-columns-in-r/
covid <- rename(covid, zipcode = Zipcode)

# check for NAs - none in either data set
# anyNA(covid_total_zip)
# anyNA(demographics)

# check if any zipcodes got doubled - no duplicates!
# length(unique(covid_total_zip$zipcode)) == length(covid_total_zip$zipcode)
# length(unique(demographics$zipcode)) == length(demographics$zipcode)

# get total cases for each zip code
covid_drop_zip <- covid %>%
  dplyr::select(-zipcode)
covid_total_zip <- covid %>%
  rowSums()
covid_total_zip <- data.frame(zipcode = covid$zipcode, case_count = covid_total_zip)

# combine datasets
df <- demographics %>%
  left_join(covid_total_zip, by = "zipcode")

# https://stackoverflow.com/questions/2185252/reshaping-data-frame-from-wide-to-long-format
# change wide to long format for better usage
covid_total_zip_long <- melt(setDT(covid_total_zip), id.vars = c("zipcode"), variable.name = 
"Date")
covid_total_zip_long <- rename(covid_total_zip_long, cases = value)
# summary(covid_total_zip_long)

# change zipcode to factor
covid_total_zip_long$zipcode <- as.factor(covid_total_zip_long$zipcode)

# NOTE: CHANGE ROWSUM - THIS IS CUMULATIVE DATA
```


```{r}
# redo df into long
# df_long <- melt(setDT(df), id.vars = c("zipcode"), variable.name = "Date")
df_long <- reshape2::melt(df, id.vars = 1:18, variable.name = "Date")
~~~

~~~~
# plot case count
ggplot(df_long, aes(case_count)) +
  geom_histogram(col = color_pal[1], fill = color_pal[2]) +
  labs(x = "Number of Cases", y = "Count", title = "Total Number of COVID Cases for Each SD
Zipcode from 4/1/2020-6/29/2021")

ggplot(df_long, aes(sqrt(case_count))) +
  geom_histogram(col = color_pal[3], fill = color_pal[4]) +
  labs(x = "Number of Cases in Zipcode", y = "Count", title = "sqrt(Total Number of COVID
Cases for Each SD Zipcode from 4/1/2020-6/29/2021)")
```



```{r Transformation of y}
rcompanion::plotNormalHistogram(sqrt(df_long$case_count), col = color_pal[3],
                                xlab = "Square Root of Cumulative Cases",
                                main = "Distribution of Transformed Response Variable")
                                # ,
                                # breaks = 30)
```



```{r Model Info & Testing, message=FALSE, fig.width=20, fig.length=20}
# look only at cumulative cases - last column
covid_cumulative <- data.frame(zipcode = covid[, 1], cumulative_cases = covid[, ncol(covid)])
%>%
  rename(zipcode = zipcode, cumulative_cases = X6.29.2021)

# merge data
full_df <- demographics %>%
  left_join(covid_cumulative, by = "zipcode")
mod <- lm(sqrt(cumulative_cases) ~ . - zipcode, data = full_df)
summary(mod)
plot(mod)

# needed to drop female since it's a function of male - otherwise error
full_df_no_women <- full_df %>%
  dplyr::select(-female)
~~~~

```
# look at rv - sqrt was best transformation on y
rcompanion::plotNormalHistogram(sqrt(full_df_no_women$cumulative_cases))

my_lm <- lm(sqrt(cumulative_cases) ~ . - zipcode, data = full_df_no_women)
summary(my_lm)
# check correlations wuth cumulative cases
cor_x_y <- cor(full_df_no_women[,2:16], full_df_no_women[,17])
corrplot(cor_x_y)

# view all correlations at once - multicollinearity?
correlations <- cor(full_df_no_women)
corrplot(correlations, col=colorRampPalette(c("red","white","green"))(200))

# redo df with rv = sqrt(y) going forward
full_df_no_women_sqrt_y <- full_df_no_women %>%
  mutate(sqrt_cum_cases = sqrt(cumulative_cases)) %>%
  dplyr::select(-cumulative_cases)

# pairwise plots to show all corrs
pairs_df <- full_df_no_women_sqrt_y %>%
  dplyr::select(-zipcode)
GGally::ggpairs(pairs_df) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

predictors_df <- full_df_no_women_sqrt_y %>%
  dplyr::select(-sqrt_cum_cases, -zipcode)

# look for multicollinearity in model - drop zipcode since it doesnt help
df_for_lm <- full_df_no_women_sqrt_y %>%
  dplyr::select(-zipcode)
lmMod <- lm(sqrt_cum_cases ~ . , data = df_for_lm)
selectedMod <- step(lmMod)
summary(selectedMod)

# check multicollinearity
all_vifs <- car::vif(selectedMod)
print(all_vifs)

signif_all <- names(all_vifs)
```

```
# Remove vars with VIF> 4 and re-build model until none of VIFs don't exceed 4.
while(any(all_vifs > 4)){
  var_with_max_vif <- names(which(all_vifs == max(all_vifs)))  # get the var with max vif
  signif_all <- signif_all[!(signif_all) %in% var_with_max_vif]  # remove
  myForm <- as.formula(paste("sqrt_cum_cases ~ ", paste (signif_all, collapse=" + "), sep=""))  #
new formula
  selectedMod <- lm(myForm, data=df_for_lm)  # re-build model with new formula
  all_vifs <- car::vif(selectedMod)
}
summary(selectedMod)
# much better!
car::vif(selectedMod)
plot(selectedMod)
summary(selectedMod)
# but, this has insignficant variables

# so lets redo stuff
all_vars <- names(selectedMod[[1]])[-1]  # names of all X variables
# Get the non-significant vars
summ <- summary(selectedMod)  # model summary
pvals <- summ[[4]][, 4]  # get all p values
not_significant <- character()  # init variables that aren't statsitically significant
not_significant <- names(which(pvals > 0.1))
not_significant <- not_significant[!not_significant %in% "(Intercept)"]  # remove 'intercept'.
Optional!

# If there are any non-significant variables,
while(length(not_significant) > 0){
  all_vars <- all_vars[!all_vars %in% not_significant[1]]
  myForm <- as.formula(paste("sqrt_cum_cases ~ ", paste (all_vars, collapse=" + "), sep=""))  #
new formula
  selectedMod <- lm(myForm, data=df_for_lm)  # re-build model with new formula

  # Get the non-significant vars.
  summ <- summary(selectedMod)
  pvals <- summ[[4]][, 4]
  not_significant <- character()
  not_significant <- names(which(pvals > 0.1))
  not_significant <- not_significant[!not_significant %in% "(Intercept)"]
}
```

~~~
summary(selectedMod)
# now all are significant!

# view diagnostics - res vs fitted should have no pattern / random errors normally dist, normal qq should be straight for normality assumption to be met, and scale location should be approximately straight. res vs lev can show influential pts that we can removed and rerun the regression w/ later
# see: https://data.library.virginia.edu/diagnostic-plots/
par(mfrow = c(1,4))
plot(selectedMod)

# check point past cooks distance - 6th obs
df_lm_adj <- df_for_lm[6,]
```

```{r Model without Influential Pts}
# plot outliers for rv
boxplot(df_for_lm$sqrt_cum_cases,
  ylab = "sqrt_cum_cases",
  col = color_pal[5],
  main = "Distribution of sqrt Cumulative Cases"
)
# no outliers via boxplot

# check point past cooks distance - 6th obs, from above plot's last output
df_lm_adj <- df_for_lm[-6,]

lmMod <- lm(sqrt_cum_cases ~ . , data = df_lm_adj)
selectedMod <- step(lmMod)
summary(selectedMod)

# check multicollinearity
all_vifs <- car::vif(selectedMod)
print(all_vifs)

signif_all <- names(all_vifs)

# Remove vars with VIF> 4 and re-build model until none of VIFs don't exceed 4.
while(any(all_vifs > 4)){
  var_with_max_vif <- names(which(all_vifs == max(all_vifs)))  # get the var with max vif
~~~

```
  signif_all <- signif_all[!(signif_all) %in% var_with_max_vif]  # remove
  myForm <- as.formula(paste("sqrt_cum_cases ~ ", paste (signif_all, collapse=" + "), sep=""))  #
new formula
  selectedMod <- lm(myForm, data=df_lm_adj)  # re-build model with new formula
  all_vifs <- car::vif(selectedMod)
}
summary(selectedMod)
# much better!
car::vif(selectedMod)
plot(selectedMod)
summary(selectedMod)
# but, this has insignficant variables

# so lets redo stuff
all_vars <- names(selectedMod[[1]])[-1]  # names of all X variables
# Get the non-significant vars
summ <- summary(selectedMod)  # model summary
pvals <- summ[[4]][, 4]  # get all p values
not_significant <- character()  # init variables that aren't statsitically significant
not_significant <- names(which(pvals > 0.1))
not_significant <- not_significant[!not_significant %in% "(Intercept)"]  # remove 'intercept'.
Optional!

# If there are any non-significant variables,
while(length(not_significant) > 0){
  all_vars <- all_vars[!all_vars %in% not_significant[1]]
  myForm <- as.formula(paste("sqrt_cum_cases ~ ", paste (all_vars, collapse=" + "), sep=""))  #
new formula
  selectedMod <- lm(myForm, data=df_lm_adj)  # re-build model with new formula

  # Get the non-significant vars.
  summ <- summary(selectedMod)
  pvals <- summ[[4]][, 4]
  not_significant <- character()
  not_significant <- names(which(pvals > 0.1))
  not_significant <- not_significant[!not_significant %in% "(Intercept)"]
}
summary(selectedMod)
# now all are significant!
```

~~~
# view diagnostics - res vs fitted should have no pattern / random errors normally dist, normal qq
should be straight for normality assumption to be met, and scale location should be
approximately straight. res vs lev can show influential pts that we can removed and rerun the
regression w/ later
# see: https://data.library.virginia.edu/diagnostic-plots/
par(mfrow = c(1,4))
plot(selectedMod)
```

```{r Quadratic Model}
# https://datascienceplus.com/fitting-polynomial-regression-r/
# NEW MODEL
mod <- lm(sqrt_cum_cases ~ black + I(black^2) + Total_hispanic + I(Total_hispanic^2) +
non_hisp_white, data = df_for_lm)
summary(mod)
par(mfrow = c(2,2))
plot(mod)

# see if dropping black squares is worth w/ anova
drop_black2 <- lm(sqrt_cum_cases ~ black + Total_hispanic + I(Total_hispanic^2) +
non_hisp_white, data = df_for_lm)
anova(drop_black2, mod)
# just barely insignificant at .05 level => drop the squared term for black
drop_black2
plot(drop_black2)
summary(drop_black2)

# remove influential pts - new model
df_rm_influential <- df_for_lm[-c(78, 100), ]
drop_black_inf_rm <- lm(sqrt_cum_cases ~ black + Total_hispanic + I(Total_hispanic^2) +
non_hisp_white, data = df_rm_influential)
plot(drop_black_inf_rm)
summary(drop_black_inf_rm)
```



----------- lag plot below, from other R file

```{r Libraries, message=FALSE}
library(tidyverse)
~~~

~~~~
library(lubridate)
library(data.table)
library(RColorBrewer)
library(corrplot)
```

```{r Formatting}
# color palette for data viz
color_pal <- RColorBrewer::brewer.pal(n = 5, name = "Dark2")
# change scientific notation to standard form
options(scipen = 100)
```



```{r Load in Data}
# change read.csv to read_csv to fix the data reading in issue for dates as columns
covid <- read_csv("C:/Users/razmi/OneDrive/Desktop/Data Analysis Exam
SDSU/SD_Zipcode_COVID_4_DAE_F22_f.csv")
demographics <- read_csv("C:/Users/razmi/OneDrive/Desktop/Data Analysis Exam
SDSU/demographic_SD_ZIP_4_DAE_F22.csv")

# notice: one data set has more rows than the other => some zipcodes are not going to be in both
# potential issue!
dim(covid)
dim(demographics)

# change capitalization to allow for merge
# https://www.sharpsightlabs.com/blog/rename-columns-in-r/
covid <- rename(covid, zipcode = Zipcode)

# check for NAs - none in either data set
# anyNA(covid)
# anyNA(demographics)

# check if any zipcodes got doubled - no duplicates!
# length(unique(covid$zipcode)) == length(covid$zipcode)
# length(unique(demographics$zipcode)) == length(demographics$zipcode)

head(covid)
head(demographics)
~~~~

```
# get total cases for each zip code
covid_drop_zip <- covid %>%
  dplyr::select(-zipcode)
covid_totals <- covid_drop_zip %>%
  rowSums()

# combine datasets
df <- covid %>%
  left_join(demographics)

# https://stackoverflow.com/questions/2185252/reshaping-data-frame-from-wide-to-long-format
# change wide to long format for better usage
covid_long <- melt(setDT(covid), id.vars = c("zipcode"), variable.name = "Date")
covid_long <- rename(covid_long, cases = value)
# summary(covid_long)

# chaneg zipcode to factor
covid_long$zipcode <- as.factor(covid_long$zipcode)

# check number of obs per day - note that they are all 113
# n_per_day <- covid_long %>%
#   group_by(Date) %>%
#   count()
# levels(as.factor(n_per_day$n))

covid_long %>%
  group_by(zipcode, cases) %>%
  count()
# number of rows is in 113 (# obs per zip) * 452 (# days) form

#############################################################################
# total cases per day, regardless of zipcode, cumulative:
cases_per_day <- covid_long %>%
  group_by(Date) %>%
  summarize(count = sum(cases))
day_count <- 1:length(cases_per_day$Date)
cases_per_day2 <- cbind(cases_per_day, day_count)

diff(cases_per_day$count)
```

```
#################################################################################

# for more readable labels, change dates with day as ID
# note it was still cluttered with rotated dates
day_count <- 1:length(cases_per_day$Date)
cases_per_day2 <- cbind(cases_per_day, day_count)

ggplot(cases_per_day2, aes(day_count, count)) +
  geom_point(col = color_pal[2]) +
  labs(x = "Days since COVID", y = "Total Confirmed Cases", title = "COVID Cases on  a Daily
Basis in San Diego County")

####################################################################################
# check change in cases per day w/ lag
n <- length(day_count) - 1
daily_new_cases <- data.frame(new_cases = diff(cases_per_day2$count)) %>%
  cbind(day = 1:n)

# plot new cases per day
ggplot(daily_new_cases, aes(day, new_cases)) +
  geom_point(col = color_pal[1]) +
  labs(x = "New Case Count from Previous Day", y = "Number of Cases", title = "Number of
New COVID Cases Per Day, from 4/1/2022")
# and its distribution
ggplot(daily_new_cases, aes(new_cases)) +
  geom_histogram(color = color_pal[3], fill = color_pal[4]) +
  labs(x = "New Cases", y = "Count")
##################################################################################

# verify above works as expected
# test <- covid_long %>%
#   filter(Date == '4/1/2020')
# sum(test$cases)

# combine datasets again- dif version w/ 19 columns
demographics$zipcode <- as.factor(demographics$zipcode)
df2 <- covid_long %>%
  left_join(demographics, by = "zipcode")

# get daily cases again, with new version of df
```

~~~
daily_case_count <- df2 %>%
  group_by(Date) %>%
  summarize(daily_cases = sum(cases))

summary(daily_case_count$daily_cases)

```


```{r}
# redo df into long
df_long <- melt(setDT(df), id.vars = c("zipcode"), variable.name = "Date")
# note: this has NAs!
# anyNA(df_long)
# length(unique(df$zipcode))
# 113 zipcodes - demographics has less! redo the join later to not have NAs

# join w/ smaller data set to avoid NAs
covid$zipcode <- as.factor(covid$zipcode)
df2 <- demographics %>%
  left_join(covid)
# df2
# get data to long format
# df2_long <- melt(setDT(df2), id.vars = c("zipcode"), variable.name = "Total_pop")
# anyNA(df2_long)
# melt(setDT(df2), id.vars = c("zipcode"), variable.name = "Date")

# # get data to long format
df2_long <- reshape2::melt(df2, id.vars = 1:17, variable.name = "Date")
```


```{r}
# daily case count, regardless of county
daily_case_count <- df2_long %>%
  group_by(Date) %>%
  summarize(daily_cases = sum(value))

day_count <- 1:length(daily_case_count$Date)
cases_per_day2 <- cbind(daily_case_count, day_count)

ggplot(cases_per_day2, aes(day_count, daily_cases)) +
~~~

~~~
  geom_point(col = color_pal[2]) +
  labs(x = "Days since COVID", y = "Total Confirmed Cases", title = "COVID Cases on  a Daily 
Basis in San Diego County")

######################################################################################
# calculate new cases since 4/1/2020
cases_per_day_dif <- diff(cases_per_day2$daily_cases)
day_count_lag1 <- 1:length(cases_per_day_dif)
dif_cases_df <- data.frame(new_cases = cases_per_day_dif, day_dif = cases_per_day_dif)

ggplot(dif_cases_df, aes(day_dif, cases_per_day_dif)) +
  geom_point(col = color_pal[2]) +
  labs(x = "Days since COVID", y = "Total Confirmed Cases", title = "COVID Cases on  a Daily 
Basis in San Diego County")

ggplot(dif_cases_df, aes(cases_per_day_dif)) +
  geom_histogram(col = color_pal[4], fill = color_pal[5], binwidth = 100) +
                        labs(x = "New Cases Compared to Previous Day",
                        y = "Count",
                        title = "Difference in Case Counts Relative to Previous Day")
```
~~~