Final Project- Data Demo

Rohan Dalal-PSTAT 131

Contents

Covid Forecasting using historical data

Introduction:

The Covid 19 Pandemic has dramatically affected day to day life. Due to the nature of the pandemic, policies have been implemented statewide to curb virus spread. Successfully forecasting short-term future Covid 19 cases can aid in implementing policies to reduce infection spread. Goal of this project is to use data from counties throughout California from May-July of 2020 and run it through various machine learning forecasting models.

Data source

The covidcast R package, which provides access to the COVIDcast Epidata API published by the Delphi group at Carnegie Mellon University. According to the covidcast R package website,This API provides daily access to a range of COVID-related signals Delphi that builds and maintains, from sources like symptom surveys and medical claims data, and also standard signals that we simply mirror, like confirmed cases and deaths. (see website here) Here is a list of the signals, we can see all the documentation for each one. This includes information about when the first data points were collected, if the data is available on a daily, or weekly basis, what regions we can call the signal for, and so on.

Data overview: Fetching/Merging/Prepping data

I plan to choose five signals to predict cases across California counties. Predictor : "visits","admits", "chngVisits" ,"covidChngVisits","gsymptoms" and Outcome : "Cases"

- "Cases": Get the number of daily new Covid cases for all the counties in California,for a given date range (example :from May 2020 to July 2020) by fetching the "US Facts Cases and Deaths" data source (https://cmu-delphi.github.io/delphi-epidata/api/covidcast-signals/usa-facts.html). This will be the Ground Truth(label)
- "visits": Get the daily percentages of doctor visits that are related to Covid in California for a given date range (example :from May 2020 to July 2020) by fetching the "Doctor Visits" data source (https://cmu-delphi.github.io/delphi-epidata/api/covidcast-signals/doctor-visits.html).
- "admits" :Get the daily hospital admissions for covid diagnosed that are related to Covid in California for a given date range (example :from May 2020 to July 2020) by fetching the "Doctor Visits" data source (https://cmu-delphi.github.io/delphi-epidata/api/covidcast-signals/hospital-admissions.html).
- "chngVisits": Get the Estimated percentage of outpatient doctor visits primarily about COVID-related symptoms in California for a given date range (example :from May 2020 to July 2020) by fetching the "Doctor Visits" data source (https://cmu-delphi.github.io/delphi-epidata/api/covidcast-signals/chng.html).
- "covidChngVisits": Get the Estimated percentage of outpatient doctor visits with confirmed COVID-19, based on Change Healthcare claims data in California for a given date range (example :from May 2020 to July 2020) by fetching the "Doctor Visits" data source (https://cmu-delphi.github.io/delphi-epidata/api/covidcast-signals/chng.html).
- "gsymptoms":Get Sum of Google search volume for anosmia and ageusia related searches in California for a given date range (example :from May 2020 to July 2020) by fetching the "Doctor Visits" data source (https://cmu-delphi.github.io/delphi-epidata/api/covidcast-signals/google-symptoms.html).

Data PreProcessing

- Get the required signals and merge data to create a csv file, clean up and tidy data.
- Observation Count: After collecting needed predictors from the data sources contains about 5428 observations, not all the predictors have missing value but some do.
- Analyze the data fetched for datatype and null/missing values – Dealing with missing/NA data : Method:1-dropping rows with missing values-disadvantage is smaller data set for modeling Method:2-imputation method-disadvantage it might limit the effectiveness of the model

I am planning to do exploratory analysis to see accuracy and effectivenes compariosn by both methods.

Motivation/Goal

Goal is to: -to build predictive models that forecast the future of the pandemic so that we can see one step ahead and prepare accordingly using the past data. -to build a predictive model that uses historical COVID cases and related data to forecast the short-term future number of COVID cases in a particular region.

Project Timeline

- April 8 -April 14: Load and tidy data
- April 14 -April 24: Exploratory analysis and Model selection
- April 24- May 10: Test and Run models
- May 10 -May 24 ; work on draft paper
- May-25 - June 2 : Any edits and finalize paper

Data collection efforts using covidcast package.

```
#install.packages('covidcast')
library(covidcast)
library(ggplot2)
# Cumulative COVID cases per 100k people on 2020-12-31
df <- covidcast_signal(data_source = "usa-facts",
                  signal = "confirmed_cumulative_prop",
                  start_day = "2020-12-31", end_day = "2020-12-31")
summary(df)
```

```
## A `covidcast_signal` dataframe with 3142 rows and 12 columns.
##
## data_source : usa-facts
## signal      : confirmed_cumulative_prop
## geo_type    : county
##
## first date                          : 2020-12-31
## last date                           : 2020-12-31
## median number of geo_values per day : 3142
```

```
# This looks at the people who reported COVID-like symptoms from their fb-survey
# from dates 5-1-2020 to 5-7-2020 in all counties
data <- covidcast_signal("fb-survey", "smoothed_cli", start_day = "2020-05-01",
                         end_day = "2020-05-07")
head(data)
```

```
##   data_source       signal geo_value time_value      issue lag missing_value
## 1   fb-survey smoothed_cli     01000 2020-05-01 2020-09-03 125             0
## 2   fb-survey smoothed_cli     01001 2020-05-01 2020-09-03 125             0
## 3   fb-survey smoothed_cli     01003 2020-05-01 2020-09-03 125             0
## 4   fb-survey smoothed_cli     01015 2020-05-01 2020-09-03 125             0
## 5   fb-survey smoothed_cli     01031 2020-05-01 2020-09-03 125             0
## 6   fb-survey smoothed_cli     01045 2020-05-01 2020-09-03 125             0
##   missing_stderr missing_sample_size     value    stderr sample_size
## 1              0                   0 0.8254101 0.1360033   1722.4551
## 2              0                   0 1.2994255 0.9671356    115.8025
## 3              0                   0 0.6965968 0.3247531    584.3194
## 4              0                   0 0.4282713 0.5485655    122.5577
## 5              0                   0 0.0255788 0.3608268    114.8318
## 6              0                   0 1.0495589 0.7086324    110.6544
```

```
# Get list of all counties in california state and store in ca_counties
county_code<-c('06000', '06001', '06003', '06005', '06007',
 '06009', '06011', '06013', '06015', '06017',
 '06019', '06021', '06023', '06025', '06027',
 '06029', '06031', '06033', '06035', '06037',
 '06039', '06041', '06043', '06045', '06047',
 '06049', '06051', '06053', '06055', '06057',
 '06059', '06061', '06063', '06065', '06067',
 '06069', '06071', '06073', '06075', '06077',
 '06079', '06081', '06083', '06085', '06087',
 '06089', '06091', '06093', '06095', '06097',
 '06099', '06101', '06103', '06105', '06107',
 '06109', '06111', '06113', '06115')
ca_counties <- county_fips_to_name(county_code)
```

```
#hospital admissions for covid diagnosed in time span defined for all counties in california state
admits <- covidcast_signal(data_source ="hospital-admissions", "smoothed_adj_covid19_from_claims",
                       start_day = "2020-05-01",
                        end_day = "2020-07-31", time_type = "day",
                         geo_type="county", geo_values=county_code)

head(admits)
```

```
##           data_source                          signal geo_value time_value
## 1 hospital-admissions smoothed_adj_covid19_from_claims     06000 2020-05-01
## 2 hospital-admissions smoothed_adj_covid19_from_claims     06001 2020-05-01
## 3 hospital-admissions smoothed_adj_covid19_from_claims     06013 2020-05-01
## 4 hospital-admissions smoothed_adj_covid19_from_claims     06031 2020-05-01
## 5 hospital-admissions smoothed_adj_covid19_from_claims     06037 2020-05-01
## 6 hospital-admissions smoothed_adj_covid19_from_claims     06059 2020-05-01
##        issue lag missing_value missing_stderr missing_sample_size    value
## 1 2020-07-03  63             0              5                   5 0.493246
## 2 2020-07-03  63             0              5                   5 3.260620
## 3 2020-07-03  63             0              5                   5 0.140425
## 4 2020-07-03  63             0              5                   5 0.410873
## 5 2020-07-03  63             0              5                   5 3.589725
## 6 2020-07-03  63             0              5                   5 0.700736
##   stderr sample_size
```

```
## 1     NA          NA
## 2     NA          NA
## 3     NA          NA
## 4     NA          NA
## 5     NA          NA
## 6     NA          NA
```

```
#Doctor visits primarily about COVID-related symptom in time span defined for all counties in californi
visits <- covidcast_signal(data_source ="doctor-visits", "smoothed_cli",
                        start_day = "2020-05-01",
                         end_day = "2020-07-31", time_type = "day",
                          geo_type="county", geo_values=county_code)

head(visits)
```

```
##     data_source       signal geo_value time_value      issue lag missing_value
## 1 doctor-visits smoothed_cli     06000 2020-05-01 2020-07-04  64             0
## 2 doctor-visits smoothed_cli     06001 2020-05-01 2020-07-04  64             0
## 3 doctor-visits smoothed_cli     06005 2020-05-01 2020-07-04  64             0
## 4 doctor-visits smoothed_cli     06007 2020-05-01 2020-07-04  64             0
## 5 doctor-visits smoothed_cli     06009 2020-05-01 2020-07-04  64             0
## 6 doctor-visits smoothed_cli     06011 2020-05-01 2020-07-04  64             0
##   missing_stderr missing_sample_size    value stderr sample_size
## 1              5                   5 1.326328     NA          NA
## 2              5                   5 2.712027     NA          NA
## 3              5                   5 0.386714     NA          NA
## 4              5                   5 0.728788     NA          NA
## 5              5                   5 0.852603     NA          NA
## 6              5                   5 0.000000     NA          NA
```

```
#county_fips_to_name(visits$geo_value)
```

```
#Doctor visits primarily about COVID-related symptom from Change Healthcare data
chngVisits <- covidcast_signal(data_source ="chng", "smoothed_adj_outpatient_cli",
                        start_day = "2020-05-01",
                         end_day = "2020-07-31",time_type = "day",
                          geo_type="county", geo_values=county_code)

head(chngVisits)
```

```
##   data_source                      signal geo_value time_value      issue lag
## 1        chng smoothed_adj_outpatient_cli     06000 2020-05-01 2021-02-21 296
## 2        chng smoothed_adj_outpatient_cli     06001 2020-05-01 2021-12-06 584
## 3        chng smoothed_adj_outpatient_cli     06005 2020-05-01 2021-12-06 584
## 4        chng smoothed_adj_outpatient_cli     06007 2020-05-01 2021-12-06 584
## 5        chng smoothed_adj_outpatient_cli     06009 2020-05-01 2021-12-06 584
## 6        chng smoothed_adj_outpatient_cli     06011 2020-05-01 2021-12-06 584
##   missing_value missing_stderr missing_sample_size     value stderr sample_size
## 1             0              5                   5 2.4825390     NA          NA
## 2             0              5                   5 2.0899372     NA          NA
## 3             0              5                   5 0.0970874     NA          NA
## 4             0              5                   5 0.2979666     NA          NA
## 5             0              5                   5 0.1941305     NA          NA
## 6             0              5                   5 0.0984252     NA          NA
```

```r
#Doctor visits primarily about COVID symptom from Change Healthcare data
covidChngVisits<- covidcast_signal(data_source ="chng", "smoothed_adj_outpatient_covid",
                        start_day = "2020-05-01",
                         end_day = "2020-07-31",time_type = "day",
                          geo_type="county", geo_values=county_code)

head(covidChngVisits)
```

```
##   data_source                        signal geo_value time_value      issue lag
## 1        chng smoothed_adj_outpatient_covid     06000 2020-05-01 2021-02-21 296
## 2        chng smoothed_adj_outpatient_covid     06001 2020-05-01 2021-12-06 584
## 3        chng smoothed_adj_outpatient_covid     06005 2020-05-01 2021-12-06 584
## 4        chng smoothed_adj_outpatient_covid     06007 2020-05-01 2021-12-06 584
## 5        chng smoothed_adj_outpatient_covid     06009 2020-05-01 2021-12-06 584
## 6        chng smoothed_adj_outpatient_covid     06011 2020-05-01 2021-12-06 584
##   missing_value missing_stderr missing_sample_size     value stderr sample_size
## 1             0              5                    5 0.4291050     NA          NA
## 2             0              5                    5 0.1879091     NA          NA
## 3             0              5                    5 0.0970874     NA          NA
## 4             0              5                    5 0.1651790     NA          NA
## 5             0              5                    5 0.0904159     NA          NA
## 6             0              5                    5 0.0984252     NA          NA
```

```r
# Sum of Google search volume for anosmia and ageusia related searches
gsymptoms <- covidcast_signal(data_source ="google-symptoms", "sum_anosmia_ageusia_smoothed_search",
                        start_day = "2020-05-01",
                         end_day = "2020-07-31",time_type = "day",
                          geo_type="county", geo_values=county_code)

head(gsymptoms)
```

```
##       data_source                              signal geo_value time_value
## 1 google-symptoms sum_anosmia_ageusia_smoothed_search     06001 2020-05-01
## 2 google-symptoms sum_anosmia_ageusia_smoothed_search     06037 2020-05-01
## 3 google-symptoms sum_anosmia_ageusia_smoothed_search     06059 2020-05-01
## 4 google-symptoms sum_anosmia_ageusia_smoothed_search     06065 2020-05-01
## 5 google-symptoms sum_anosmia_ageusia_smoothed_search     06067 2020-05-01
## 6 google-symptoms sum_anosmia_ageusia_smoothed_search     06071 2020-05-01
##        issue lag missing_value missing_stderr missing_sample_size     value
## 1 2021-01-14 258             0              5                    5 0.2200000
## 2 2021-01-14 258             0              5                    5 0.2171429
## 3 2021-01-14 258             0              5                    5 0.1742857
## 4 2021-01-14 258             0              5                    5 0.1771429
## 5 2021-01-14 258             0              5                    5 0.1314286
## 6 2021-01-14 258             0              5                    5 0.1871429
##   stderr sample_size
## 1     NA          NA
## 2     NA          NA
## 3     NA          NA
## 4     NA          NA
## 5     NA          NA
## 6     NA          NA
```

```r
#Number of new confirmed COVID-19 cases, daily for Ground Truth(label)
cases <- covidcast_signal(data_source ="indicator-combination", "confirmed_incidence_num",
                          start_day = "2020-05-01",
                           end_day = "2020-07-31",time_type = "day",
                            geo_type="county", geo_values=county_code)

head(cases)
```

```
##             data_source                  signal geo_value time_value      issue
## 1 indicator-combination confirmed_incidence_num     06000 2020-05-01 2020-11-12
## 2 indicator-combination confirmed_incidence_num     06001 2020-05-01 2020-11-12
## 3 indicator-combination confirmed_incidence_num     06003 2020-05-01 2020-11-12
## 4 indicator-combination confirmed_incidence_num     06005 2020-05-01 2020-11-12
## 5 indicator-combination confirmed_incidence_num     06007 2020-05-01 2020-11-12
## 6 indicator-combination confirmed_incidence_num     06009 2020-05-01 2020-11-12
##   lag missing_value missing_stderr missing_sample_size value stderr sample_size
## 1 195             0              5                    5     0     NA          NA
## 2 195             0              5                    5    33     NA          NA
## 3 195             0              5                    5     0     NA          NA
## 4 195             0              5                    5     0     NA          NA
## 5 195             0              5                    5     0     NA          NA
## 6 195             0              5                    5     0     NA          NA
```

```r
# Merge all the signals fetched above (5 feature +one label)
data <-aggregate_signals( list(visits, admits, chngVisits, covidChngVisits,gsymptoms,cases))

head(data)
```

```
##   geo_value time_value value+0:doctor-visits_smoothed_cli
## 1     06000 2020-07-31                          3.554633
## 2     06001 2020-07-31                          5.691242
## 3     06005 2020-07-31                          0.692647
## 4     06007 2020-07-31                          3.349509
## 5     06009 2020-07-31                          3.189537
## 6     06011 2020-07-31                          4.740800
##   value+0:hospital-admissions_smoothed_adj_covid19_from_claims
## 1                                                     5.086037
## 2                                                     1.525183
## 3                                                           NA
## 4                                                           NA
## 5                                                           NA
## 6                                                           NA
##   value+0:chng_smoothed_adj_outpatient_cli
## 1                                16.411706
## 2                                10.408440
## 3                                 3.002966
## 4                                 1.888170
## 5                                 6.243463
## 6                                27.298557
##   value+0:chng_smoothed_adj_outpatient_covid
## 1                                   0.232762
## 2                                   1.284386
## 3                                   0.933840
```

```
## 4                                    0.083726
## 5                                    2.130005
## 6                                    1.046734
##   value+0:google-symptoms_sum_anosmia_ageusia_smoothed_search
## 1                                                         NA
## 2                                                  0.2657143
## 3                                                         NA
## 4                                                         NA
## 5                                                         NA
## 6                                                         NA
##   value+0:indicator-combination_confirmed_incidence_num
## 1                                                     0
## 2                                                     0
## 3                                                     6
## 4                                                    34
## 5                                                    17
## 6                                                    11
```

```
#Fetch only needed data, rename to sensible column headers
library(dplyr)
#names(data)
library(janitor)
data<-data%>% clean_names()

#names(data)
df =data%>% rename(
            visits= value_0_doctor_visits_smoothed_cli,
            admits = value_0_hospital_admissions_smoothed_adj_covid19_from_claims,
            chngVisits =value_0_chng_smoothed_adj_outpatient_cli ,
            covidChngVisits =value_0_chng_smoothed_adj_outpatient_covid ,
            gsymptoms = value_0_google_symptoms_sum_anosmia_ageusia_smoothed_search ,
            cases = value_0_indicator_combination_confirmed_incidence_num
            )


head(df)
```

```
##   geo_value time_value   visits   admits chngVisits covidChngVisits gsymptoms
## 1     06000 2020-07-31 3.554633 5.086037  16.411706        0.232762        NA
## 2     06001 2020-07-31 5.691242 1.525183  10.408440        1.284386 0.2657143
## 3     06005 2020-07-31 0.692647       NA   3.002966        0.933840        NA
## 4     06007 2020-07-31 3.349509       NA   1.888170        0.083726        NA
## 5     06009 2020-07-31 3.189537       NA   6.243463        2.130005        NA
## 6     06011 2020-07-31 4.740800       NA  27.298557        1.046734        NA
##   cases
## 1     0
## 2     0
## 3     6
## 4    34
## 5    17
## 6    11
```

```r
#Analyze the data fetched for datatype and null/missing values
dim(df)
```

```
## [1] 5428    8
```

```r
colSums(is.na(df))
```

```
##       geo_value      time_value          visits          admits      chngVisits
##               0               0            1149            3329              92
## covidChngVisits       gsymptoms           cases
##              92            4382               0
```

Data cleaning

```r
#Preprocessing Method:1-dropping rows with missing values-disadvantage is smaller data set for modeling

newdf<-na.omit(df)
dim(newdf)
```

```
## [1] 1030    8
```

```r
colSums(is.na(newdf))
```

```
##       geo_value      time_value          visits          admits      chngVisits
##               0               0               0               0               0
## covidChngVisits       gsymptoms           cases
##               0               0               0
```

```r
# Write filtered data into a new file.
write.csv(newdf,"completedata.csv")

#completedata.csv is ready to be used for modeling
```

Data imputation

```r
#Preprocessing Method:2-imputation method-disadvantage it might limit the effectiveness of your model
#clean data for missing values by imputation method(replace missing with mode values)
newdf1 <- df
# Return the column names containing missing observations
list_na <- colnames(newdf1)[ apply(newdf1, 2, anyNA) ]
# Create mean
average_missing <- apply(newdf1[,colnames(newdf1) %in% list_na],
      2,
      mean,
      na.rm =  TRUE)
average_missing
```

```
##          visits          admits      chngVisits covidChngVisits       gsymptoms
##       3.4267524       2.7863828       3.7105346       0.4316880       0.3311349
```

```
# Create a new variable with the mean and median
newdf1_replace <- newdf1 %>%
   mutate(visits  = ifelse(is.na(visits), average_missing[1], visits),
          admits = ifelse(is.na(admits), average_missing[2], admits),
          chngVisits = ifelse(is.na(chngVisits), average_missing[3], chngVisits),
          covidChngVisits = ifelse(is.na(covidChngVisits), average_missing[4], covidChngVisits),
          gsymptoms = ifelse(is.na(gsymptoms), average_missing[5], gsymptoms)
          )
colSums(is.na(newdf1_replace))
```

```
##       geo_value      time_value          visits          admits      chngVisits
##               0               0               0               0               0
## covidChngVisits       gsymptoms           cases
##               0               0               0
```

```
write.csv(newdf1_replace,"imputeddata.csv")
#imputeddata.csv is ready to be used for modeling
```

In data gathering process, I started data analysis and found lots of missing values for google trends signals and many counties did not have sufficient data to consider hence generated 2 different datasets "imputeddata.csv" (newdf1) and "completedata.csv"(newdf) to test different approaches.

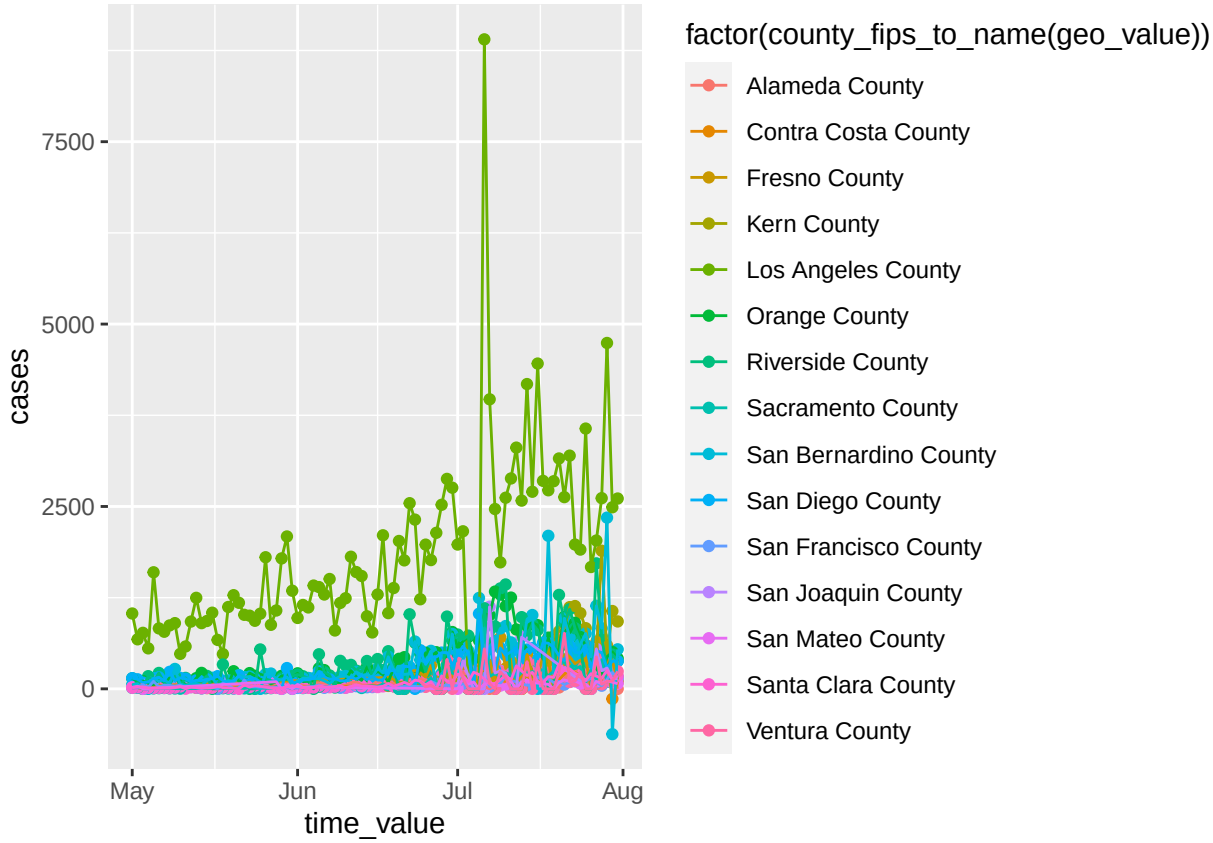
Data Imputation : In generated "completedata.csv"(newdf), chose to drop all observations that have missing values. Upon inspection, over three thousand of the 5428 entries had one or more missing fields. Removing all the data with missing value leads to a disadvantage to having a smaller data set for modeling. We end up with 1030 datapoints in our clean data.

In second set "imputeddata.csv" I imputed by replacing the mode values, generated data was a larger set but disadvantage is going to be limiting the effectiveness of the model.Finally, we decided it would be best to keep the data true versus embedding values like the mean/median for so many entries and look for models that performed well despite having less data. Choice was to run models against "completedata.csv" with 1030 data points.

Exploratory Data Analysis

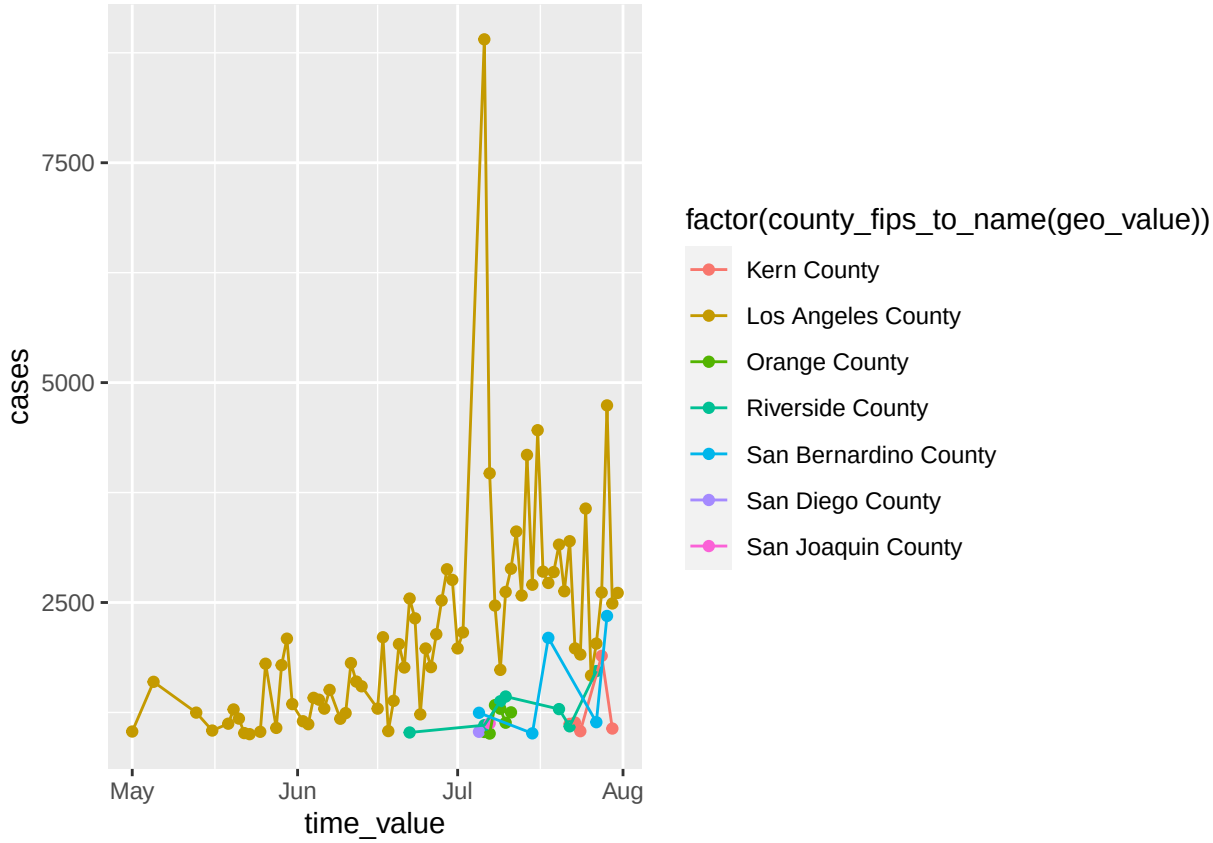
This entire data analysis will be based only on the data set that excludes missing value so our modelling is effective where all the variables are represented. First we will try to find the county that has maximum covid cases documented with the time interval chosen for the test.

```
newdf %>%
group_by(geo_value , time_value) %>%
ggplot(aes(x = time_value, y = cases, colour=factor(county_fips_to_name(geo_value))))+
  geom_point() + geom_line()
```



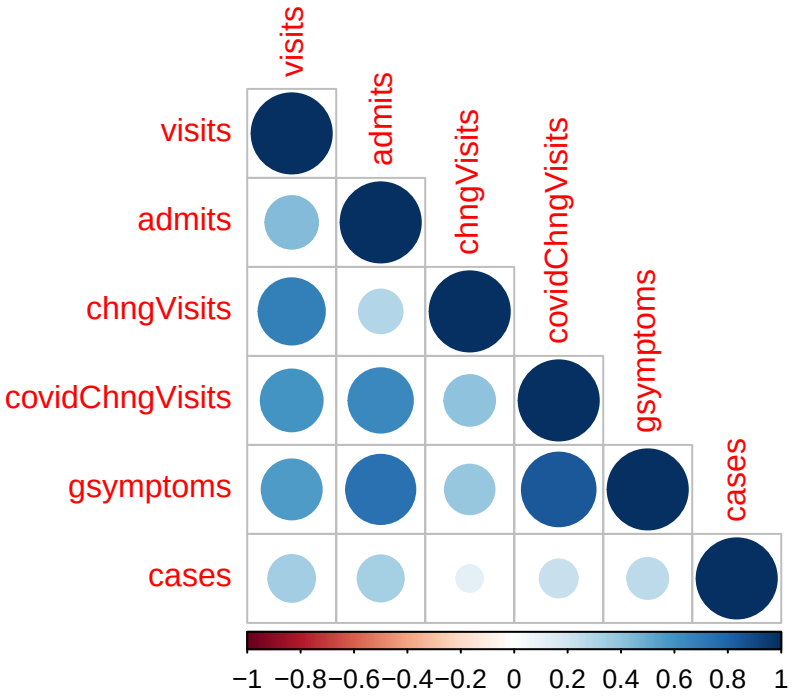
Above plot clearly shows that Los Angeles county has the heighest cases recorded. We are going to take a close look at the data by filtering on cases count greater than 1000.

```
sorted<-newdf %>%
  group_by(geo_value , time_value) %>%
   filter(cases > 1000) %>%
   arrange(geo_value)


sorted %>%
group_by(geo_value , time_value) %>%
ggplot(aes(x = time_value, y = cases, colour=factor(county_fips_to_name(geo_value))))+
  geom_point() + geom_line()
```



It is important to see the corelation on number of cases generated to the signals choses to collect the matrix, Drawing a coorplot will give us an idea of strength of the signals that adds value to the data modeling for future forecasting

```
library(corrplot)
library(corrr)

newdf %>%
  select(is.numeric) %>%
  cor() %>%
  corrplot(type = "lower")
```



Number if visits and hospital admission are positivly correlated to the number of cases. Anosmia_ageusia google symptoms searches also have a postive corelation to the factor.

Data Splitting for Cross Validation and Prediction:

Firstly, we train the models on covid cases collected from different data sources for all counties in CA for a time period: May 1, 2020 to -June30, 2020 (training set). We predicted the remaining data from July 1-July30. Our metric will be the Root Mean Squared Error (RMSE) computed with the predicted and ground-truth time series

```
library(tidyverse)
library(tidymodels)

# sort the date first so we can split the data set
newdf <- newdf %>%
group_by(geo_value , time_value) %>%
   arrange(time_value)
# after sorting the dataframe, split the dataframe
split_date <-'2020-06-30'
filterdf_train <-newdf %>%
  filter(time_value <= split_date)
filterdf_test <-newdf %>%
  filter(time_value >split_date)
tail(filterdf_train)
```

```
## # A tibble: 6 x 8
```

```
## # Groups:   geo_value, time_value [6]
##   geo_value time_value visits admits chngVisits covidChngVisits gsymptoms cases
##   <chr>     <date>      <dbl>  <dbl>      <dbl>           <dbl>     <dbl> <dbl>
## 1 06067     2020-06-30   3.05  0.462       2.08           0.343     0.313   219
## 2 06071     2020-06-30   4.70  4.05        5.07           0.723     0.766   753
## 3 06073     2020-06-30   6.82  1.36        3.24           0.444     0.389   317
## 4 06075     2020-06-30   7.92  0.435      16.3            0.342     0.23     42
## 5 06085     2020-06-30   4.30  1.41        4.56           0.280     0.207   105
## 6 06111     2020-06-30   4.35  1.68        8.77           0.318     0.213     0
```

```
dim(filterdf_train)
```

```
## [1] 577   8
```

```
dim(filterdf_test)
```

```
## [1] 453   8
```

```
ggplot(filterdf_test)+geom_point(aes(x = time_value, y = cases, col = "red")) +
  geom_point(data = filterdf_train, aes(x = time_value, y = cases), col = "green")
```

Model Building

The training data set has about 577 observations and the testing data set has just under 453 bservations.

```
filterdf_recipe <- recipe(cases ~ visits+ admits + chngVisits + covidChngVisits+
                          gsymptoms, filterdf_train) %>%
  step_novel(all_nominal_predictors()) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_predictors())
```

Linear Model

```
lm_model <- linear_reg() %>%
  set_engine("lm")
lm_wflow <- workflow() %>%
  add_model(lm_model) %>%
  add_recipe(filterdf_recipe)


lm_fit <- fit(lm_wflow, filterdf_train)

lm_fit %>%
  # This returns the parsnip object:
  extract_fit_parsnip() %>%
  # Now tidy the linear model object:
  tidy()
```

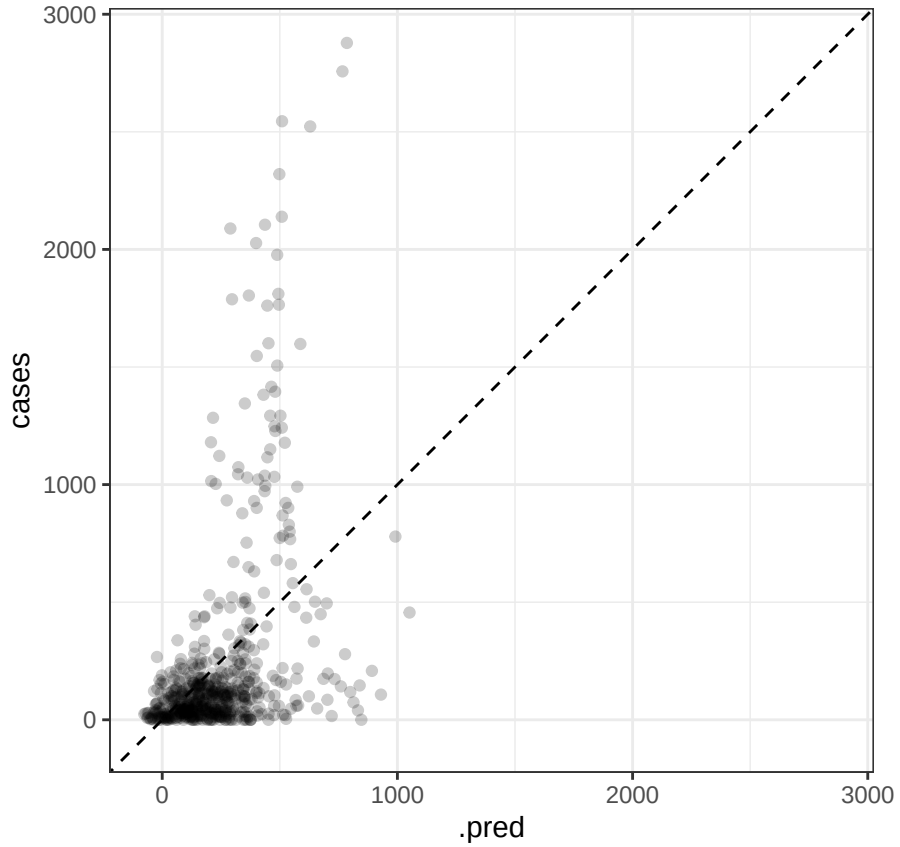
```
## # A tibble: 6 x 5
##   term            estimate std.error statistic  p.value
##   <chr>              <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)        244.       16.3     15.0  2.83e-43
## 2 visits              90.5      22.4      4.04 6.18e- 5
## 3 admits             124.       18.3      6.79 2.75e-11
## 4 chngVisits         -25.2      20.4     -1.24 2.17e- 1
## 5 covidChngVisits     83.4      28.4      2.94 3.41e- 3
## 6 gsymptoms          -46.8      28.7     -1.63 1.03e- 1
```

```
filtered_train_res <- predict(lm_fit, new_data = filterdf_train %>% select(-cases))


filtered_train_res <- bind_cols(filtered_train_res,filterdf_train )
filtered_train_res %>%
  head()
```

```
## # A tibble: 6 x 9
##   .pred geo_value time_value visits admits chngVisits covidChngVisits gsymptoms
##   <dbl> <chr>     <date>      <dbl>  <dbl>      <dbl>           <dbl>     <dbl>
## 1 256.  06001     2020-05-01  2.71   3.26       2.09           0.188      0.22
## 2 477.  06037     2020-05-01  5.79   3.59       2.64           0.336      0.217
## 3 202.  06059     2020-05-01  4.82   0.701      1.80           0.235      0.174
## 4 659.  06065     2020-05-01  4.54   6.58       1.11           0.314      0.177
## 5  23.2 06067     2020-05-01  0.998  1.08       0.426          0.0905     0.131
```

```
## 6 131.  06071     2020-05-01  1.93   1.42       1.33           0.285      0.187
## # ... with 1 more variable: cases <dbl>
```

```
filtered_train_res %>%
  ggplot(aes(x = .pred, y = cases)) +
  geom_point(alpha = 0.2) +
  geom_abline(lty = 2) +
  theme_bw() +
  coord_obs_pred()
```




```
rmse(filtered_train_res, truth = cases, estimate = .pred)
```

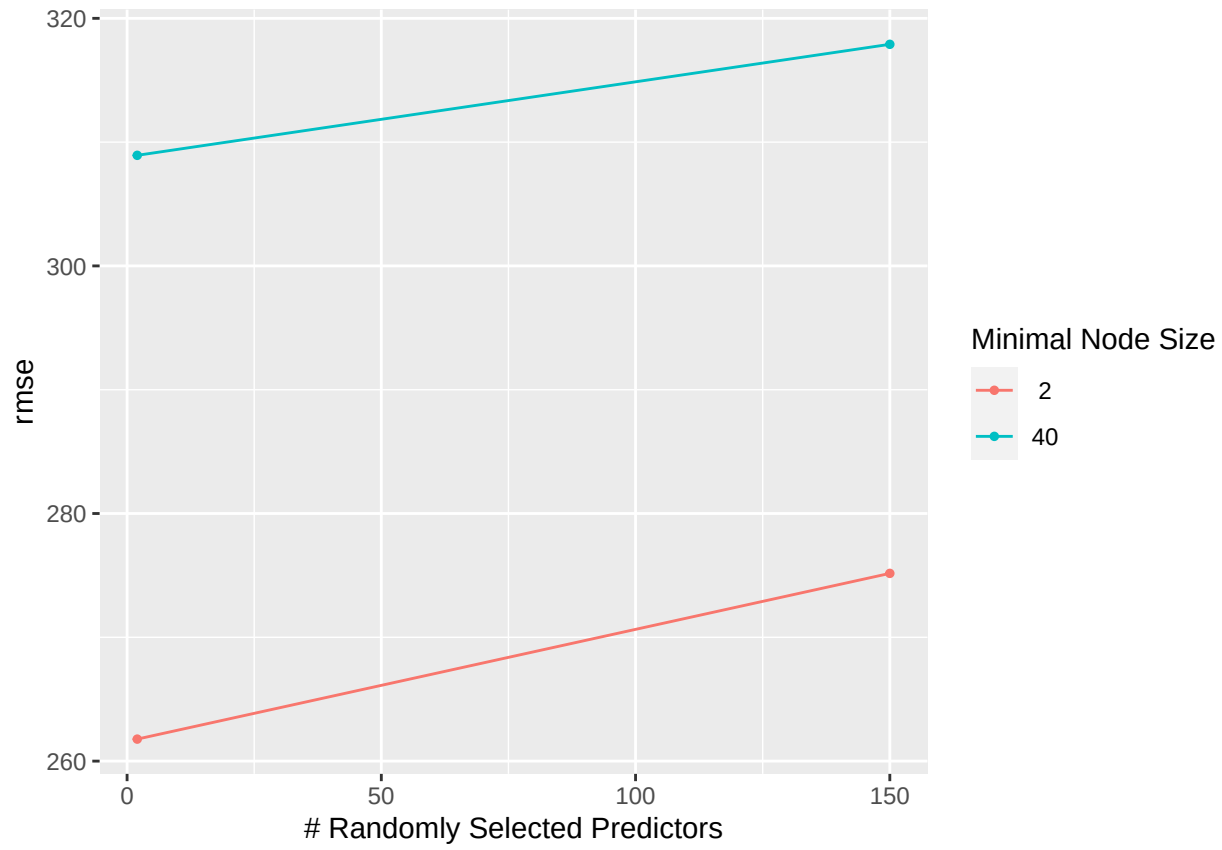
```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        388.
```

Random Forrest Model

```
rf_model <-
  rand_forest(
              min_n = tune(),
```

```
              mtry = tune(),
              mode = "regression") %>%
  set_engine("ranger")

rf_workflow <- workflow() %>%
  add_model(rf_model) %>%
  add_recipe(filterdf_recipe)
```

```
Auto_folds <-  vfold_cv(filterdf_train, strata = cases, v = 5)
rf_params <- parameters(rf_model) %>%
  update(mtry = mtry(range= c(2, 150)))

# define grid
rf_grid <- grid_regular(rf_params, levels = 2)

rf_tune <- rf_workflow %>%
  tune_grid(
    resamples = Auto_folds,
    # how does it complete the models in those workflows
    grid = rf_grid)
autoplot(rf_tune, metric = "rmse")
```




```
# Write Out Results & Workflow ----
#save(rf_tune, rf_workflow, file = "data/model_fitting/rf_tune.rda")
```

Taking a quick peak at the autoplot() function, it is clear that rmse increasing as the number of randomly selected predictors increases.

```
show_best(rf_tune, metric = "rmse") %>% select(-.estimator, -.config)
```

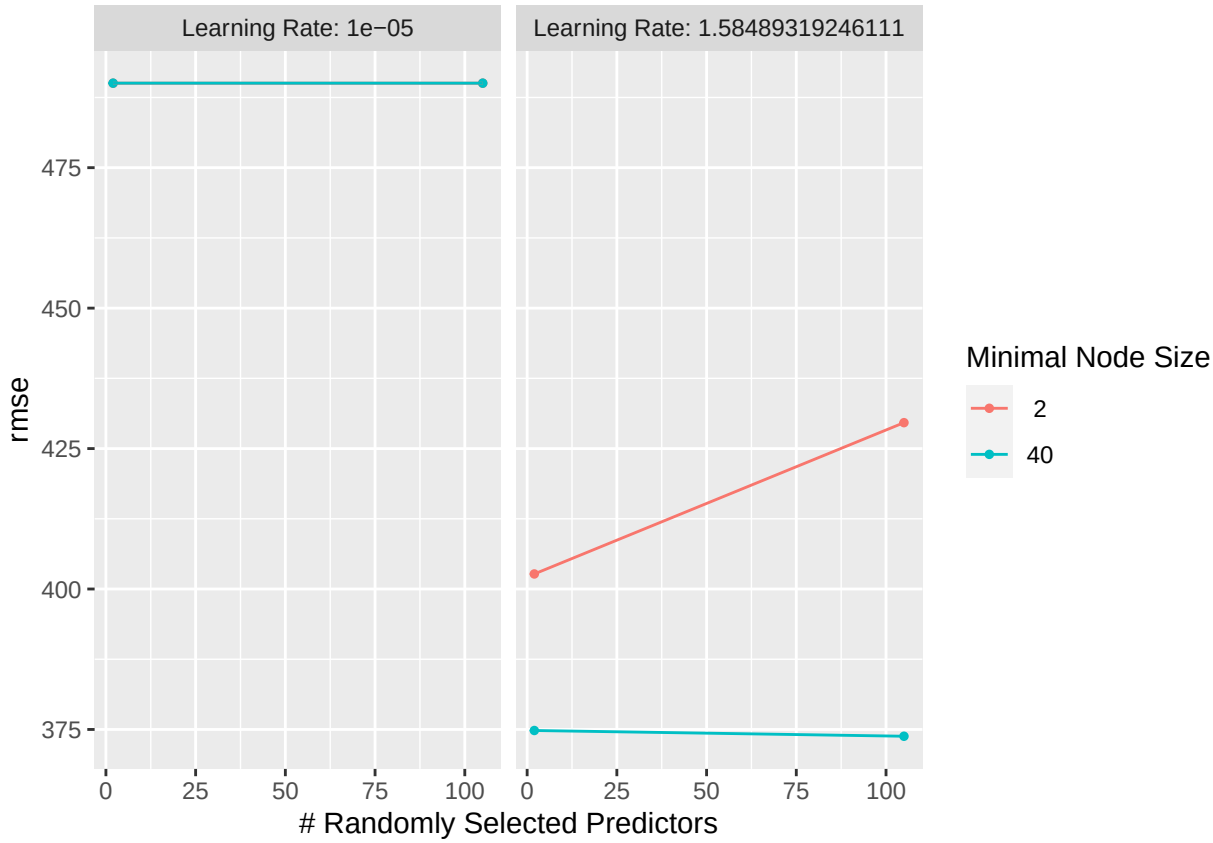
```
## # A tibble: 4 x 6
##    mtry min_n .metric  mean     n std_err
##   <int> <int> <chr>   <dbl> <int>   <dbl>
## 1     2     2 rmse     262.     5    28.1
## 2   150     2 rmse     275.     5    26.6
## 3     2    40 rmse     309.     5    27.8
## 4   150    40 rmse     318.     5    25.2
```

Boost Tree Model

```
bt_model <- boost_tree(mode = "regression",
                       min_n = tune(),
                       mtry = tune(),
                       learn_rate = tune()) %>%
  set_engine("xgboost")

bt_workflow <- workflow() %>%
  add_model(bt_model) %>%
  add_recipe(filterdf_recipe)
```

```
bt_params <- parameters(bt_model) %>%
  update(mtry = mtry(range= c(2, 105)),
         learn_rate = learn_rate(range = c(-5, 0.2))
  )

# define grid
bt_grid <- grid_regular(bt_params, levels = 2)
```

```
bt_tune <- bt_workflow %>%
  tune_grid(
    resamples = Auto_folds,
    grid = bt_grid
    )
autoplot(bt_tune, metric = "rmse")
```




```
show_best(bt_tune, metric = "rmse") %>% select(-.estimator, -.config)
```

```
## # A tibble: 5 x 7
##    mtry min_n learn_rate .metric  mean     n std_err
##   <int> <int>      <dbl> <chr>   <dbl> <int>   <dbl>
## 1   105    40    1.58    rmse     374.     5    20.9
## 2     2    40    1.58    rmse     375.     5    21.8
## 3     2     2    1.58    rmse     403.     5    20.7
## 4   105     2    1.58    rmse     430.     5    15.4
## 5     2     2    0.00001 rmse     490.     5    39.8
```

Final Model Building

We'll create a workflow that has tuned in the name, so we can identify it. We'll finalize the workflow by taking the parameters from the best model (the random forest model) using the select_best() function.

Analysis of The Test Set:

lets fit the model to the testing data set and create a few stored data sets for some analysis!

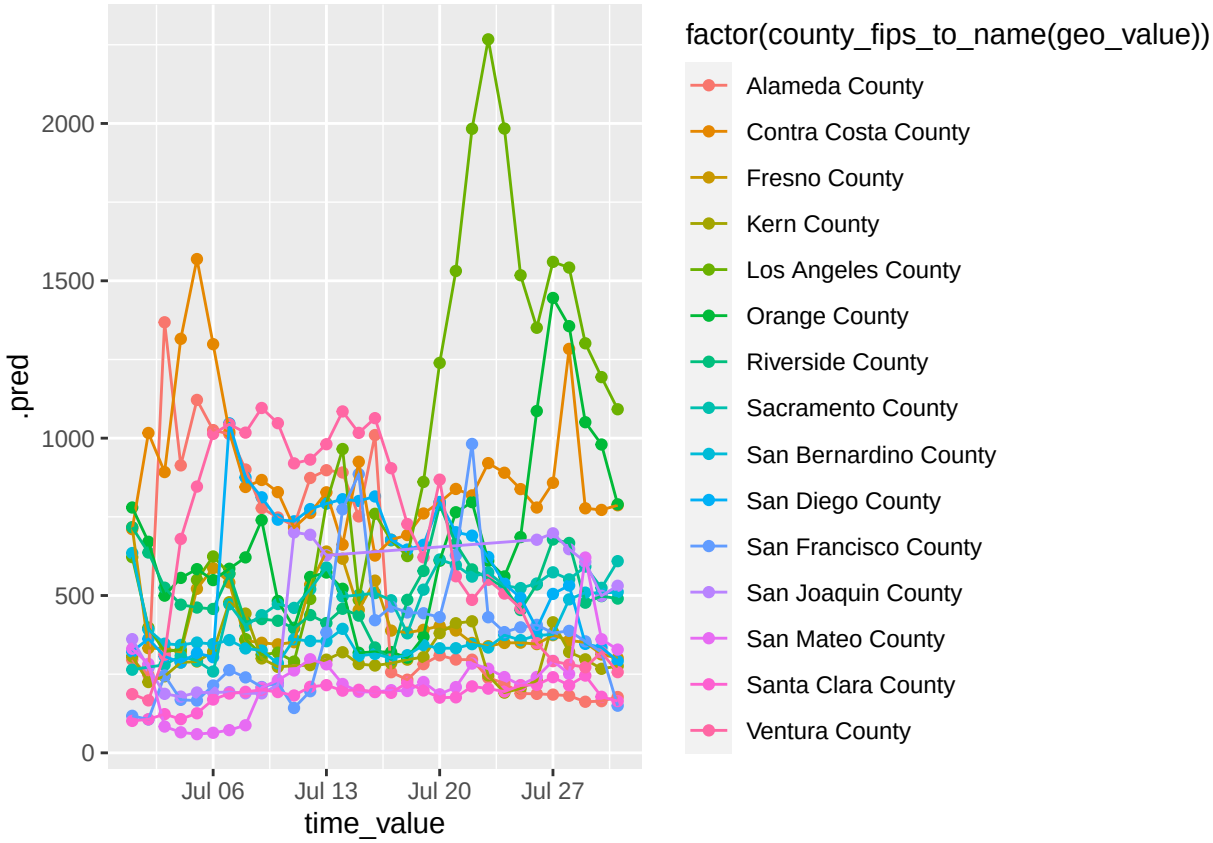
```
rf_workflow_tuned <- rf_workflow %>%
  finalize_workflow(select_best(rf_tune, metric = "rmse"))
```

```
rf_results <- fit(rf_workflow_tuned, filterdf_train)

final_metric <- metric_set(rmse)

model_test_predictions <- predict(rf_results, new_data = filterdf_test) %>%
  bind_cols(filterdf_test %>% select(cases))

model_test_predictions_type <- predict(rf_results, new_data = filterdf_test) %>%
  bind_cols(filterdf_test %>% select(cases, geo_value, time_value))

model_test_predictions %>%
  final_metric(truth = cases, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        780.
```

Our model returned an rmse of 820 on our testing data, which is higher than rmse on the training data. This means my model did overfitting to the training data.

```
model_test_predictions %>%
group_by(geo_value , time_value) %>%
ggplot(aes(x = time_value, y = .pred, colour=factor(county_fips_to_name(geo_value))))+
  geom_point() + geom_line()
```



Forecasting the timeseries approaches

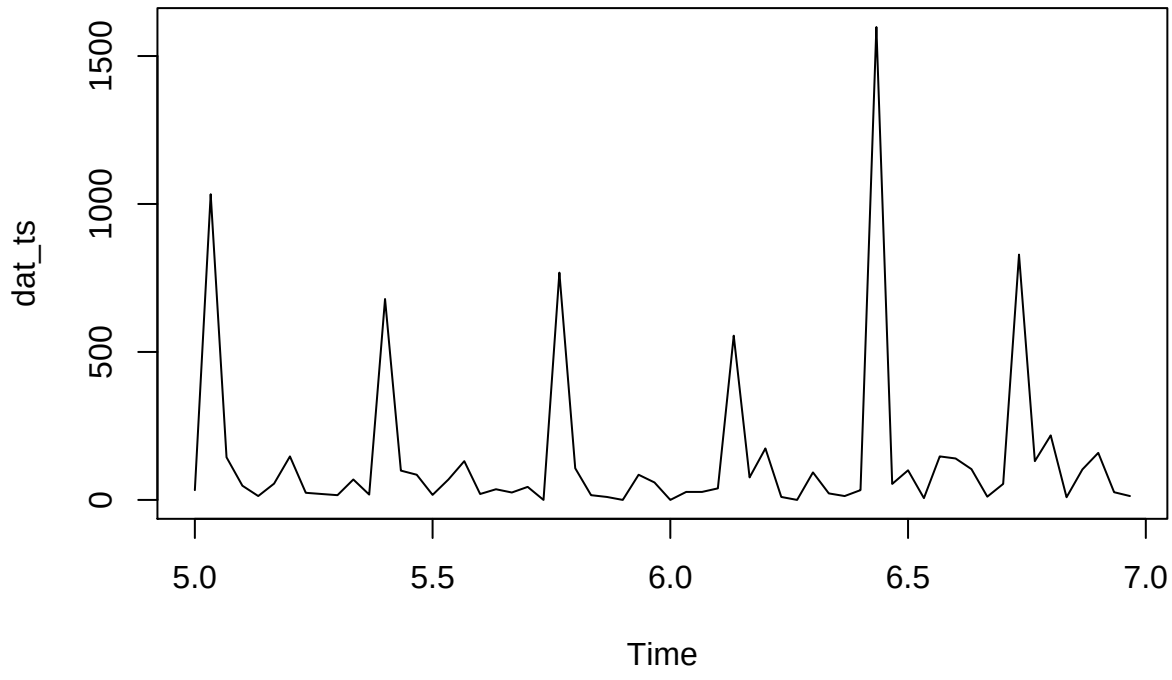
There are many available forecasting statistics model in forecast package that can be used to predict the future days of covid predictions. Below is the test resaults for next 20days of prediction using the simple exponential smoothing model.

```
dat_train<-filterdf_train
dat_test<-filterdf_test

library(TSstudio)
library(forecast)

nrow(dat_train); nrow(dat_test)
```

```
## [1] 577
```

```
## [1] 453
```

```
dat_ts <- ts(dat_train[, 8], start = c(5,1), end = c(6,30), frequency = 30)
plot.ts(dat_ts)
```




```
ts_info(dat_ts)
```

```
##  The dat_ts series is a ts object with 1 variable and 60 observations
```

```
##  Frequency: 30
##  Start time: 5 1
##  End time: 6 30
```

```
#lines 2 to 4
se_model <-  ses(dat_ts, h = 20)
summary(se_model)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = dat_ts, h = 20)
##
##   Smoothing parameters:
##     alpha = 1e-04
##
##   Initial states:
##     l = 143.6152
##
##   sigma:  286.5717
##
##      AIC     AICc      BIC
## 928.5852 929.0138 934.8683
##
## Error measures:
##                      ME    RMSE      MAE  MPE MAPE      MASE        ACF1
## Training set 0.4097984 281.755 157.1839 -Inf  Inf 0.7357649 -0.08126098
##
## Forecasts:
##          Point Forecast     Lo 80   Hi 80     Lo 95    Hi 95
## 7.000000       143.6176 -223.6388 510.874 -418.0525 705.2878
## 7.033333       143.6176 -223.6388 510.874 -418.0525 705.2878
## 7.066667       143.6176 -223.6388 510.874 -418.0525 705.2878
## 7.100000       143.6176 -223.6388 510.874 -418.0525 705.2878
## 7.133333       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.166667       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.200000       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.233333       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.266667       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.300000       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.333333       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.366667       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.400000       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.433333       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.466667       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.500000       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.533333       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.566667       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.600000       143.6176 -223.6388 510.874 -418.0526 705.2878
## 7.633333       143.6176 -223.6388 510.874 -418.0526 705.2878
```

SE model results in RMSE=281 very close to actual training data set., Below plot shows the next 20 days of covid prediction for the cases in state of california.

```
autoplot(se_model)
```