final_covid

Willis Banks

2023-12-06

Data Source

The data below comes from the NYPD historical shooting incidents, as publicly available in the link below. The data is separated by file into United States and Global confirmed cases of COVID-19 and confirmed deaths due to COVID-19. Additionally, there is a reference table included for looking up things such as country population.

```
dataURL_US_confirmed <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19
dataURL_US_deaths <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_da
dataURL_Global_confirmed <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covi
dataURL_Global_deaths <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_1
dataURL_Global_recovered<- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid
dataURL_lookup <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/

us_confirmed_raw <- read_csv(dataURL_US_confirmed,show_col_types = FALSE)
us_deaths_raw <- read_csv(dataURL_US_deaths,show_col_types = FALSE)
gl_confirmed_raw <- read_csv(dataURL_Global_confirmed,show_col_types = FALSE)
gl_deaths_raw <- read_csv(dataURL_Global_deaths,show_col_types = FALSE)
gl_recovered_raw <- read_csv(dataURL_Global_recovered,show_col_types = FALSE)
lookup_raw <- read_csv(dataURL_lookup,show_col_types = FALSE)
```

Cleaning

Several steps to clean the data are performed. Ultimately, the four data sets for US/Global cases/deaths are joined together into two tables, one for both global and US data sets.

- For the global data
  - First, the table is pivoted such that each row corresponds to a date/location pair and the associated data, including the new cases and deaths values
  - Latitude and longitude are removed as irrelevant to the analysis
  - The two tables are joined together
  - Province/State and Country/Region are joined to create CombinedKey
  - Country/region and province/state are renamed to avoid using irregular characters
  - Date is cast as a datetime variable
  - Country population is joined to the table from the reference table
  - The parameters of interest are selected explicitly to create the final table
- For the US data
  - Similarly to global data, the tables are pivoted to create cases/deaths per day
  - Date is cast as a datetime variable

  - Latitude and longitude are dropped
  - The two table are joined into the final USA data table

```
gl_confirmed <- gl_confirmed_raw %>% pivot_longer(cols = -c('Province/State','Country/Region',Lat,Long)
gl_deaths <- gl_deaths_raw %>% pivot_longer(cols = -c('Province/State','Country/Region',Lat,Long),names_
global <- gl_confirmed %>% full_join(gl_deaths) %>% rename(Country_Region = 'Country/Region', Province_S
```

```
## Joining with `by = join_by(`Province/State`, `Country/Region`, date)`
```

```
global <- global %>% filter(cases > 0) %>% unite("CombinedKey",c(Province_State,Country_Region),sep=", "
global <- global %>% left_join(lookup_raw, by = c("Province_State","Country_Region")) %>% select(-c(UID

us_confirmed <- us_confirmed_raw %>% pivot_longer(cols = -(UID:Combined_Key),names_to = 'date', values_t
us_deaths <- us_deaths_raw %>% pivot_longer(cols = -(UID:Population),names_to = 'date', values_to = 'dea
us <- us_confirmed %>% full_join(us_deaths)
```

```
## Joining with `by = join_by(Admin2, Province_State, Country_Region,
## Combined_Key, date)`
```

Visualizations

The next series of cells generate visualizations regarding the propagation and lethality of COVID-19 in the USA as well as the state of Missouri.

The first two graphs show the spread of COVID-19 on a logarithic scale, as well as the lethality, as a cumulative value. While interesting, it runs into the issue of log scales flattening. To a layman, it may appear that COVID slowed or stopped when, in reality, it's a matter of scale. Regardless of the size of the entire US versus the size of the state of Missouri, both plots exhibit this problem. This is not to say these plots are not useful, but it highlights the drawbacks of using such a plot.

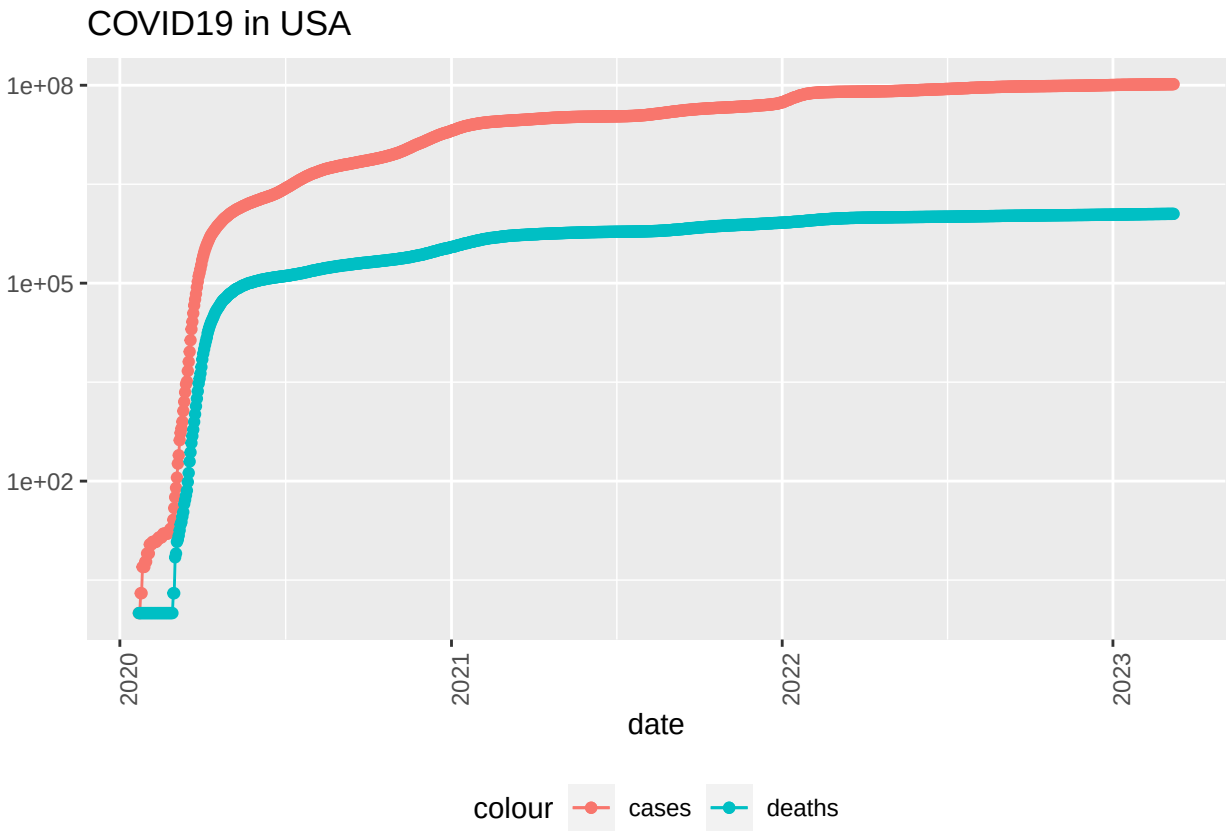
```
us_statewise <- us %>% group_by(Province_State, Country_Region, date) %>% summarise(cases=sum(cases),dea
```

```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can
## override using the `.groups` argument.
```

```
us_total <- us %>% group_by(Country_Region,date) %>% summarize(cases=sum(cases),deaths=sum(deaths),Popul
```

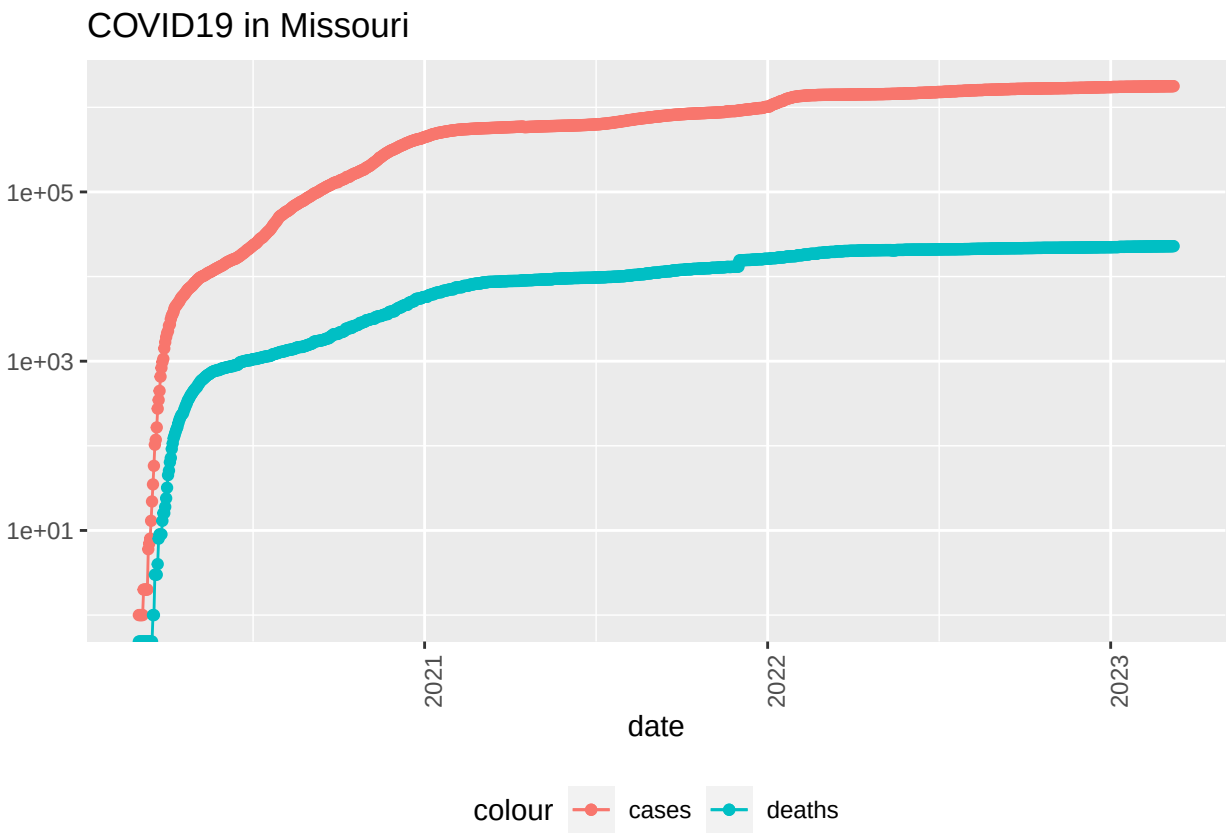
```
## `summarise()` has grouped output by 'Country_Region'. You can override using
## the `.groups` argument.
```

```
us_total %>% filter(cases>0) %>% ggplot(aes(x=date,y=cases)) + geom_line(aes(color='cases')) + geom_poi
```




```
state <- 'Missouri'
us_statewise %>% filter(Province_State == state) %>%  filter(cases>0) %>% ggplot(aes(x=date,y=cases)) +
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```



To address this issue, the second set of plots was created. While it does remain on a log scale, the new plots show only new cases and deaths on any given day. Because the scale is so much smaller, the flattening of the data isn't seen. Because the specific number of how many new cases/deaths on a given day varies greatly, the nice clean line of the former plots vanishes and the new plot appears messier. Though, if you look closely at certain parts of the data, information can be inferred about reporting methods. If you look to the Missouri plot, you'll see that in 2022, there begins a series of regular points that have a smooth curve to them, while days between them drop to zero. This is due to all the cases and deaths being reported once a week.

```
us_total <- us_total %>% mutate(new_cases = cases-lag(cases),new_deaths=deaths-lag(deaths))
us_total %>% filter(cases>0) %>% ggplot(aes(x=date,y=new_cases)) + geom_line(aes(color='new_cases')) + g
```

```
## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced
```

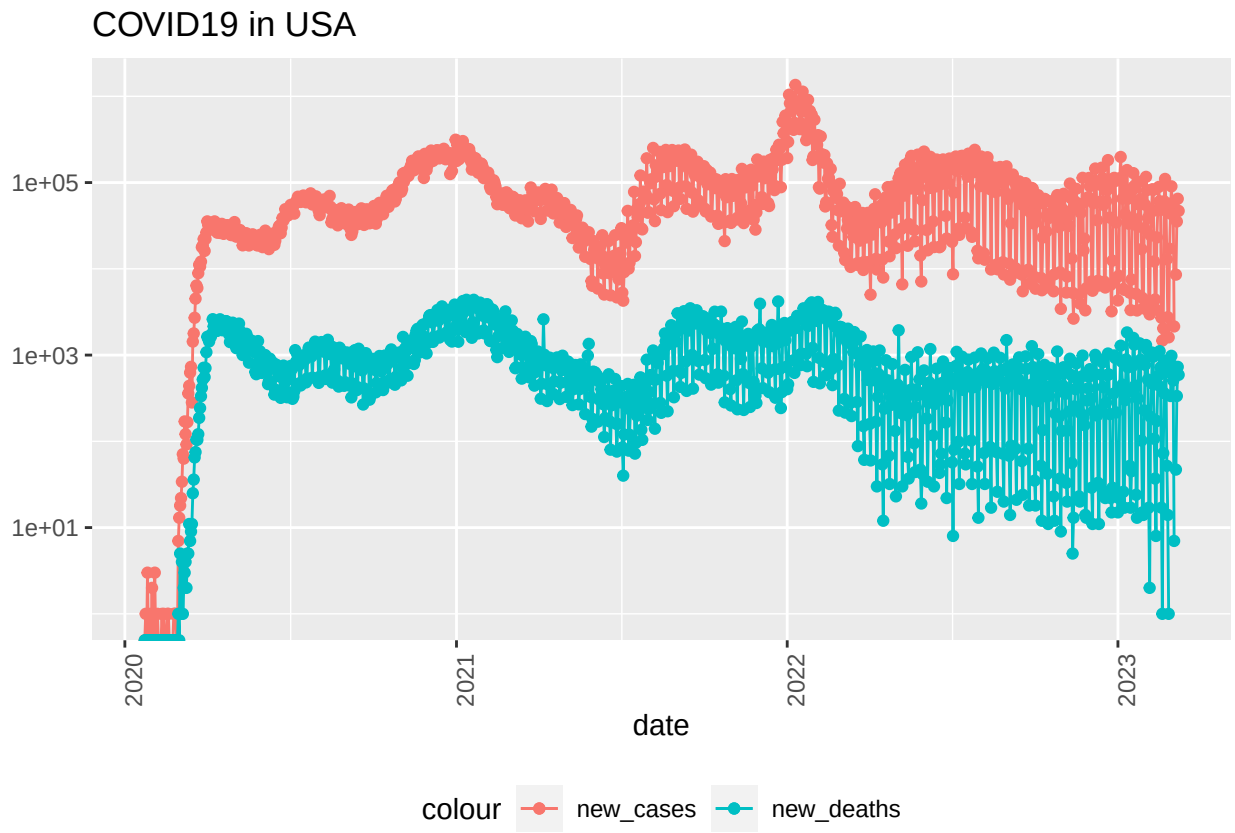
```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 2 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 4 rows containing missing values (`geom_point()`).
```




```
us_statewise <- us_statewise %>% mutate(new_cases = cases-lag(cases),new_deaths=deaths-lag(deaths))
state <- 'Missouri'
us_statewise %>% filter(Province_State == state) %>%  filter(cases>0) %>% ggplot(aes(x=date,y=new_cases
```

```
## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced
```

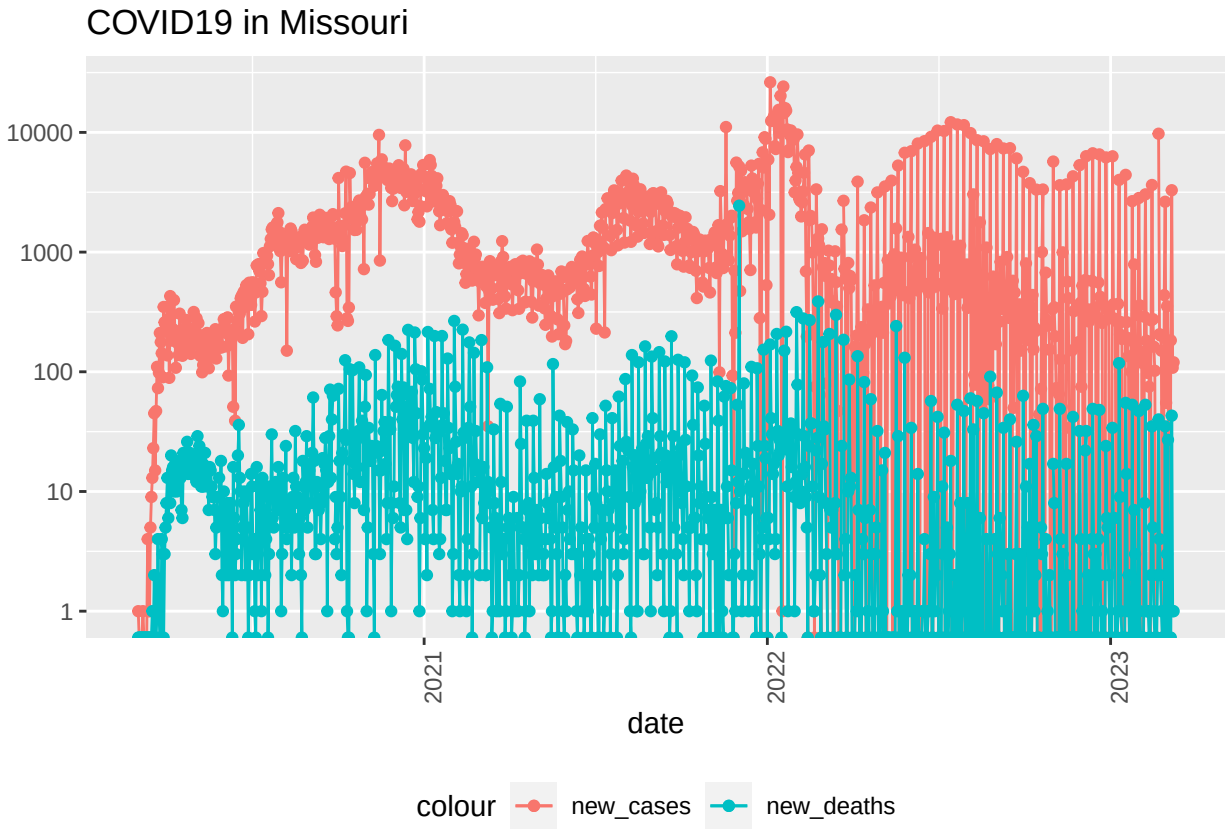
```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 4 rows containing missing values (`geom_point()`).

## Warning: Removed 19 rows containing missing values (`geom_point()`).
```



Data Exploration

To look deeper into the data, two more variables are generated, cases per thousand and deaths per thousand. These are population adjusted values so that one can more accurately compare across different populations. Below are shown the best and worst of the US data, as ordered by deaths per thousand (named dpt in the data).

It is worth noting that, even with adjusted variables, direct comparison is not a silver bullet. American Samoa, the Northern Mariana Islands, and the Virgin Islands are the best on the list, but they are also very small populations compared to the rest of the United Stats, as well as islands. As such, their approaches to the pandemic could more easily reach a greater percentage of their population and they also have better capability to control individuals entering or leaving the areas. As such, even identical approaches executed in American Samoa and Arizona are liable to have very different numbers. As such, you should expect any model to have noise based on difficult to capture features such as these.

```
us_stateTotal <- us_statewise %>% group_by(Province_State) %>% summarize(deaths = max(deaths),cases=max
us_stateTotal %>% slice_min(dpt,n=10)
```

```
## # A tibble: 10 x 6
##    Province_State           deaths   cases population   cpt   dpt
##    <chr>                     <dbl>   <dbl>      <dbl> <dbl> <dbl>
##  1 American Samoa               34    8320      55641  150. 0.611
##  2 Northern Mariana Islands     41   13666      55144  248. 0.744
##  3 Virgin Islands              130   24813     107268  231. 1.21
##  4 Hawaii                     1841  380608    1415872  269. 1.30
##  5 Vermont                     929  152618     623989  245. 1.49
##  6 Puerto Rico                5823 1101469    3754939  293. 1.55
##  7 Utah                       5298 1090346    3205958  340. 1.65
##  8 Alaska                     1486  307655     740995  415. 2.01
##  9 District of Columbia       1432  177945     705749  252. 2.03
## 10 Washington                15683 1928913    7614893  253. 2.06
```

```
us_stateTotal %>% slice_max(dpt,n=10)
```

```
## # A tibble: 10 x 6
##    Province_State deaths   cases population   cpt   dpt
##    <chr>           <dbl>   <dbl>      <dbl> <dbl> <dbl>
##  1 Arizona         33102 2443514    7278717  336.  4.55
##  2 Oklahoma        17972 1290929    3956971  326.  4.54
##  3 Mississippi     13370  990756    2976149  333.  4.49
##  4 West Virginia    7960  642760    1792147  359.  4.44
##  5 New Mexico       9061  670929    2096829  320.  4.32
##  6 Arkansas        13020 1006883    3017804  334.  4.31
##  7 Alabama         21032 1644533    4903185  335.  4.29
##  8 Tennessee       29263 2515130    6829174  368.  4.28
##  9 Michigan        42205 3064125    9986857  307.  4.23
## 10 Kentucky        18130 1718471    4467673  385.  4.06
```

Modeling

Below is a simple linear regression model attempting to predict the value of deaths per thousand based on cases per thousand. While it is obvious that these two things are directly correlated and partially causal, it serves as an example of even a causal relationship to not be the only factor.

Without addressing flawed reporting methods wherein non-COVID related deaths were attributed to it (which would serve as irreducible noise), this model demonstrates that factors beyond simple disease propagation are present that influence how many people died. If that were the only factor, one would expect a straight line. Because this data operates on a large scale, individual health choices become noise that can be disregarded.

To make better use of this model, one approach would be to add new parameters to the model to better predict the deaths per thousand. If, for example, one of those parameters was reasonably controllable, one could use it to reduce the number of deaths during future pandemics.

```
obviousModel <- lm(dpt ~ cpt, data=us_stateTotal)
us_oModel <- us_stateTotal %>% mutate(pred = predict(obviousModel)) %>% mutate(res2 = (pred-dpt)^2)
summary(obviousModel)
```

```
##
## Call:
## lm(formula = dpt ~ cpt, data = us_stateTotal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3352 -0.5978  0.1491  0.6535  1.2086
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.36167    0.72480  -0.499     0.62
## cpt          0.01133    0.00232   4.881 9.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8615 on 54 degrees of freedom
## Multiple R-squared:  0.3061, Adjusted R-squared:  0.2933
## F-statistic: 23.82 on 1 and 54 DF,  p-value: 9.763e-06
```

Shown below is a plot of the predicted values, in red, against the real values, in blue. Also included are green values which represent the squared error of any given data point. It is only by luck that these values were on the same scale as dpt and so can be shown directly. In analysis, such points can be used to rapidly locate irregular values to see if there is an issue with the data or if that data point was significantly different than the rest in a useful way.

```
us_oModel %>% ggplot() + geom_point(aes(x=cpt,y=dpt),color="blue") + geom_point(aes(x=cpt,y=pred),color=
```