Corona_Analysis

Steven Smith, PhD

3/18/2020

Contents

The 2019-2020 Coronavirus Pandemic Analysis

Contact: Smith Research

BACKGROUND & APPROACH

I wanted to track and trend the coronavirus outbreak on my own curiosity. There are some interesting questions that may fall out of this, as it is a very historic moment, including scientifically and analytically (we have a large amount of data being shared across the globe, analyzed in real-time). The world has come to a halt because of it.
This analysis attempts to answer the following questions (more to come):

1. What does the trend of the pandemic look like to date?

2. What are future case predictions based on historical model?
3. What interesting quirks or patterns emerge?

ASSUMPTIONS & LIMITATIONS: * This data is limited by the source. I realized early on that depending on source there were conflicting # of cases. Originally I was using JHU data… but this was always 'ahead' of the Our World In Data. I noticed that JHU's website was buggy- you clicked on the U.S. stats but it didn't reflect the U.S.. So I changed data sources to be more consistent with what is presented in the media (and Our World In Data has more extensive plots I can compare my own to). An interesting aside might be why the discrepancy? Was I missing something?
* Defintiions are important as is the idea that multiple varibales accumulate in things like total cases (more testing for example).

SOURCE RAW DATA: * https://ourworldindata.org/coronavirus
* https://github.com/CSSEGISandData/COVID-19/
*

INPUT DATA LOCATION: github (https://github.com/sbs87/coronavirus/tree/master/data)

OUTPUT DATA LOCATIOn: github (https://github.com/sbs87/coronavirus/tree/master/results)

TIMESTAMP

Start: ##—— Mon Apr 20 00:44:36 2020 ——##

PRE-ANALYSIS

The following sections are outside the scope of the 'analysis' but are still needed to prepare everything

UPSTREAM PROCESSING/ANALYSIS

1. Google Mobility Scraping, script available at get_google_mobility.py

```
# Mobility data has to be extracted from Google PDF reports using a web scraping script (python , writt

# See get_google_mobility.py for local script

python3 get_google_mobility.py
# writes csv file of mobility data as "mobility.csv"
```

SET UP ENVIORNMENT

Load libraries and set global variables

```
# timestamp start
timestamp()
## ##------ Mon Apr 20 00:44:36 2020 ------##

# clear previous enviornment
rm(list = ls())

##------------------------------------------
## LIBRARIES
##------------------------------------------
library(plyr)
library(tidyverse)
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  3.0.0     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
```

```
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
library(ggplot2)
library(reshape2)
##
## Attaching package: 'reshape2'
## The following object is masked from 'package:tidyr':
##
##     smiths
library(plot.utils)
library(utils)
library(knitr)

##------------------------------------------

##------------------------------------------
# GLOBAL VARIABLES
##------------------------------------------
user_name <- Sys.info()["user"]
working_dir <- paste0("/Users/", user_name, "/Projects/coronavirus/")  # don't forget trailing /
results_dir <- paste0(working_dir, "results/")  # assumes diretory exists
results_dir_custom <- paste0(results_dir, "custom/")  # assumes diretory exists


Corona_Cases.source_url <- "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/cs
Corona_Cases.US.source_url <- "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data
Corona_Deaths.US.source_url <- "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_dat
Corona_Deaths.source_url <- "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/c

Corona_Cases.fn <- paste0(working_dir, "data/", basename(Corona_Cases.source_url))
Corona_Cases.US.fn <- paste0(working_dir, "data/", basename(Corona_Cases.US.source_url))
Corona_Deaths.fn <- paste0(working_dir, "data/", basename(Corona_Deaths.source_url))
Corona_Deaths.US.fn <- paste0(working_dir, "data/", basename(Corona_Deaths.US.source_url))
default_theme <- theme_bw() + theme(text = element_text(size = 14))  # fix this
##------------------------------------------
```

FUNCTIONS

List of functions

function_name	description
prediction_model	outputs case estumate for given log-linear moder parameters slope and intercept
make_long	converts input data to long format (specialized cases)

function_name	description
name_overlaps	outputs the column names intersection and set diffs of two data frame
find_linear_index	finds the first date at which linearaity occurs

```
##-------------------------------------------
## FUNCTION: prediction_model
##-------------------------------------------
## --- //// ----
# Takes days vs log10 (case) linear model parameters and a set of days since 100 cases and outputs a da
## --- //// ----
prediction_model<-function(m=1,b=0,days=1){
  total_cases<-m*days+b
  total_cases.log<-log(total_cases,10)
  prediction<-data.frame(days=days,Total_confirmed_cases_perstate=total_cases)
  return(prediction)
}
##-------------------------------------------

##-------------------------------------------
## FUNCTION: make_long
##-------------------------------------------
## --- //// ----
# Takes wide-format case data and converts into long format, using date and total cases as variable/val
## --- //// ----
make_long<-function(data_in,variable.name = "Date",
                   value.name = "Total_confirmed_cases",
                   id.vars=c("case_type","Province.State","Country.Region","Lat","Long","City","Populat

long_data<-melt(data_in,
                id.vars = id.vars,
                variable.name=variable.name,
                value.name=value.name)
return(long_data)

}
##-------------------------------------------

## THIS WILL BE IN UTILS AT SOME POINT
name_overlaps<-function(df1,df2){
i<-intersect(names(df1),
names(df2))
sd1<-setdiff(names(df1),
names(df2))
sd2<-setdiff(names(df2),names(df1))
cat("intersection:\n",paste(i,"\n"))
cat("in df1 but not df2:\n",paste(sd1,"\n"))
cat("in df2 but not df1:\n",paste(sd2,"\n"))
```

```
return(list("int"=i,"sd_1_2"=sd1,"sd_2_1"=sd2))
}

##-------------------------------------------

##-------------------------------------------
## FUNCTION: find_linear_index
##-------------------------------------------
## --- //// ----
# Find date at which total case data is linear (for a given data frame)
## --- //// ----

find_linear_index<-function(tmp,running_avg=5){
  tmp$Total_confirmed_cases_perstate.log<-log(tmp$Total_confirmed_cases_perstate,2)
  derivative<-data.frame(matrix(nrow = nrow(tmp),ncol = 4))
  names(derivative)<-c("m.time","mm.time","cumsum","date")

  # First derivative
  for(t in 2:nrow(tmp)){
    slope.t<- tmp[t,"Total_confirmed_cases_perstate.log"]- tmp[t-1,"Total_confirmed_cases_perstate.log"]
    derivative[t,"m.time"]<-slope.t
    derivative[t,"date"]<-as.Date(tmp[t,"Date"])
  }

  # Second derivative
  for(t in 2:nrow(derivative)){
    slope.t<- derivative[t,"m.time"]- derivative[t-1,"m.time"]
    derivative[t,"mm.time"]<-slope.t
  }

  #Compute running sum of second derivative (window = 5). Choose point at which within 0.2
  for(t in running_avg:nrow(derivative)){
    slope.t<- sum(abs(derivative[t:(t-4),"mm.time"])<0.2,na.rm = T)
    derivative[t,"cumsum"]<-slope.t
  }

  #Find date -5 from the stablility point
  linear_begin<-min(derivative[!is.na(derivative$cumsum) & derivative$cumsum==running_avg,"date"])-runn

  return(linear_begin)
}
```

READ IN DATA

- total number of cases. current source: https://github.com/CSSEGISandData (precvious source https://ourworldindata.org/coronavirus)

```
# Q: do we want to archive previous versions? Maybe an auto git mv?

##-------------------------------------------
## Download and read in latest data from github
##-------------------------------------------
download.file(Corona_Cases.source_url, destfile = Corona_Cases.fn)
Corona_Totals.raw <- read.csv(Corona_Cases.fn, header = T, stringsAsFactors = F)
```

```r
download.file(Corona_Cases.US.source_url, destfile = Corona_Cases.US.fn)
Corona_Totals.US.raw <- read.csv(Corona_Cases.US.fn, header = T, stringsAsFactors = F)

download.file(Corona_Deaths.source_url, destfile = Corona_Deaths.fn)
Corona_Deaths.raw <- read.csv(Corona_Deaths.fn, header = T, stringsAsFactors = F)

download.file(Corona_Deaths.US.source_url, destfile = Corona_Deaths.US.fn)
Corona_Deaths.US.raw <- read.csv(Corona_Deaths.US.fn, header = T, stringsAsFactors = F)

# latest date on all data:
paste("US deaths:", names(Corona_Deaths.US.raw)[ncol(Corona_Deaths.US.raw)])
```

```
## [1] "US deaths: X4.19.20"
```

```r
paste("US total:", names(Corona_Totals.US.raw)[ncol(Corona_Totals.US.raw)])
```

```
## [1] "US total: X4.19.20"
```

```r
paste("World deaths:", names(Corona_Deaths.raw)[ncol(Corona_Deaths.raw)])
```

```
## [1] "World deaths: X4.19.20"
```

```r
paste("World total:", names(Corona_Totals.raw)[ncol(Corona_Totals.raw)])
```

```
## [1] "World total: X4.19.20"
```

PROCESS DATA

- Convert to long format
- Fix date formatting/convert to numeric date
- Log10 transform total # cases

```r
##-------------------------------------------
## Combine death and total data frames
##-------------------------------------------
Corona_Totals.raw$case_type<-"total"
Corona_Totals.US.raw$case_type<-"total"
Corona_Deaths.raw$case_type<-"death"
Corona_Deaths.US.raw$case_type<-"death"

# for some reason, Population listed in US death file but not for other data... Weird. When combining,
Corona_Totals.US.raw$Population<-"NA"
Corona_Totals.raw$Population<-"NA"
Corona_Deaths.raw$Population<-"NA"

Corona_Cases.raw<-rbind(Corona_Totals.raw,Corona_Deaths.raw)
Corona_Cases.US.raw<-rbind(Corona_Totals.US.raw,Corona_Deaths.US.raw)
#TODO: custom utils- setdiff, intersect names... option to output in merging too
##-------------------------------------------
# prepare raw datasets for eventual combining
##-------------------------------------------
Corona_Cases.raw$City<-"NA" # US-level data has Cities
Corona_Cases.US.raw$Country_Region<-"US_state" # To differentiate from World-level stats
```

```
Corona_Cases.US.raw<-plyr::rename(Corona_Cases.US.raw,c("Province_State"="Province.State",
                                                   "Country_Region"="Country.Region",
                                                   "Long_"="Long",
                                                   "Admin2"="City"))



##--------------------------------------------
## Convert to long format
##--------------------------------------------
#JHU has a gross file format. It's in wide format with each column is the date in MM/DD/YY. So read thi
# Furthermore, the World and US level data is formatted differently, containing different columns, etc.

Corona_Cases.long<-rbind(make_long(select(Corona_Cases.US.raw,-c(UID,iso2,iso3,code3,FIPS,Combined_Key)
make_long(Corona_Cases.raw))



##--------------------------------------------
## Fix date formatting, convert to numeric date
##--------------------------------------------
Corona_Cases.long$Date<-gsub(Corona_Cases.long$Date,pattern = "^X",replacement = "0") # leading 0 read
Corona_Cases.long$Date<-gsub(Corona_Cases.long$Date,pattern = "20$",replacement = "2020") # ends in .20
Corona_Cases.long$Date<-as.Date(Corona_Cases.long$Date,format = "%m.%d.%y")
Corona_Cases.long$Date.numeric<-as.numeric(Corona_Cases.long$Date)

kable(table(select(Corona_Cases.long,c("Country.Region","case_type"))),caption = "Number of death and t
```

Table 2: Number of death and total case longitudinal datapoints per geographical region

	death	total
Afghanistan	89	89
Albania	89	89
Algeria	89	89
Andorra	89	89
Angola	89	89
Antigua and Barbuda	89	89
Argentina	89	89
Armenia	89	89
Australia	712	712
Austria	89	89
Azerbaijan	89	89
Bahamas	89	89
Bahrain	89	89
Bangladesh	89	89
Barbados	89	89
Belarus	89	89
Belgium	89	89
Belize	89	89
Benin	89	89
Bhutan	89	89
Bolivia	89	89
Bosnia and Herzegovina	89	89
Botswana	89	89

	death	total
Brazil	89	89
Brunei	89	89
Bulgaria	89	89
Burkina Faso	89	89
Burma	89	89
Burundi	89	89
Cabo Verde	89	89
Cambodia	89	89
Cameroon	89	89
Canada	1335	1335
Central African Republic	89	89
Chad	89	89
Chile	89	89
China	2937	2937
Colombia	89	89
Congo (Brazzaville)	89	89
Congo (Kinshasa)	89	89
Costa Rica	89	89
Cote d'Ivoire	89	89
Croatia	89	89
Cuba	89	89
Cyprus	89	89
Czechia	89	89
Denmark	267	267
Diamond Princess	89	89
Djibouti	89	89
Dominica	89	89
Dominican Republic	89	89
Ecuador	89	89
Egypt	89	89
El Salvador	89	89
Equatorial Guinea	89	89
Eritrea	89	89
Estonia	89	89
Eswatini	89	89
Ethiopia	89	89
Fiji	89	89
Finland	89	89
France	979	979
Gabon	89	89
Gambia	89	89
Georgia	89	89
Germany	89	89
Ghana	89	89
Greece	89	89
Grenada	89	89
Guatemala	89	89
Guinea	89	89
Guinea-Bissau	89	89
Guyana	89	89
Haiti	89	89
Holy See	89	89

	death	total
Honduras	89	89
Hungary	89	89
Iceland	89	89
India	89	89
Indonesia	89	89
Iran	89	89
Iraq	89	89
Ireland	89	89
Israel	89	89
Italy	89	89
Jamaica	89	89
Japan	89	89
Jordan	89	89
Kazakhstan	89	89
Kenya	89	89
Korea, South	89	89
Kosovo	89	89
Kuwait	89	89
Kyrgyzstan	89	89
Laos	89	89
Latvia	89	89
Lebanon	89	89
Liberia	89	89
Libya	89	89
Liechtenstein	89	89
Lithuania	89	89
Luxembourg	89	89
Madagascar	89	89
Malawi	89	89
Malaysia	89	89
Maldives	89	89
Mali	89	89
Malta	89	89
Mauritania	89	89
Mauritius	89	89
Mexico	89	89
Moldova	89	89
Monaco	89	89
Mongolia	89	89
Montenegro	89	89
Morocco	89	89
Mozambique	89	89
MS Zaandam	89	89
Namibia	89	89
Nepal	89	89
Netherlands	445	445
New Zealand	89	89
Nicaragua	89	89
Niger	89	89
Nigeria	89	89
North Macedonia	89	89
Norway	89	89

	death	total
Oman	89	89
Pakistan	89	89
Panama	89	89
Papua New Guinea	89	89
Paraguay	89	89
Peru	89	89
Philippines	89	89
Poland	89	89
Portugal	89	89
Qatar	89	89
Romania	89	89
Russia	89	89
Rwanda	89	89
Saint Kitts and Nevis	89	89
Saint Lucia	89	89
Saint Vincent and the Grenadines	89	89
San Marino	89	89
Sao Tome and Principe	89	89
Saudi Arabia	89	89
Senegal	89	89
Serbia	89	89
Seychelles	89	89
Sierra Leone	89	89
Singapore	89	89
Slovakia	89	89
Slovenia	89	89
Somalia	89	89
South Africa	89	89
South Sudan	89	89
Spain	89	89
Sri Lanka	89	89
Sudan	89	89
Suriname	89	89
Sweden	89	89
Switzerland	89	89
Syria	89	89
Taiwan*	89	89
Tanzania	89	89
Thailand	89	89
Timor-Leste	89	89
Togo	89	89
Trinidad and Tobago	89	89
Tunisia	89	89
Turkey	89	89
Uganda	89	89
Ukraine	89	89
United Arab Emirates	89	89
United Kingdom	979	979
Uruguay	89	89
US	89	89
US_state	290229	290229
Uzbekistan	89	89

	death	total
Venezuela	89	89
Vietnam	89	89
West Bank and Gaza	89	89
Western Sahara	89	89
Yemen	89	89
Zambia	89	89
Zimbabwe	89	89

```
# Decouple population and lat/long data, refactor to make it more tidy
metadata_columns<-c("Lat","Long","Population")
metadata<-unique(select(filter(Corona_Cases.long,case_type=="death"),c("Country.Region","Province.State
Corona_Cases.long<-select(Corona_Cases.long,-all_of(metadata_columns))

# Some counties are not summarized on the country level. collapse all but US
Corona_Cases.long<-rbind.fill(ddply(filter(Corona_Cases.long,!Country.Region=="US_state"),c("case_type"

# Put total case and deaths side-by-side (wide)
Corona_Cases<-spread(Corona_Cases.long,key = case_type,value = Total_confirmed_cases)

#Compute moratlity rate
Corona_Cases$mortality_rate<-Corona_Cases$death/Corona_Cases$total

#TMP
Corona_Cases<-plyr::rename(Corona_Cases,c("total"="Total_confirmed_cases","death"="Total_confirmed_deat

##------------------------------------------
## log10 transform total # cases
##------------------------------------------
Corona_Cases$Total_confirmed_cases.log<-log(Corona_Cases$Total_confirmed_cases,10)
Corona_Cases$Total_confirmed_deaths.log<-log(Corona_Cases$Total_confirmed_deaths,10)
##------------------------------------------

##------------------------------------------
## Compute # of days since 100th for US data
##------------------------------------------

# Find day that 100th case was found for Country/Province. NOTE: Non US countries may have weird provin
# TODO: consider city-level summary as well. This data may be sparse

Corona_Cases<-merge(Corona_Cases,ddply(filter(Corona_Cases,Total_confirmed_cases>100),c("Country.Region
Corona_Cases$Days_since_100<-Corona_Cases$Date.numeric-Corona_Cases$case100_date

##------------------------------------------
## Add population and lat/long data (CURRENTLY US ONLY)
##------------------------------------------

kable(filter(metadata,(is.na(Country.Region) | is.na(Population) )) %>% select(c("Country.Region","Provi
```

Table 3: Regions for which either population or Country is NA

Country.Region	Province.State	City

```
# Drop missing data
metadata<-filter(metadata,!(is.na(Country.Region) | is.na(Population) ))
# Convert remaining pop to numeric
metadata$Population<-as.numeric(metadata$Population)
```

```
## Warning: NAs introduced by coercion
```

```
# Add metadata to cases
Corona_Cases<-merge(Corona_Cases,metadata,all.x = T)

##-------------------------------------------
## Compute total and death cases relative to population
##-------------------------------------------

Corona_Cases$Total_confirmed_cases.per100<-100*Corona_Cases$Total_confirmed_cases/Corona_Cases$Populatio
Corona_Cases$Total_confirmed_deaths.per100<-100*Corona_Cases$Total_confirmed_deaths/Corona_Cases$Popula


##-------------------------------------------
## Filter df for US state-wide stats
##-------------------------------------------

Corona_Cases.US_state<-filter(Corona_Cases,Country.Region=="US_state" & Total_confirmed_cases>0 )
kable(table(select(Corona_Cases.US_state,c("Province.State"))),caption = "Number of longitudinal datapoi
```

Table 4: Number of longitudinal datapoints (total/death) per state

Var1	Freq
Alabama	1787
Alaska	266
Arizona	521
Arkansas	1717
California	2087
Colorado	1673
Connecticut	304
Delaware	115
Diamond Princess	34
District of Columbia	35
Florida	2053
Georgia	4244
Grand Princess	35
Guam	35
Hawaii	187
Idaho	808
Illinois	2131
Indiana	2513
Iowa	1955
Kansas	1431
Kentucky	2259

Var1	Freq
Louisiana	1875
Maine	466
Maryland	785
Massachusetts	567
Michigan	2105
Minnesota	1788
Mississippi	2308
Missouri	2152
Montana	709
Nebraska	952
Nevada	322
New Hampshire	345
New Jersey	819
New Mexico	656
New York	1945
North Carolina	2609
North Dakota	699
Northern Mariana Islands	20
Ohio	2317
Oklahoma	1562
Oregon	944
Pennsylvania	1921
Puerto Rico	35
Rhode Island	202
South Carolina	1397
South Dakota	931
Tennessee	2474
Texas	4551
Utah	512
Vermont	450
Virgin Islands	35
Virginia	3064
Washington	1357
West Virginia	960
Wisconsin	1679
Wyoming	514

```
Corona_Cases.US_state<-merge(Corona_Cases.US_state,ddply(filter(Corona_Cases.US_state,Total_confirmed_ca
Corona_Cases.US_state$Days_since_100_state<-Corona_Cases.US_state$Date.numeric-Corona_Cases.US_state$cas
```

ANALYSIS

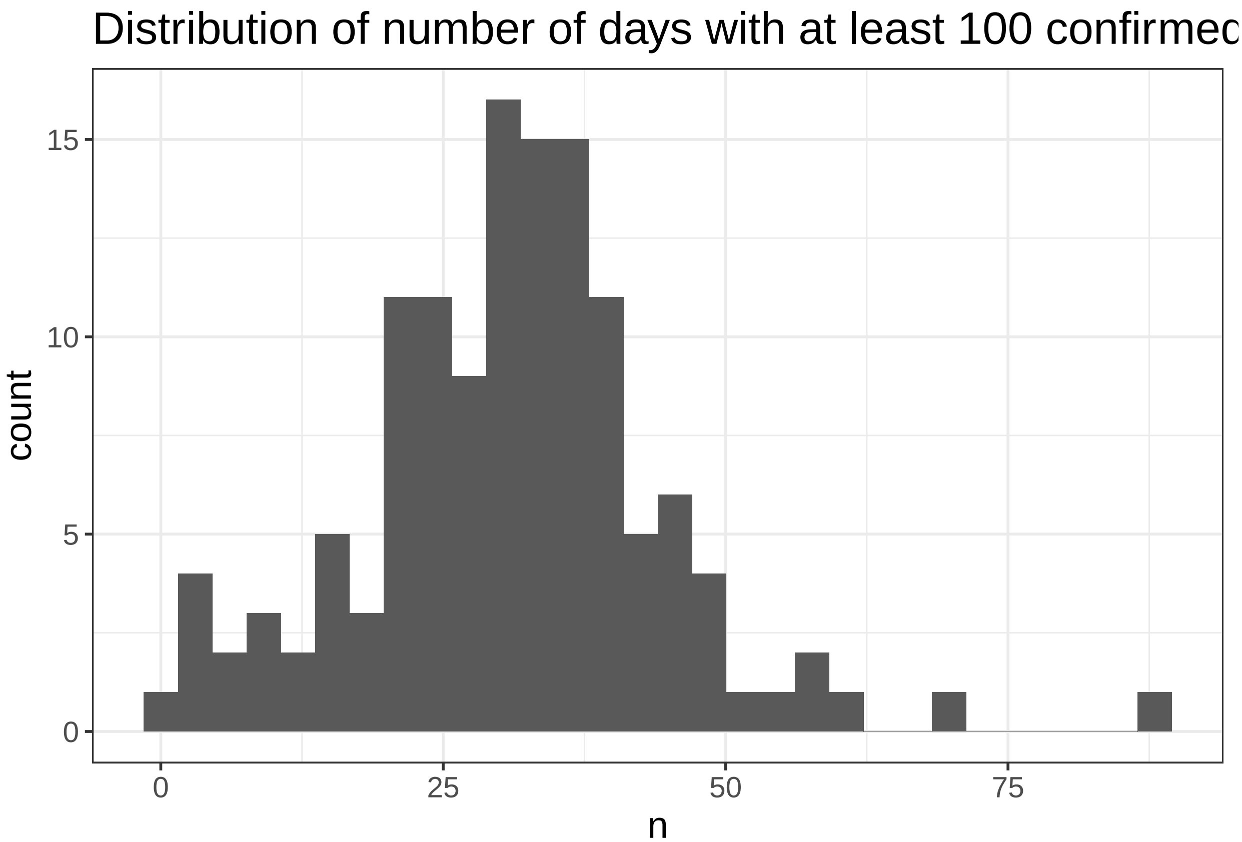
Q1: What is the trend in cases, mortality across geopgraphical regions?

Plot # of cases vs time
* For each geographical set:
* comparative longitudinal case trend (absolute & log scale)
* comparative longitudinal mortality trend
* death vs total correlation

question	dataset	x	y	color	facet	pch	dimentions
comparative_longitudinal_case_log_trend	**long**	time	log cases	geography	none (case type?)	case_type	[15, 50, 4] geography x (2 scale?) case type
comparative longitudinal case trend	long	time	cases	geography	case_type	?	[15, 50, 4] geography x (2+ scale) case type
comparative longitudinal mortality trend	wide	time	mortality rate	geography	none	none	[15, 50, 4] geography
death vs total correlation	wide	cases	deaths	geography	none	none	[15, 50, 4] geography

```r
# total cases vs time
# death cases vs time
# mortality rate vs time
# death vs mortality


  # death vs mortality
  # total & death case vs time (same plot)

#<question> <x> <y> <colored> <facet> <dataset>
## trend in case/deaths over time, comapred across regions <time> <log cases> <geography*> <none> <.wide
## trend in case/deaths over time, comapred across regions <time> <cases> <geography*> <case_type> <.lor
## trend in mortality rate over time, comapred across regions <time> <mortality rate> <geography*> <none
## how are death/mortality related/correlated? <time> <log cases> <geography*> <none>
## how are death and case load correlated? <cases> <deaths>

# lm for each?? - > apply lm from each region starting from 100th case. m, b associated with each.
    # input: geographical regsion, logcase vs day (100th case)
    # output: m, b for each geographical region ID



#total/death on same plot-  differ by 2 logs, so when plotting log, use pch. when plotting absolute, n
#when plotting death and case on same, melt.

#CoronaCases - > filter sets (3)
  #world - choose countries with sufficent data

N<-ddply(filter(Corona_Cases,Total_confirmed_cases>100),c("Country.Region"),summarise,n=length(Country.R
ggplot(filter(N,n<100),aes(x=n))+
  geom_histogram()+
  default_theme+
  ggtitle("Distribution of number of days with at least 100 confirmed cases for each region")
```

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




```
kable(arrange(N,-n),caption="Sorted number of days with at least 100 confirmed cases")
```

Table 6: Sorted number of days with at least 100 confirmed cases

Country.Region	n
US_state	8792
China	89
Diamond Princess	70
Korea, South	60
Japan	59
Italy	57
Iran	54
Singapore	51
France	50
Germany	50
Spain	49
US	48
Switzerland	46
United Kingdom	46
Belgium	45
Netherlands	45
Norway	45
Sweden	45
Austria	43
Malaysia	42
Australia	41
Bahrain	41

Country.Region	n
Denmark	41
Canada	40
Qatar	40
Iceland	39
Brazil	38
Czechia	38
Finland	38
Greece	38
Iraq	38
Israel	38
Portugal	38
Slovenia	38
Egypt	37
Estonia	37
India	37
Ireland	37
Kuwait	37
Philippines	37
Poland	37
Romania	37
Saudi Arabia	37
Indonesia	36
Lebanon	36
San Marino	36
Thailand	36
Chile	35
Pakistan	35
Luxembourg	34
Peru	34
Russia	34
Ecuador	33
Slovakia	33
South Africa	33
United Arab Emirates	33
Armenia	32
Colombia	32
Croatia	32
Mexico	32
Panama	32
Serbia	32
Taiwan*	32
Turkey	32
Argentina	31
Bulgaria	31
Latvia	31
Algeria	30
Costa Rica	30
Dominican Republic	30
Hungary	30
Uruguay	30
Andorra	29
Bosnia and Herzegovina	29

Country.Region	n
Jordan	29
Lithuania	29
Morocco	29
New Zealand	29
North Macedonia	29
Vietnam	29
Albania	28
Cyprus	28
Malta	28
Moldova	28
Brunei	27
Burkina Faso	27
Sri Lanka	27
Tunisia	27
Ukraine	26
Azerbaijan	25
Ghana	25
Kazakhstan	25
Oman	25
Senegal	25
Venezuela	25
Afghanistan	24
Cote d'Ivoire	24
Cuba	23
Mauritius	23
Uzbekistan	23
Cambodia	22
Cameroon	22
Honduras	22
Nigeria	22
West Bank and Gaza	22
Belarus	21
Georgia	21
Bolivia	20
Kosovo	20
Kyrgyzstan	20
Montenegro	20
Congo (Kinshasa)	19
Kenya	18
Niger	17
Guinea	16
Rwanda	16
Trinidad and Tobago	16
Paraguay	15
Bangladesh	14
Djibouti	12
El Salvador	11
Guatemala	10
Madagascar	9
Mali	8
Congo (Brazzaville)	5
Jamaica	5

Country.Region	n
Gabon	3
Somalia	3
Tanzania	3
Ethiopia	2
Burma	1

```
# Pick top 15 countries with data
max_colors<-12
# find way to fix this- China has diff provences. Plot doesnt look right...
sufficient_data<-arrange(filter(N,!Country.Region %in% c("US_state", "Diamond Princess")),-n)[1:max_colo
kable(sufficient_data,caption = paste0("Top ",max_colors," countries with sufficient data"))
```

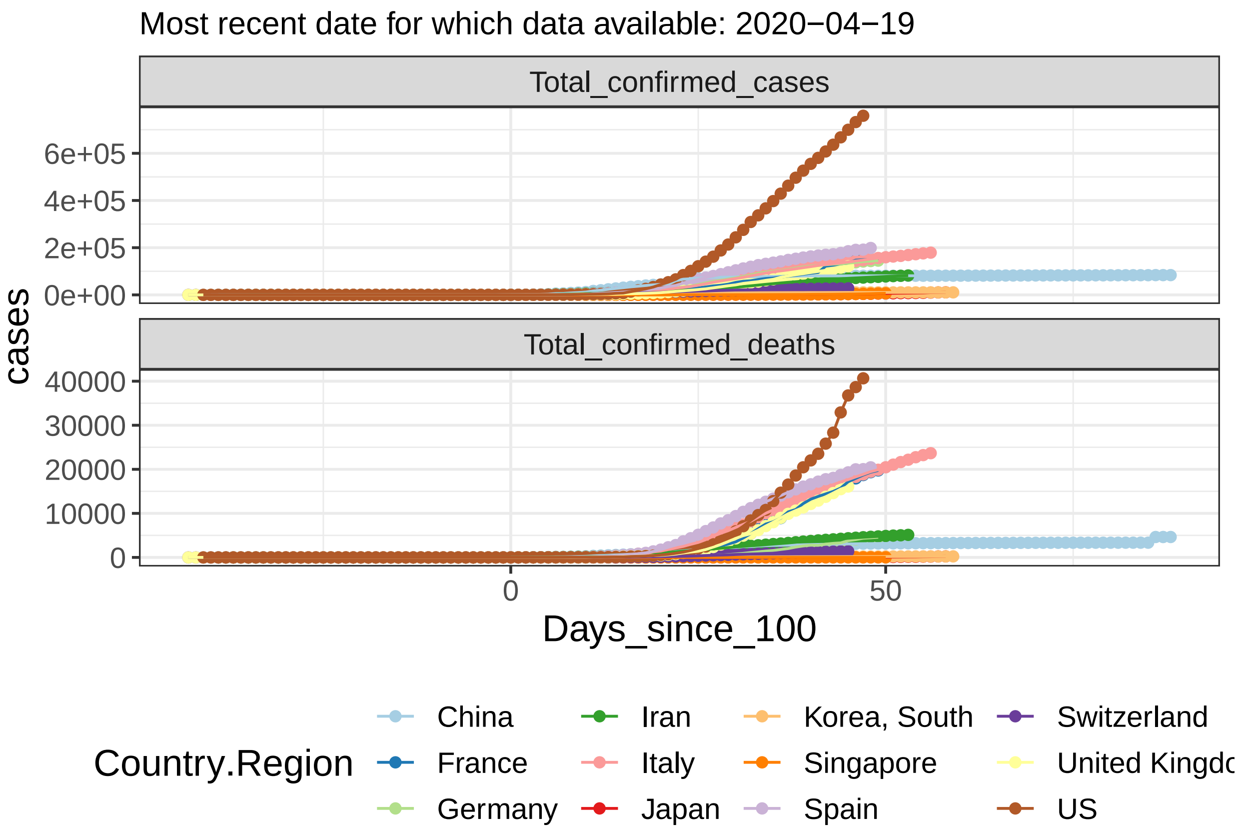
Table 7: Top 12 countries with sufficient data

Country.Region	n
China	89
Korea, South	60
Japan	59
Italy	57
Iran	54
Singapore	51
France	50
Germany	50
Spain	49
US	48
Switzerland	46
United Kingdom	46

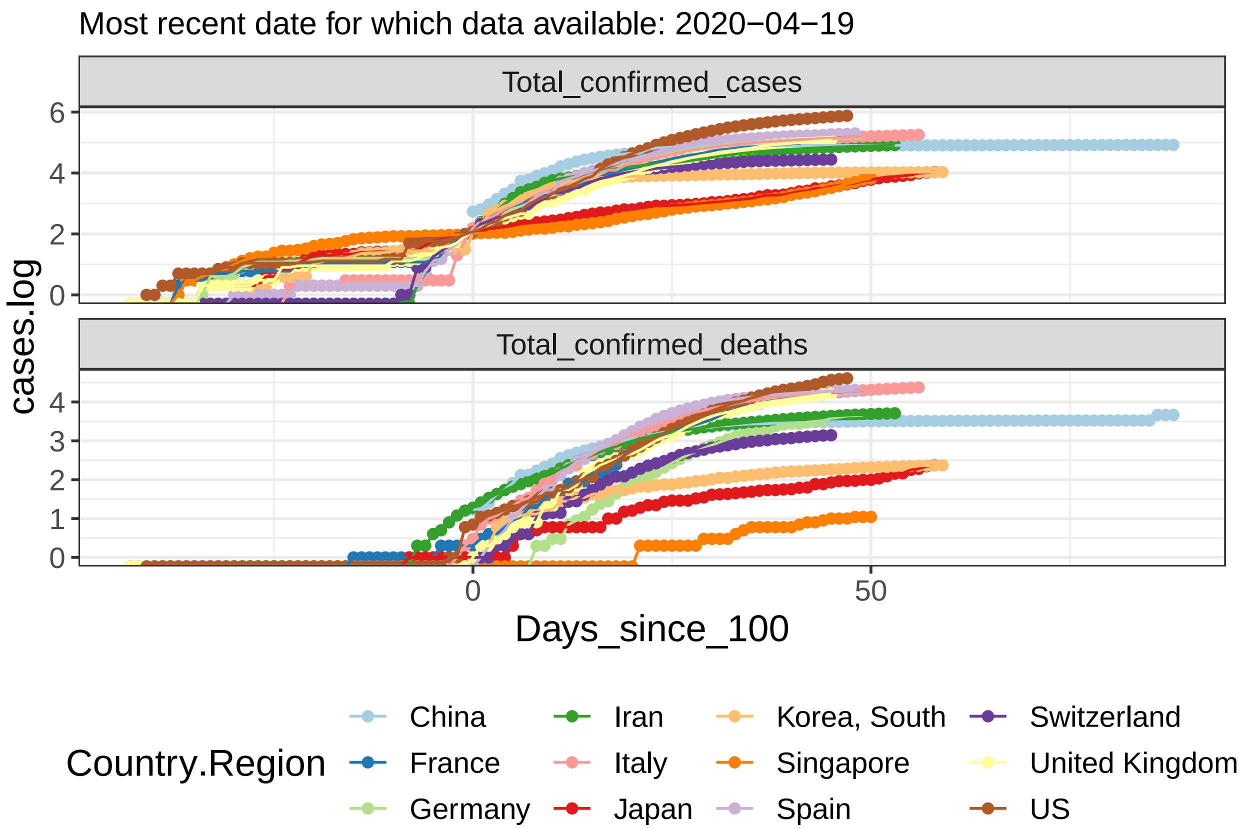
```
Corona_Cases.world<-filter(Corona_Cases,Country.Region %in% c(sufficient_data$Country.Region))



  #us
  #    - by state
Corona_Cases.US<-filter(Corona_Cases,Country.Region=="US" & Total_confirmed_cases>0)
# summarize
#!City %in% c("Unassigned")
  #    - specific cities
#mortality_rate!=Inf & mortality_rate<=1
Corona_Cases.UScity<-filter(Corona_Cases,Province.State %in% c("Pennsylvania","Maryland","New York","New

measure_vars_long<-c("Total_confirmed_cases.log","Total_confirmed_cases","Total_confirmed_deaths","Total
melt_arg_list<-list(variable.name = "case_type",value.name = "cases",measure.vars = c("Total_confirmed_c
melt_arg_list$data=NULL


melt_arg_list$data=select(Corona_Cases.world,-ends_with(match = "log"))
Corona_Cases.world.long<-do.call(melt,melt_arg_list)
melt_arg_list$data=select(Corona_Cases.UScity,-ends_with(match = "log"))
Corona_Cases.UScity.long<-do.call(melt,melt_arg_list)
melt_arg_list$data=select(Corona_Cases.US_state,-ends_with(match = "log"))
```

```r
Corona_Cases.US_state.long<-do.call(melt,melt_arg_list)

Corona_Cases.world.long$cases.log<-log(Corona_Cases.world.long$cases,10)
Corona_Cases.US_state.long$cases.log<-log(Corona_Cases.US_state.long$cases,10)
Corona_Cases.UScity.long$cases.log<-log(Corona_Cases.UScity.long$cases,10)


# what is the current death and total case load for US? For world? For states?
#-absolute
#-log

# what is mortality rate (US, world)
#-absolute

#how is death and case correlated? (US, world)
#-absolute
```

```r
#Corona_Cases.US<-filter(Corona_Cases,Country.Region=="US" & Total_confirmed_cases>0)
#Corona_Cases.US.case100<-filter(Corona_Cases.US, Days_since_100>=0)
# linear model parameters
#(model_fit<-lm(formula = Total_confirmed_cases.log~Days_since_100,data= Corona_Cases.US.case100 ))

#(slope<-model_fit$coefficients[2])
#(intercept<-model_fit$coefficients[1])

# Correlation coefficient
#cor(x = Corona_Cases.US.case100$Days_since_100,y = Corona_Cases.US.case100$Total_confirmed_cases.log)

##------------------------------------------
## Plot World Data
##------------------------------------------
# Timestamp for world
timestamp_plot.world<-paste("Most recent date for which data available:",max(Corona_Cases.world$Date))#


# Base template for plots
baseplot.world<-ggplot(data=NULL,aes(x=Days_since_100,col=Country.Region))+
  default_theme+
  scale_color_brewer(type = "qualitative",palette = "Paired")+
  ggtitle(paste("Log10 cases over time,",timestamp_plot.world))+
  theme(legend.position = "bottom",plot.title = element_text(size=12))


##/////////////////////////
### Plot Longitudinal cases

(Corona_Cases.world.long.plot<-baseplot.world+
    geom_point(data=Corona_Cases.world.long,aes(y=cases))+
    geom_line(data=Corona_Cases.world.long,aes(y=cases))+
    facet_wrap(~case_type,scales = "free_y",ncol=1)+
    ggtitle(timestamp_plot.world)
    )
```



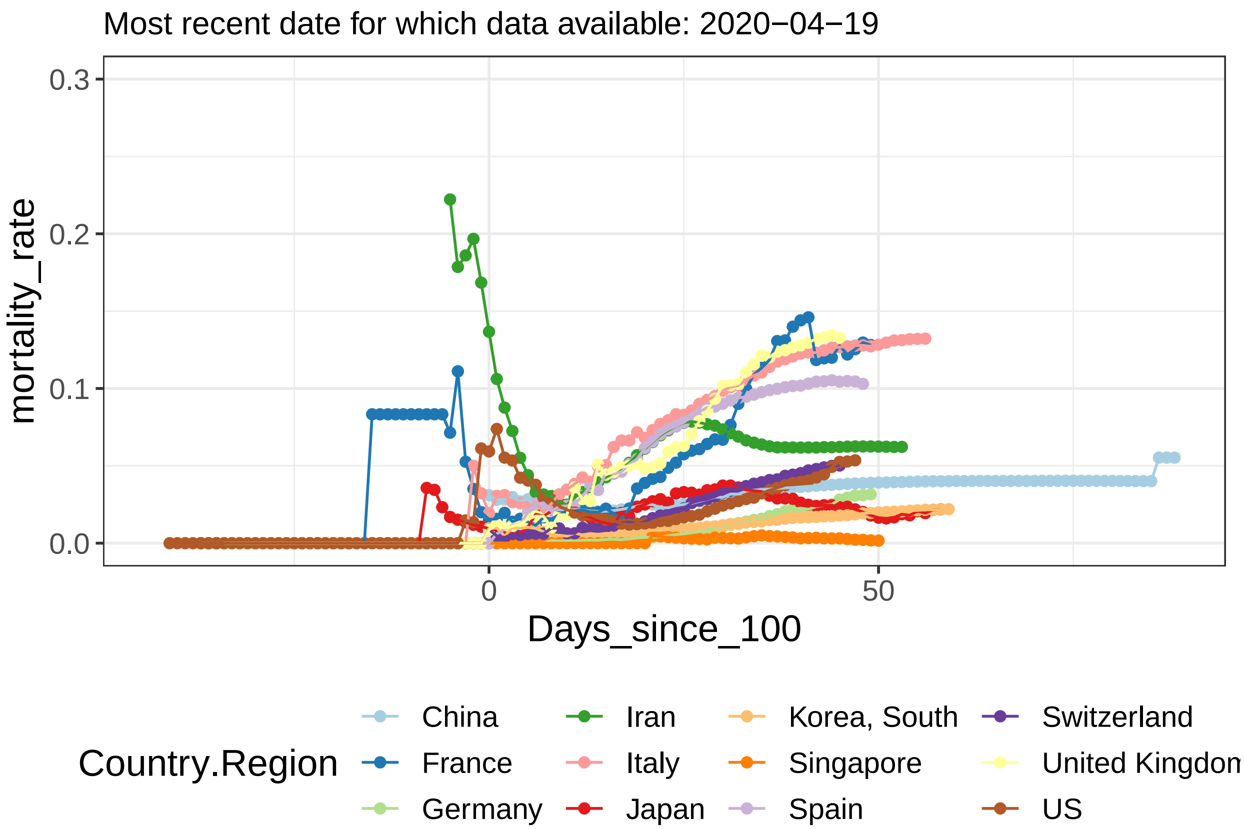

```
(Corona_Cases.world.loglong.plot<-baseplot.world+
    geom_point(data=Corona_Cases.world.long,aes(y=cases.log))+
    geom_line(data=Corona_Cases.world.long,aes(y=cases.log))+
    facet_wrap(~case_type,scales = "free_y",ncol=1)+
    ggtitle(timestamp_plot.world))
```



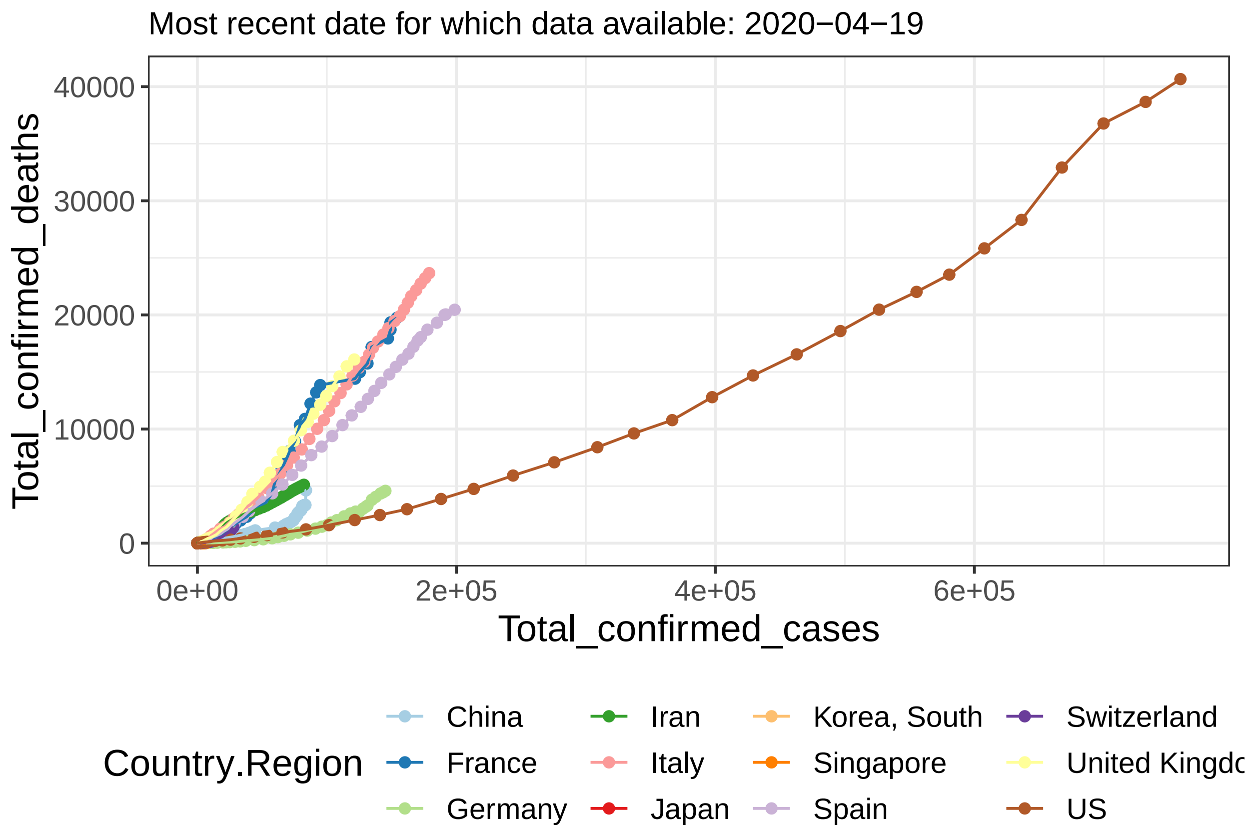

```
##/////////////////////////
### Plot Longitudinal mortality rate

(Corona_Cases.world.mortality.plot<-baseplot.world+
    geom_point(data=Corona_Cases.world,aes(y=mortality_rate))+
    geom_line(data=Corona_Cases.world,aes(y=mortality_rate))+
    ylim(c(0,0.3))+
    ggtitle(timestamp_plot.world))
```

```
## Warning: Removed 100 rows containing missing values (geom_point).
```

```
## Warning: Removed 100 row(s) containing missing values (geom_path).
```




```
##/////////////////////////
### Plot death vs total case correlation

(Corona_Cases.world.casecor.plot<-ggplot(Corona_Cases.world,aes(x=Total_confirmed_cases,y=Total_confirm
  geom_point()+
  geom_line()+
  default_theme+
  scale_color_brewer(type = "qualitative",palette = "Paired")+
  ggtitle(paste("Log10 cases over time,",timestamp_plot.world))+
  theme(legend.position = "bottom",plot.title = element_text(size=12))+
    ggtitle(timestamp_plot.world))
```




```
### Write polots

write_plot(Corona_Cases.world.long.plot,wd = results_dir)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/Corona_Cases.world.long.plot.png"
```

```
write_plot(Corona_Cases.world.loglong.plot,wd = results_dir)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/Corona_Cases.world.loglong.plot.png"
```

```
write_plot(Corona_Cases.world.mortality.plot,wd = results_dir)
```

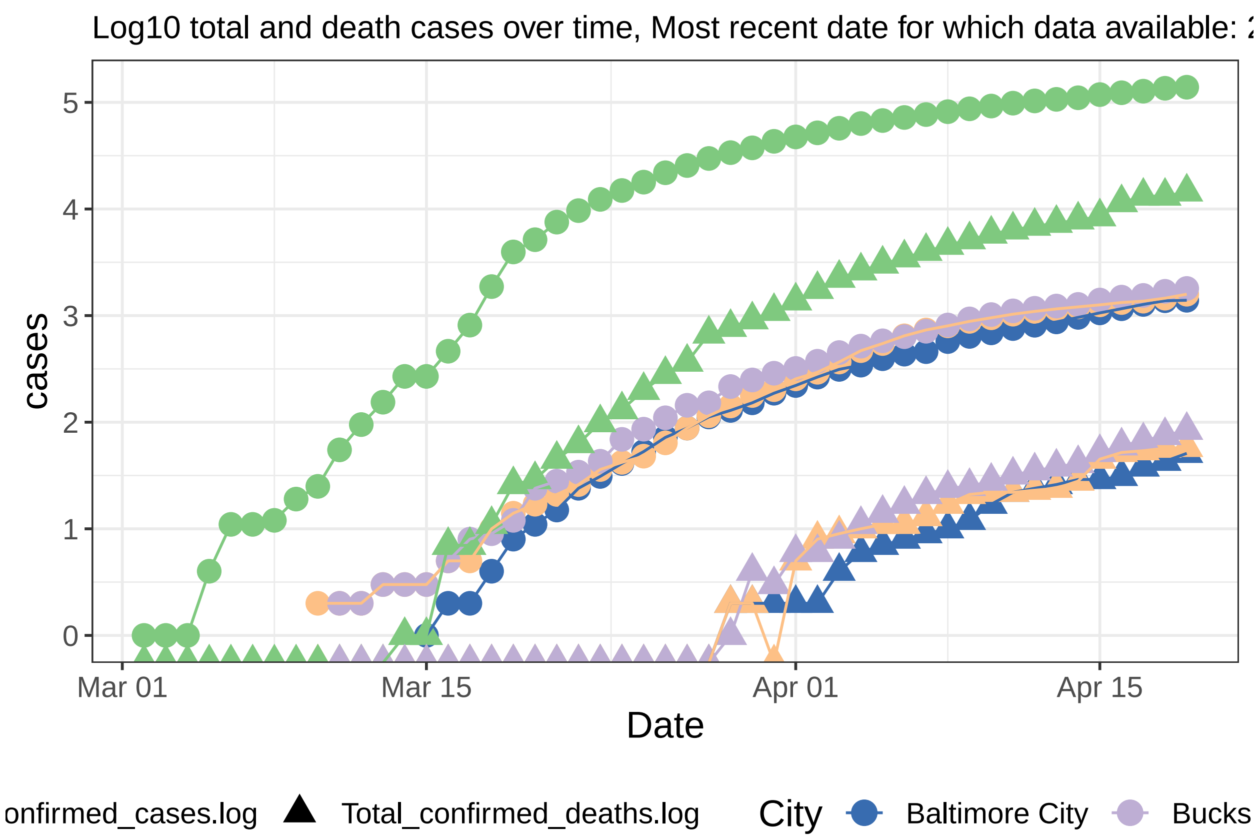
```
## Warning: Removed 100 rows containing missing values (geom_point).

## Warning: Removed 100 row(s) containing missing values (geom_path).

## [1] "/Users/stevensmith/Projects/coronavirus/results/Corona_Cases.world.mortality.plot.png"
```

```
write_plot(Corona_Cases.world.casecor.plot,wd = results_dir)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/Corona_Cases.world.casecor.plot.png"
```

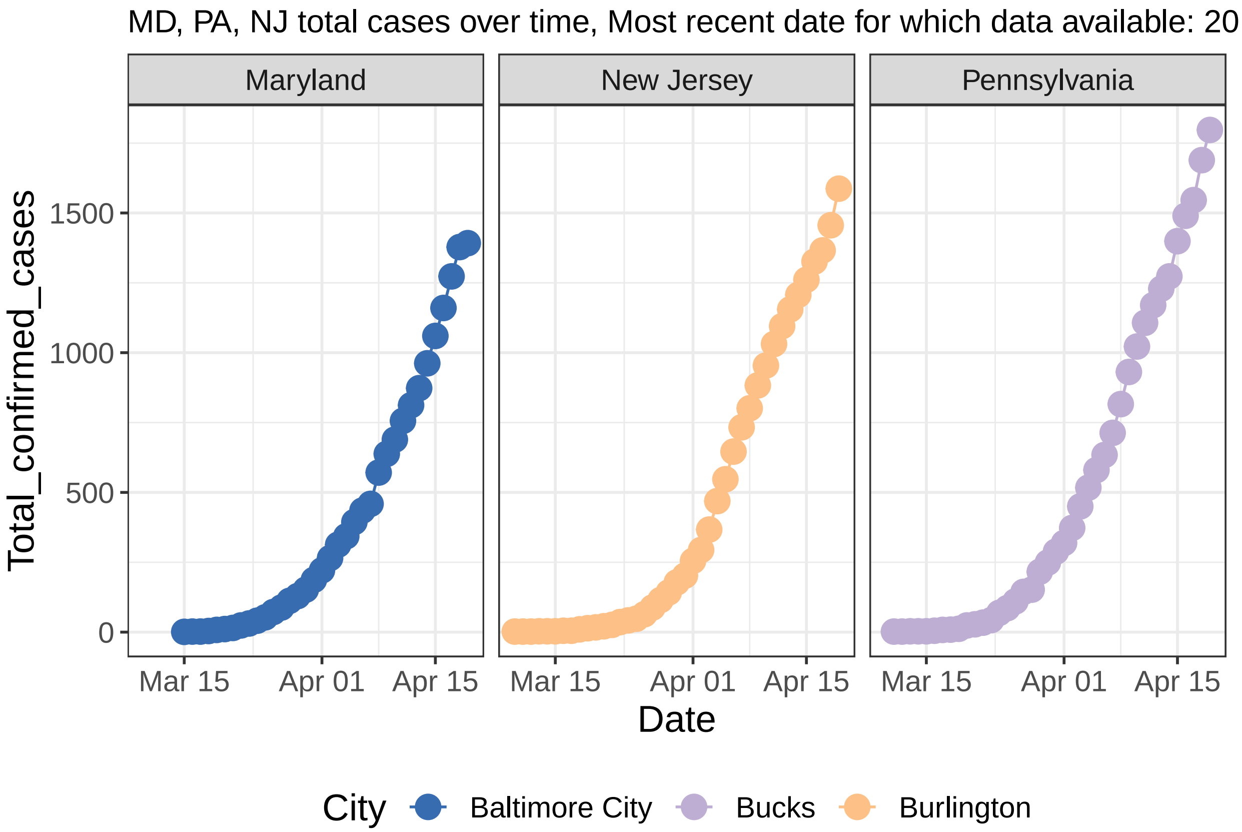
```
##------------------------------------------
## Plot US State Data
##-----------------------------------------

baseplot.US<-ggplot(data=NULL,aes(x=Days_since_100_state,col=case_type))+
  default_theme+
  facet_wrap(~Province.State)+
  ggtitle(paste("Log10 cases over time,",timestamp_plot.world))
```

```
Corona_Cases.US_state.long.plot<-baseplot.US+geom_point(data=Corona_Cases.US_state.long,aes(y=cases.log)
##------------------------------------------
## Plot US City Data
##-----------------------------------------

Corona_Cases.US.plotdata<-filter(Corona_Cases.US_state,Province.State %in% c("Pennsylvania","Maryland",
                                   City %in% c("Bucks","Baltimore City", "New York","Burlington") &
                                   Total_confirmed_cases>0)
timestamp_plot<-paste("Most recent date for which data available:",max(Corona_Cases.US.plotdata$Date))#

city_colors<-c("Bucks"='#beaed4',"Baltimore City"='#386cb0', "New York"='#7fc97f',"Burlington"='#fdc086

##/////////////////////////
### Plot death vs total case correlation

(Corona_Cases.city.loglong.plot<-ggplot(melt(Corona_Cases.US.plotdata,measure.vars = c("Total_confirmed_
  geom_point(size=4)+
    geom_line()+
  default_theme+
  #facet_wrap(~case_type)+
    ggtitle(paste("Log10 total and death cases over time,",timestamp_plot))+
theme(legend.position = "bottom",plot.title = element_text(size=12))+
    scale_color_manual(values = city_colors))
```

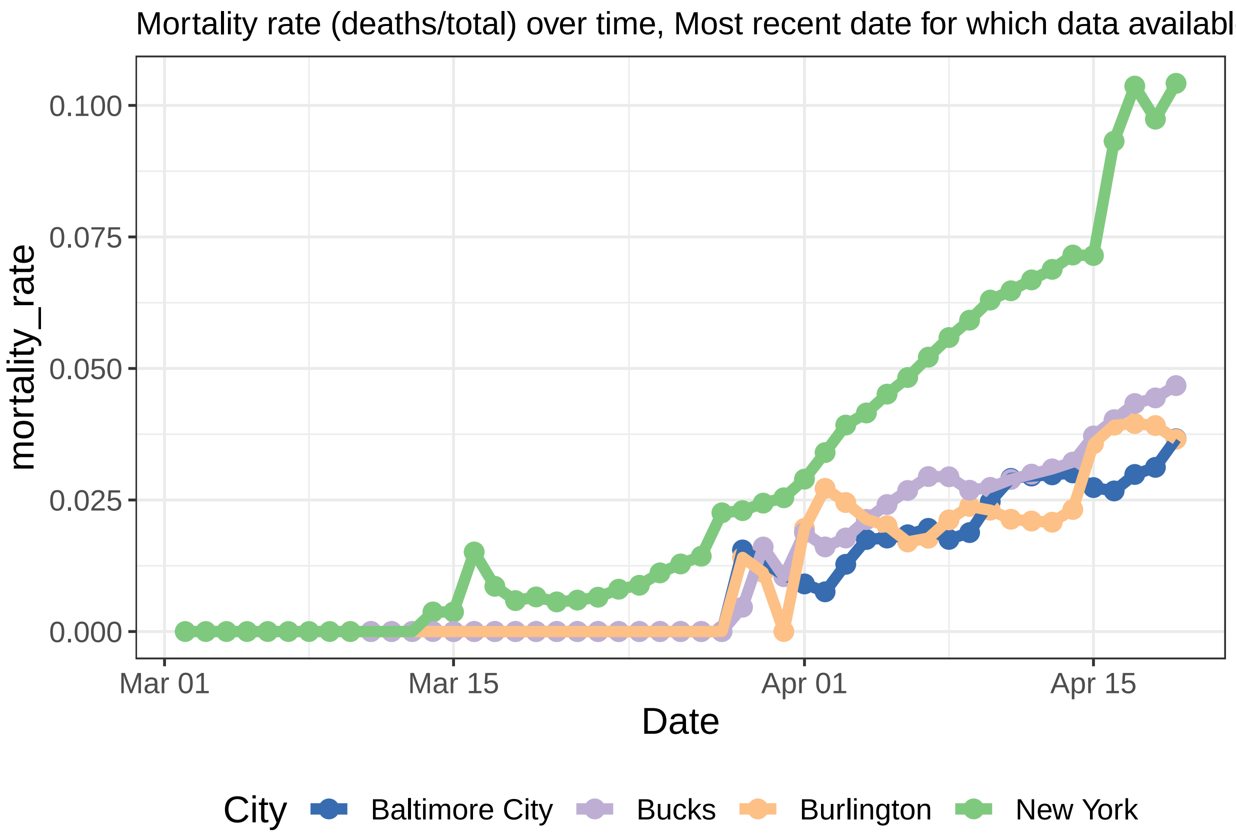



```
(Corona_Cases.city.long.plot<-ggplot(filter(Corona_Cases.US.plotdata,Province.State !="New York"),aes(x=
  geom_point(size=4)+
  geom_line()+
```

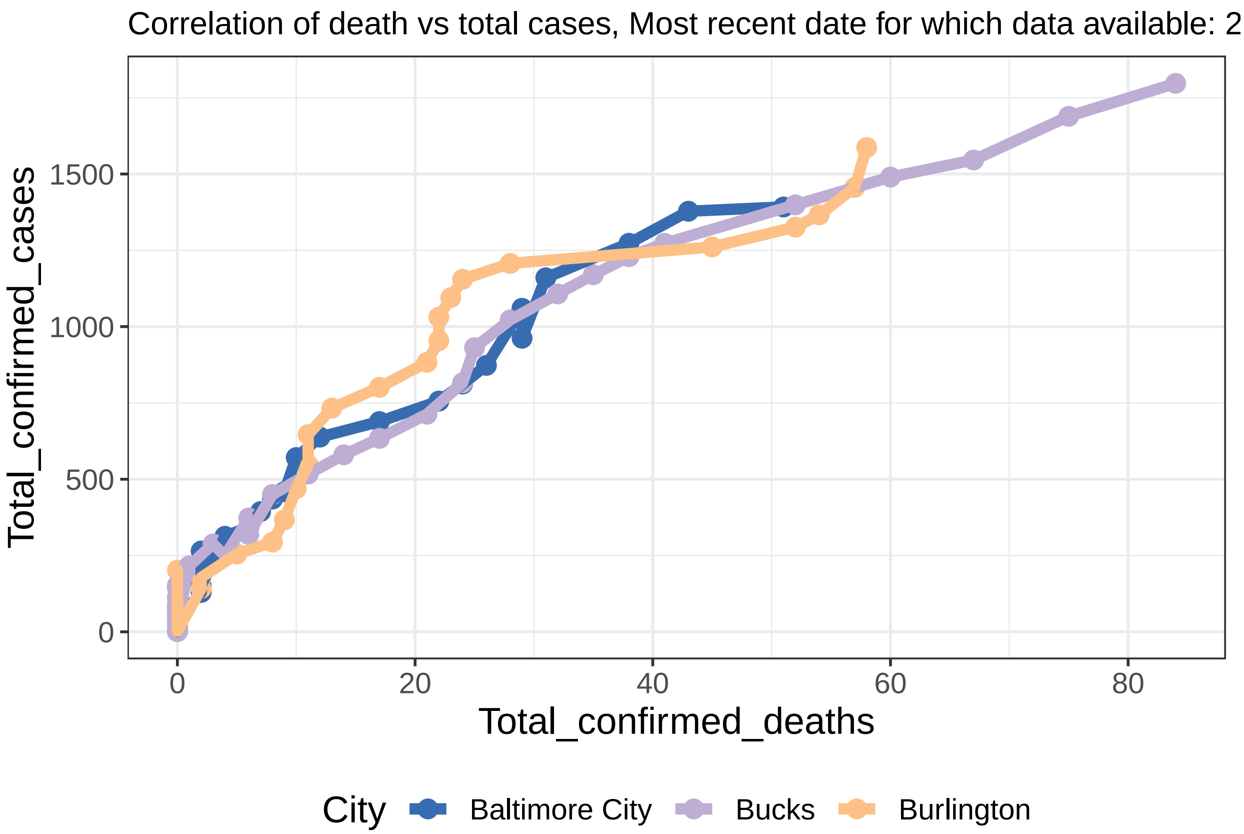
```
  default_theme+
  facet_grid(~Province.State,scales = "free_y")+
  ggtitle(paste("MD, PA, NJ total cases over time,",timestamp_plot))+
  theme(legend.position = "bottom",plot.title = element_text(size=12))+
  scale_color_manual(values = city_colors))
```



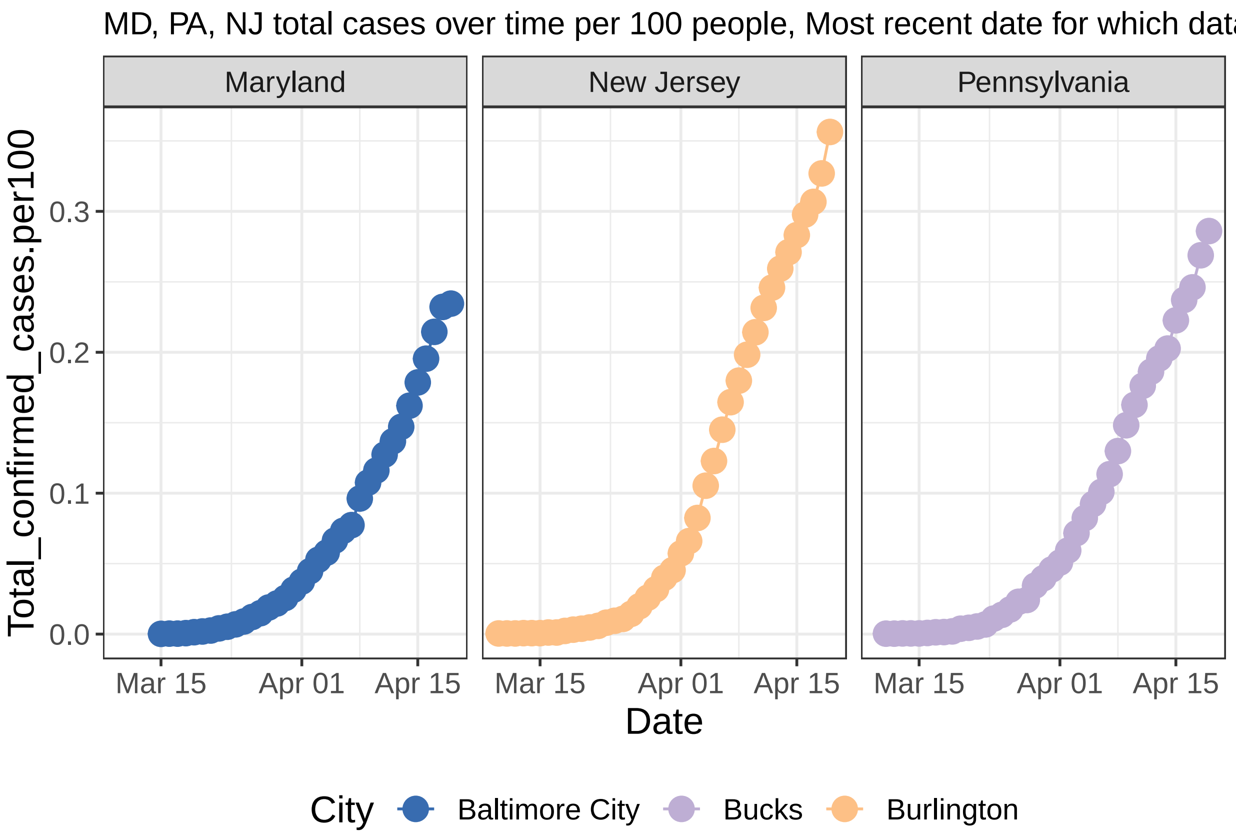

```
(Corona_Cases.city.mortality.plot<-ggplot(Corona_Cases.US.plotdata,aes(x=Date,y=mortality_rate,col=City
  geom_point(size=3)+
  geom_line(size=2)+
  default_theme+
  ggtitle(paste("Mortality rate (deaths/total) over time,",timestamp_plot))+
  theme(legend.position = "bottom",plot.title = element_text(size=12))+
  scale_color_manual(values = city_colors))
```




```
(Corona_Cases.city.casecor.plot<-ggplot(filter(Corona_Cases.US.plotdata,Province.State !="New York"),aes
  geom_point(size=3)+
  geom_line(size=2)+
  default_theme+
  ggtitle(paste("Correlation of death vs total cases,",timestamp_plot))+
  theme(legend.position = "bottom",plot.title = element_text(size=12))+
  scale_color_manual(values = city_colors))
```




```
(Corona_Cases.city.long.normalized.plot<-ggplot(filter(Corona_Cases.US.plotdata,Province.State !="New Y
  geom_point(size=4)+
  geom_line()+
  default_theme+
  facet_grid(~Province.State)+
  ggtitle(paste("MD, PA, NJ total cases over time per 100 people,",timestamp_plot))+
  theme(legend.position = "bottom",plot.title = element_text(size=12))+
  scale_color_manual(values = city_colors))
```




```
write_plot(Corona_Cases.city.long.plot,wd = results_dir_custom)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/custom/Corona_Cases.city.long.plot.png"
```

```
write_plot(Corona_Cases.city.loglong.plot,wd = results_dir_custom)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/custom/Corona_Cases.city.loglong.plot.png"
```

```
write_plot(Corona_Cases.city.mortality.plot,wd = results_dir_custom)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/custom/Corona_Cases.city.mortality.plot.png"
```

```
write_plot(Corona_Cases.city.casecor.plot,wd = results_dir_custom)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/custom/Corona_Cases.city.casecor.plot.png"
```

```
write_plot(Corona_Cases.city.long.normalized.plot,wd = results_dir_custom)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/custom/Corona_Cases.city.long.normalized.plot.p
```

Q1b what is the model

Fit the cases to a linear model 1. Find time at which the case vs date becomes linear in each plot
2. Fit linear model for each city

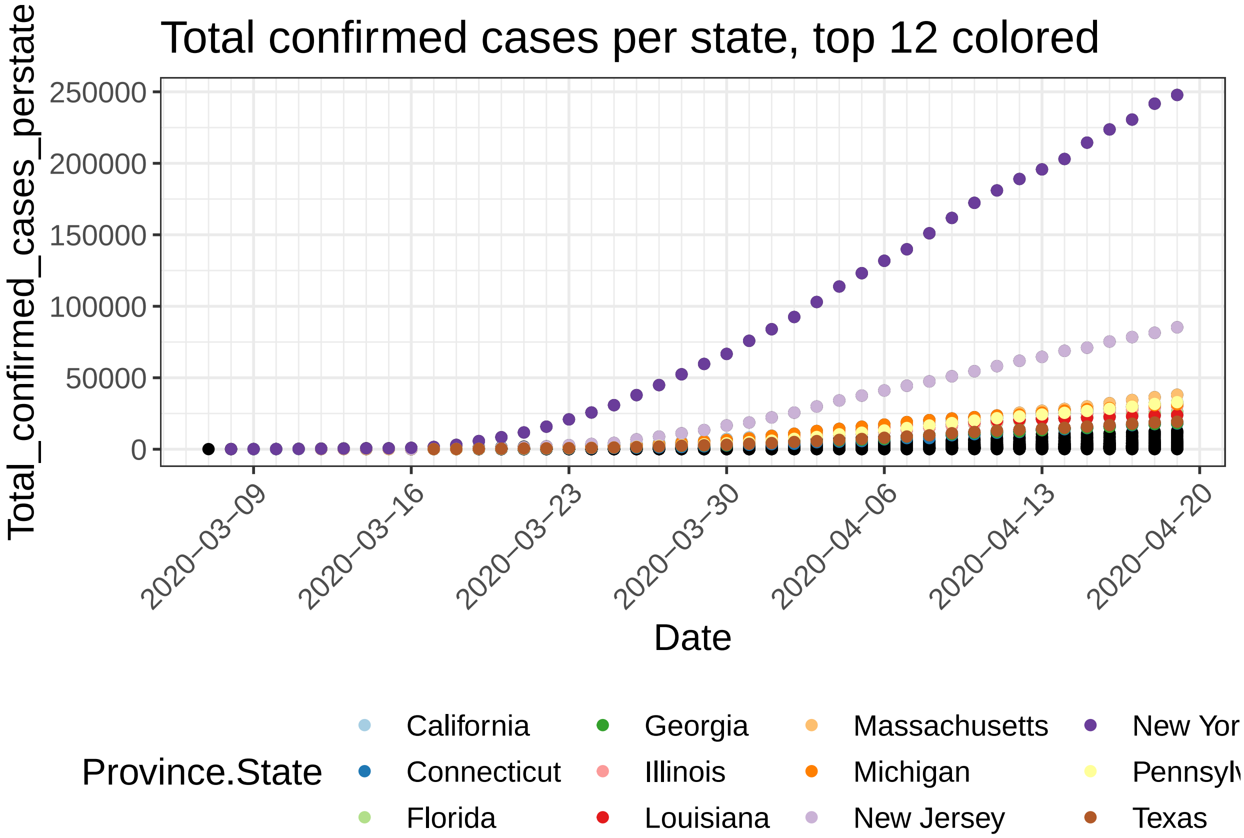
```
# What is the predict # of cases for the next few days?
# How is the model performing historically?

Corona_Cases.US_state.summary<-ddply(Corona_Cases.US_state,
                                     c("Province.State","Date"),
                                     summarise,
```

```
                                            Total_confirmed_cases_perstate=sum(Total_confirmed_cases)) %>%
    filter(Total_confirmed_cases_perstate>100)

# Compute the states with the most cases (for coloring and for linear model)
top_states_totals<-head(ddply(Corona_Cases.US_state.summary,c("Province.State"),summarise, Total_confir

kable(top_states_totals,caption = "Top 12 States, total count ")
```

Table 8: Top 12 States, total count

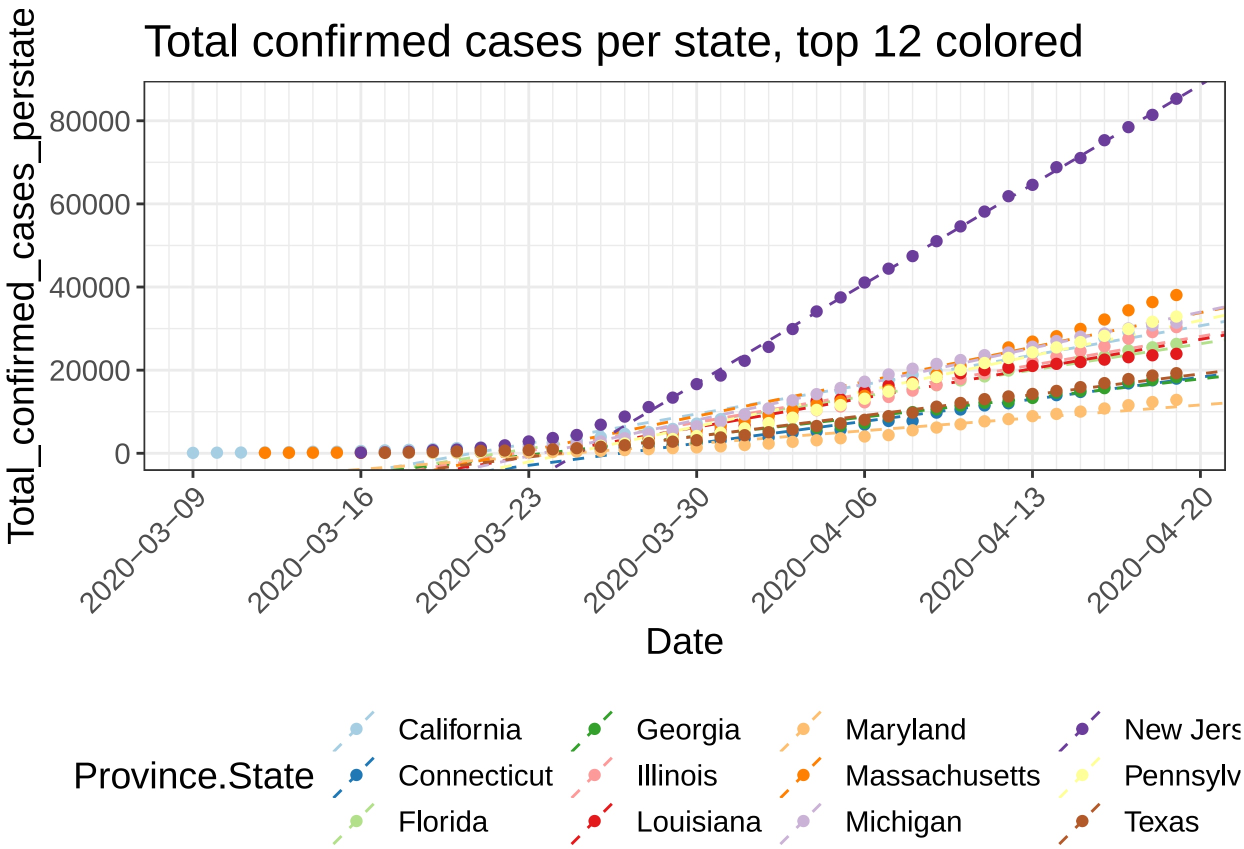
Province.State	Total_confirmed_cases_perstate.max
New York	247815
New Jersey	85301
Massachusetts	38077
Pennsylvania	32902
California	31431
Michigan	31424
Illinois	30357
Florida	26314
Louisiana	23928
Texas	19260
Georgia	18301
Connecticut	17962

```
top_states<-top_states_totals$Province.State

# Manually fix states so that Maryland is switched out for New York
top_states_modified<-c(top_states[top_states !="New York"],"Maryland")

# Plot with all states:
(Corona_Cases.US_state.summary.plot<-ggplot(Corona_Cases.US_state.summary,aes(x=Date,y=Total_confirmed_c
  geom_point()+
  geom_point(data=filter(Corona_Cases.US_state.summary,Province.State %in% top_states),aes(col=Province
  scale_color_brewer(type = "qualitative",palette = "Paired")+
  default_theme+
  theme(axis.text.x = element_text(angle=45,hjust=1),legend.position = "bottom")+
  ggtitle("Total confirmed cases per state, top 12 colored")+
  scale_x_date(date_breaks="1 week",date_minor_breaks="1 day"))
```




```
##-------------------------------------------
## Fit linear model to time vs total cases
##------------------------------------------

# First, find the date at which each state's cases vs time becomes lienar (2nd derivative is about 0)
li<-ddply(Corona_Cases.US_state.summary,c("Province.State"),find_linear_index)

# Compute linear model for each state starting at the point at which data becomes linear
for(i in 1:nrow(li)){
  Province.State.i<-li[i,"Province.State"]
  date.i<-li[i,"V1"]
  data.i<-filter(Corona_Cases.US_state.summary,Province.State==Province.State.i & as.numeric(Date) >= da
  model_results<-lm(data.i,formula = Total_confirmed_cases_perstate~Date)
  slope<-model_results$coefficients[2]
  intercept<-model_results$coefficients[1]
  li[li$Province.State==Province.State.i,"m"]<-slope
  li[li$Province.State==Province.State.i,"b"]<-intercept
  }

# Compute top state case load with fitted model

(Corona_Cases.US_state.lm.plot<-ggplot(filter(Corona_Cases.US_state.summary,Province.State %in% top_sta
    geom_abline(data=filter(li,Province.State %in% top_states_modified),
                aes(slope = m,intercept = b,col=Province.State),lty=2)+
    geom_point(aes(x=Date,y=Total_confirmed_cases_perstate,col=Province.State))+
    scale_color_brewer(type = "qualitative",palette = "Paired")+
    default_theme+
```

```
    theme(axis.text.x = element_text(angle=45,hjust=1),legend.position = "bottom")+
    ggtitle("Total confirmed cases per state, top 12 colored")+
    scale_x_date(date_breaks="1 week",date_minor_breaks="1 day"))
```




```
##------------------------------------------
## Predict the number of total cases over the next week
##------------------------------------------

predicted_days<-c(0,1,2,3,7)+as.numeric(as.Date("2020-04-20"))

predicted_days_df<-data.frame(matrix(ncol=3))
names(predicted_days_df)<-c("Province.State","days","Total_confirmed_cases_perstate")

# USe model parameters to estiamte case loads
for(state.i in top_states_modified){
  predicted_days_df<-rbind(predicted_days_df,
                        data.frame(Province.State=state.i,
                                   prediction_model(m = li[li$Province.State==state.i,"m"],
                                                    b =li[li$Province.State==state.i,"b"] ,
                                                    days =predicted_days )))
  }

predicted_days_df$Date<-as.Date(predicted_days_df$days,origin="1970-01-01")

kable(predicted_days_df,caption = "Predicted total cases over the next week for selected states")
```

Table 9: Predicted total cases over the next week for selected states

Province.State	days	Total_confirmed_cases_perstate	Date
NA	NA	NA	NA
New Jersey	18372	88629.22	2020-04-20
New Jersey	18373	92047.77	2020-04-21
New Jersey	18374	95466.31	2020-04-22
New Jersey	18375	98884.85	2020-04-23
New Jersey	18379	112559.03	2020-04-27
Massachusetts	18372	33836.49	2020-04-20
Massachusetts	18373	35022.45	2020-04-21
Massachusetts	18374	36208.42	2020-04-22
Massachusetts	18375	37394.39	2020-04-23
Massachusetts	18379	42138.26	2020-04-27
Pennsylvania	18372	31965.08	2020-04-20
Pennsylvania	18373	33179.91	2020-04-21
Pennsylvania	18374	34394.75	2020-04-22
Pennsylvania	18375	35609.58	2020-04-23
Pennsylvania	18379	40468.90	2020-04-27
California	18372	30687.57	2020-04-20
California	18373	31698.86	2020-04-21
California	18374	32710.16	2020-04-22
California	18375	33721.45	2020-04-23
California	18379	37766.63	2020-04-27
Michigan	18372	33972.97	2020-04-20
Michigan	18373	35210.00	2020-04-21
Michigan	18374	36447.04	2020-04-22
Michigan	18375	37684.07	2020-04-23
Michigan	18379	42632.21	2020-04-27
Illinois	18372	28121.06	2020-04-20
Illinois	18373	29105.73	2020-04-21
Illinois	18374	30090.41	2020-04-22
Illinois	18375	31075.09	2020-04-23
Illinois	18379	35013.80	2020-04-27
Florida	18372	26357.07	2020-04-20
Florida	18373	27259.24	2020-04-21
Florida	18374	28161.42	2020-04-22
Florida	18375	29063.59	2020-04-23
Florida	18379	32672.28	2020-04-27
Louisiana	18372	27412.33	2020-04-20
Louisiana	18373	28418.79	2020-04-21
Louisiana	18374	29425.24	2020-04-22
Louisiana	18375	30431.69	2020-04-23
Louisiana	18379	34457.51	2020-04-27
Texas	18372	19094.04	2020-04-20
Texas	18373	19808.68	2020-04-21
Texas	18374	20523.32	2020-04-22
Texas	18375	21237.97	2020-04-23
Texas	18379	24096.54	2020-04-27
Georgia	18372	17890.48	2020-04-20
Georgia	18373	18544.63	2020-04-21
Georgia	18374	19198.78	2020-04-22
Georgia	18375	19852.92	2020-04-23
Georgia	18379	22469.51	2020-04-27

Province.State	days	Total_confirmed_cases_perstate	Date
Connecticut	18372	18305.99	2020-04-20
Connecticut	18373	19063.59	2020-04-21
Connecticut	18374	19821.19	2020-04-22
Connecticut	18375	20578.79	2020-04-23
Connecticut	18379	23609.18	2020-04-27
Maryland	18372	11631.74	2020-04-20
Maryland	18373	12074.21	2020-04-21
Maryland	18374	12516.68	2020-04-22
Maryland	18375	12959.15	2020-04-23
Maryland	18379	14729.04	2020-04-27

```
##-------------------------------------------
## Write plots
##------------------------------------------

write_plot(Corona_Cases.US_state.summary.plot,wd = results_dir)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/Corona_Cases.US_state.summary.plot.png"
```

```
write_plot(Corona_Cases.US_state.lm.plot,wd = results_dir)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/Corona_Cases.US_state.lm.plot.png"
```

```
##-------------------------------------------
## Write tables
##------------------------------------------

write.csv(predicted_days_df,file = paste0(results_dir,"predicted_total_cases_days.csv"),quote = F,row.na
```

Q2: What is the predicted number of cases?

What is the prediction of COVID-19 based on model thus far? Additional questions:

WHy did it take to day 40 to start a log linear trend? How long will it be till x number of cases? When will the plateu happen? Are any effects noticed with social distancing? Delays

```
##-------------------------------------------
## Prediction and Prediction Accuracy
##-------------------------------------------



today_num<-max(Corona_Cases.US$Days_since_100)
predicted_days<-today_num+c(1,2,3,7)

#mods = dlply(mydf, .(x3), lm, formula = y ~ x1 + x2)
#today:
Corona_Cases.US[Corona_Cases.US$Days_since_100==(today_num-1),]
Corona_Cases.US[Corona_Cases.US$Days_since_100==today_num,]
Corona_Cases.US$type<-"Historical"


#prediction_values<-prediction_model(m=slope,b=intercept,days = predicted_days)$Total_confirmed_cases

#histoical_model<-data.frame(date=today_num,m=slope,b=intercept)
```

```
# model for previous y days
#historical_model_predictions<-data.frame(day_x=NULL,Days_since_100=NULL,Total_confirmed_cases=NULL,Tot
# for(i in c(1,2,3,4,5,6,7,8,9,10)){
#   #i<-1
# day_x<-today_num-i # 1, 2, 3, 4
# day_x_nextweek<-day_x+c(1,2,3)
# model_fit_x<-lm(data = filter(Corona_Cases.US.case100,Days_since_100 < day_x),formula = Total_confirm
# prediction_day_x_nextweek<-prediction_model(m = model_fit_x$coefficients[2],b = model_fit_x$coefficie
# prediction_day_x_nextweek$type<-"Predicted"
# acutal_day_x_nextweek<-filter(Corona_Cases.US,Days_since_100 %in% day_x_nextweek) %>% select(c(Days_s
# acutal_day_x_nextweek$type<-"Historical"
# historical_model_predictions.i<-data.frame(day_x=day_x,rbind(acutal_day_x_nextweek,prediction_day_x_n
# historical_model_predictions<-rbind(historical_model_predictions.i,historical_model_predictions)
# }

#historical_model_predictions.withHx<-rbind.fill(historical_model_predictions,data.frame(Corona_Cases.U
#historical_model_predictions.withHx$Total_confirmed_cases.log2<-log(historical_model_predictions.withH

(historical_model_predictions.plot<-ggplot(historical_model_predictions.withHx,aes(x=Days_since_100,y=To
    geom_point(size=3)+
    default_theme+
    theme(legend.position = "bottom")+
      #geom_abline(slope = slope,intercept =intercept,lty=2)+
    #facet_wrap(~case_type,ncol=1)+
    scale_color_manual(values = c("Historical"="#377eb8","Predicted"="#e41a1c")))
write_plot(historical_model_predictions.plot,wd=results_dir)
```

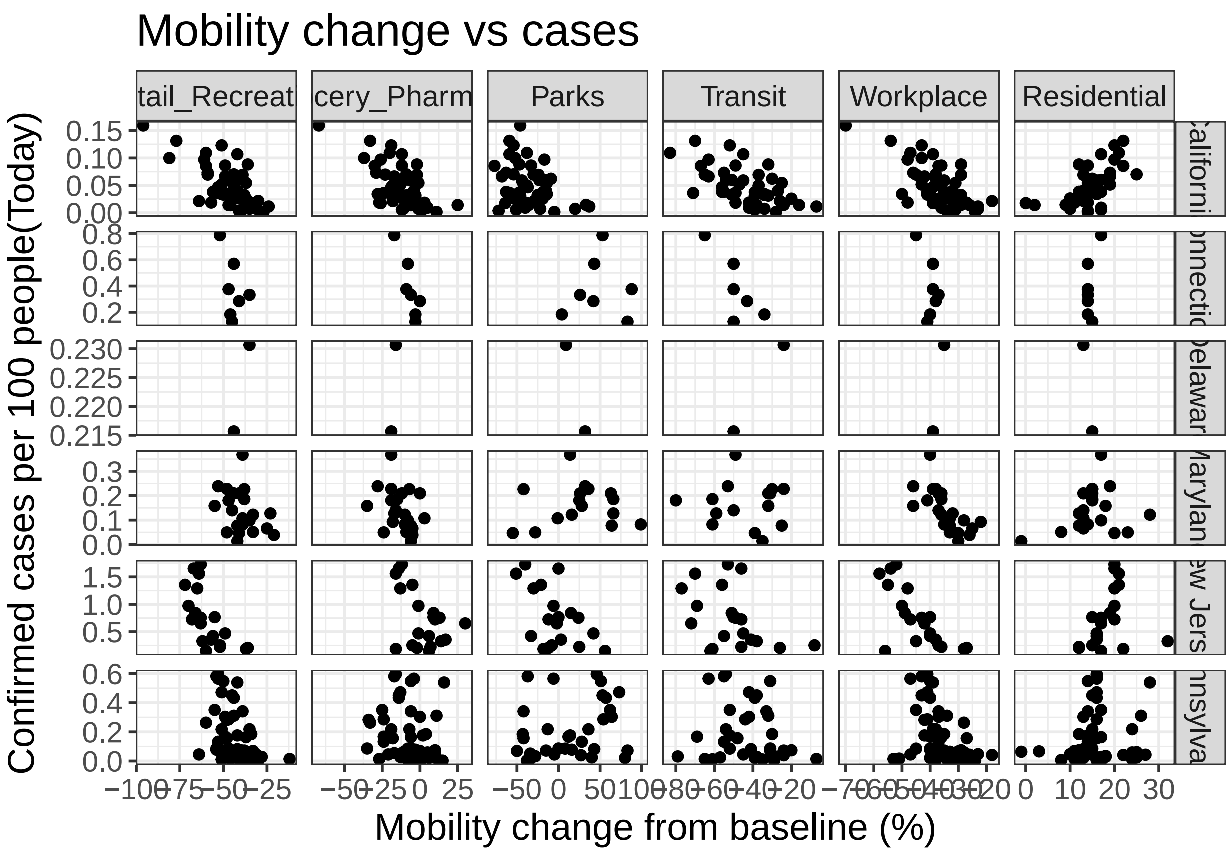
Q3: What is the effect on social distancing, descreased mobility on case load?

Load data from Google which compoutes % change in user mobility relative to baseline for * Recreation
* Workplace
* Residence
* Park
* Grocery

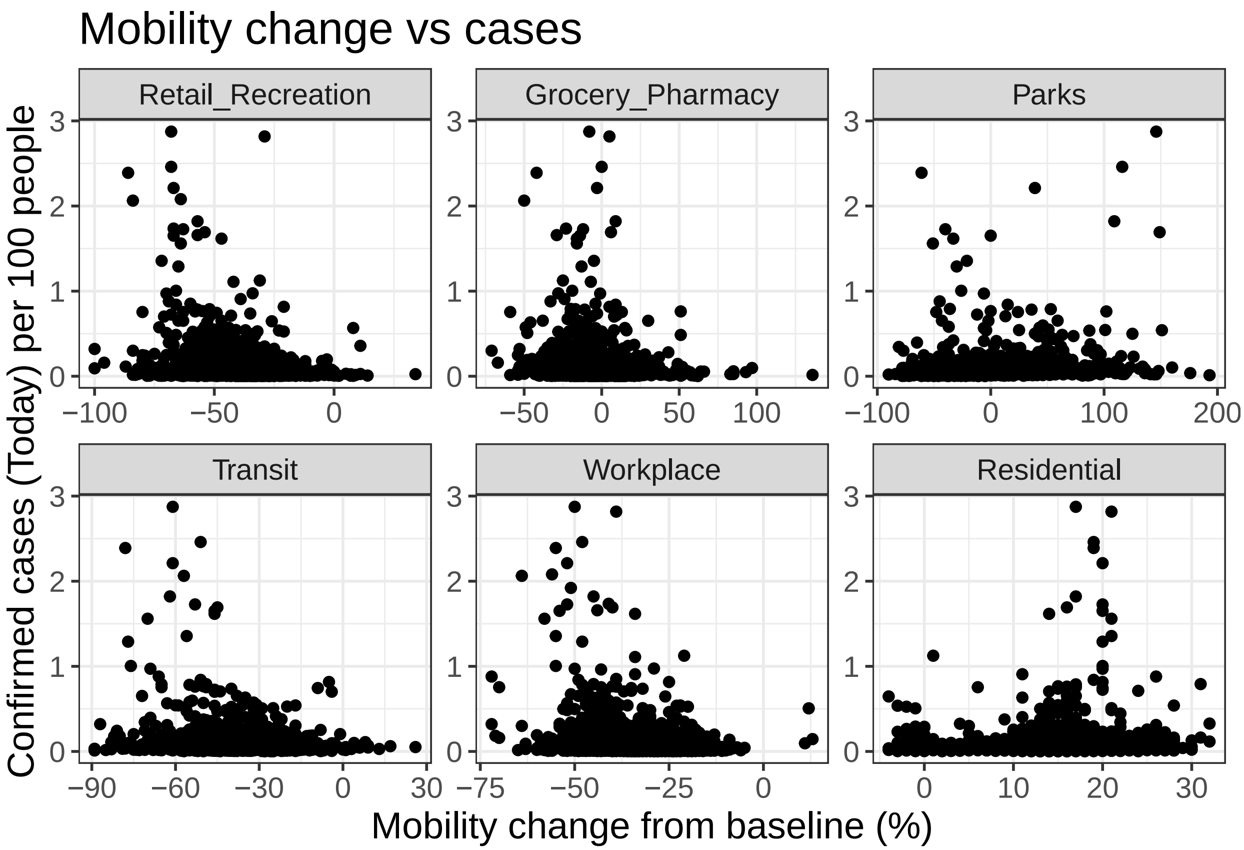
Data from https://www.google.com/covid19/mobility/

```
# See pre-processing section for script on gathering mobility data

# UNDER DEVELOPMENT

mobility<-read.csv("/Users/stevensmith/Projects/MIT_COVID19/mobility.csv",header = T,stringsAsFactors =
#mobility$Retail_Recreation<-as.numeric(sub(mobility$Retail_Recreation,pattern = "%",replacement = ""))
#mobility$Workplace<-as.numeric(sub(mobility$Workplace,pattern = "%",replacement = ""))
#mobility$Residential<-as.numeric(sub(mobility$Residential,pattern = "%",replacement = ""))

##------------------------------------------
## Show relationship between mobility and caseload
##------------------------------------------
mobility$County<-gsub(mobility$County,pattern = " County",replacement = "")
Corona_Cases.US_state.mobility<-merge(Corona_Cases.US_state,plyr::rename(mobility,c("State"="Province.St

#Corona_Cases.US_state.tmp<-merge(metadata,Corona_Cases.US_state.tmp)
# Needs to happen upsteam, see todos
```

```r
#Corona_Cases.US_state.tmp$Total_confirmed_cases.perperson<-Corona_Cases.US_state.tmp$Total_confirmed_c
mobility_measures<-c("Retail_Recreation","Grocery_Pharmacy","Parks","Transit","Workplace","Residential")

plot_data<-filter(Corona_Cases.US_state.mobility, Date.numeric==max(Corona_Cases.US_state$Date.numeric)
plot_data$value<-as.numeric(gsub(plot_data$value,pattern = "%",replacement = ""))
plot_data<-filter(plot_data,!is.na(value))

(mobility.plot<-ggplot(filter(plot_data,Province.State %in% c("Pennsylvania","Maryland","New Jersey","C
  facet_grid(Province.State~variable,scales = "free")+
  xlab("Mobility change from baseline (%)")+
  ylab(paste0("Confirmed cases per 100 people(Today)"))+
  default_theme+
  ggtitle("Mobility change vs cases"))
```




```r
(mobility.global.plot<-ggplot(plot_data,aes(y=Total_confirmed_cases.per100,x=value))+geom_point()+
  facet_wrap(~variable,scales = "free")+
  xlab("Mobility change from baseline (%)")+
  ylab(paste0("Confirmed cases (Today) per 100 people"))+
  default_theme+
  ggtitle("Mobility change vs cases"))
```




```
plot_data.permobility_summary<-ddply(plot_data,c("Province.State","variable"),summarise,cor=cor(y =Tota

kable(plot_data.permobility_summary,caption = "Ranked per-state mobility correlation with total confirm
```

Table 10: Ranked per-state mobility correlation with total confirmed cases

Province.State	variable	cor	median_change
Alaska	Transit	-1.0000000	-63.0
Delaware	Retail_Recreation	1.0000000	-39.5
Delaware	Grocery_Pharmacy	1.0000000	-17.5
Delaware	Parks	-1.0000000	20.5
Delaware	Transit	1.0000000	-37.0
Delaware	Workplace	1.0000000	-37.0
Delaware	Residential	-1.0000000	14.0
Hawaii	Parks	0.9974390	-72.0
Hawaii	Transit	0.9881845	-89.0
Alaska	Residential	0.9643548	13.0
Utah	Workplace	-0.9278361	-37.0
Vermont	Parks	0.9218804	-35.5
New Hampshire	Parks	0.9154004	-20.0
South Dakota	Parks	0.9073927	-26.0
Utah	Retail_Recreation	-0.9011841	-40.0
Connecticut	Grocery_Pharmacy	-0.8840109	-6.0
Hawaii	Grocery_Pharmacy	0.8765364	-34.0
Massachusetts	Workplace	-0.8533862	-39.0
Alaska	Grocery_Pharmacy	-0.8475482	-7.0

Province.State	variable	cor	median_change
Utah	Grocery_Pharmacy	-0.8344952	-4.0
Hawaii	Retail_Recreation	0.8063133	-56.0
North Dakota	Parks	-0.8010272	-34.0
Connecticut	Transit	-0.7839332	-50.0
Rhode Island	Workplace	-0.7826355	-39.5
Utah	Residential	-0.7456066	12.0
New Mexico	Parks	0.7257117	-31.5
Utah	Transit	-0.7217481	-18.0
New Jersey	Workplace	-0.7145321	-44.0
North Dakota	Retail_Recreation	-0.7062088	-43.5
Kansas	Parks	0.6956347	72.0
Massachusetts	Retail_Recreation	-0.6899492	-44.0
California	Retail_Recreation	-0.6866533	-44.0
Maryland	Workplace	-0.6851697	-35.0
California	Workplace	-0.6755135	-36.0
New York	Workplace	-0.6695909	-34.5
Vermont	Grocery_Pharmacy	-0.6684592	-25.0
Maine	Transit	-0.6576257	-50.0
New Jersey	Retail_Recreation	-0.6506045	-62.5
Utah	Parks	-0.6447047	17.0
New York	Retail_Recreation	-0.6286212	-46.0
Connecticut	Residential	0.6215920	14.0
California	Grocery_Pharmacy	-0.6166681	-12.0
California	Residential	0.6163958	14.0
Rhode Island	Residential	-0.6015461	18.5
Montana	Workplace	-0.5901126	-40.5
California	Transit	-0.5884422	-42.0
Nevada	Transit	-0.5883407	-20.0
Massachusetts	Grocery_Pharmacy	-0.5814299	-7.0
Alaska	Workplace	-0.5674858	-34.0
Rhode Island	Retail_Recreation	-0.5671790	-45.0
West Virginia	Parks	0.5503722	-27.0
Connecticut	Workplace	-0.5384628	-39.0
Montana	Transit	-0.5330404	-41.0
Maine	Workplace	-0.5321404	-30.0
Nevada	Retail_Recreation	-0.5246252	-43.0
New Jersey	Parks	-0.5243403	-6.0
Montana	Retail_Recreation	-0.5165435	-51.0
Idaho	Workplace	-0.5154127	-29.5
Kansas	Grocery_Pharmacy	-0.5068814	-14.0
Minnesota	Parks	0.4995832	-10.0
Montana	Parks	-0.4797368	-58.0
Maine	Parks	0.4791450	-31.0
New Jersey	Grocery_Pharmacy	-0.4741164	2.5
Montana	Residential	0.4580929	14.0
Idaho	Transit	-0.4576084	-30.0
Connecticut	Retail_Recreation	-0.4558955	-45.0
Arizona	Grocery_Pharmacy	-0.4527302	-15.0
Pennsylvania	Workplace	-0.4407464	-36.0
Vermont	Residential	0.4399112	11.5
Arkansas	Parks	-0.4336942	-12.0
Massachusetts	Transit	-0.4334904	-45.0

Province.State	variable	cor	median_change
New Mexico	Residential	0.4314707	13.5
Idaho	Grocery_Pharmacy	-0.4262862	-4.0
New York	Parks	0.4257908	20.0
Rhode Island	Parks	0.4244184	52.0
New York	Transit	-0.4228355	-48.0
New Jersey	Transit	-0.4206132	-50.5
Montana	Grocery_Pharmacy	-0.4058904	-16.0
Michigan	Workplace	-0.3972226	-40.0
Pennsylvania	Retail_Recreation	-0.3953493	-45.0
Colorado	Residential	0.3868972	14.0
Virginia	Retail_Recreation	-0.3821241	-35.0
Idaho	Retail_Recreation	-0.3790315	-41.0
Illinois	Transit	-0.3784819	-31.0
Vermont	Retail_Recreation	0.3753874	-57.0
Florida	Parks	-0.3752529	-43.0
Virginia	Transit	-0.3697916	-33.0
Colorado	Workplace	-0.3679019	-39.0
New Mexico	Grocery_Pharmacy	-0.3616207	-11.5
Alabama	Workplace	-0.3597073	-29.0
Arizona	Transit	0.3552493	-38.0
Maryland	Grocery_Pharmacy	-0.3548086	-10.0
Oregon	Parks	0.3540172	16.5
New Mexico	Retail_Recreation	-0.3506276	-42.5
Alaska	Retail_Recreation	0.3475180	-39.0
Maryland	Retail_Recreation	-0.3434351	-39.0
Rhode Island	Grocery_Pharmacy	0.3389646	-7.5
Arizona	Residential	0.3384400	13.0
Minnesota	Transit	-0.3338592	-28.5
Colorado	Retail_Recreation	-0.3311956	-44.0
Mississippi	Parks	0.3289354	-25.0
North Dakota	Grocery_Pharmacy	-0.3284422	-9.5
Colorado	Parks	-0.3266929	2.0
South Dakota	Transit	-0.3260936	-40.0
California	Parks	-0.3252406	-38.0
Washington	Transit	-0.3227539	-33.5
Florida	Residential	0.3222273	14.0
Wisconsin	Transit	-0.3180787	-23.5
Texas	Transit	0.3150704	-42.0
Arkansas	Retail_Recreation	-0.3131198	-30.0
Florida	Transit	-0.3112155	-49.0
Nebraska	Grocery_Pharmacy	-0.3094833	0.0
Idaho	Parks	0.3088466	-22.0
Colorado	Grocery_Pharmacy	-0.3045091	-17.0
Illinois	Workplace	-0.3029210	-30.0
Arizona	Retail_Recreation	-0.3012594	-42.5
Mississippi	Grocery_Pharmacy	-0.2992033	-8.0
North Dakota	Workplace	0.2951603	-33.5
New York	Grocery_Pharmacy	-0.2939106	8.0
Virginia	Workplace	-0.2899666	-31.5
Pennsylvania	Parks	0.2840859	13.0
Kansas	Retail_Recreation	-0.2836909	-39.0
New Jersey	Residential	0.2811709	18.0

Province.State	variable	cor	median_change
Colorado	Transit	-0.2809509	-36.0
Maine	Grocery_Pharmacy	-0.2749941	-13.0
Oklahoma	Grocery_Pharmacy	0.2743052	0.0
Virginia	Grocery_Pharmacy	-0.2671235	-8.0
Oregon	Residential	0.2661769	10.5
Florida	Workplace	-0.2656199	-33.0
Arkansas	Residential	0.2631416	12.0
Kentucky	Parks	0.2616025	28.5
Georgia	Grocery_Pharmacy	-0.2570742	-10.0
Tennessee	Retail_Recreation	-0.2557854	-30.0
New Hampshire	Grocery_Pharmacy	-0.2551133	-6.0
Indiana	Grocery_Pharmacy	-0.2535594	-5.5
New Hampshire	Residential	-0.2529407	14.0
North Dakota	Residential	0.2517843	17.0
Maryland	Residential	0.2493939	15.0
Iowa	Workplace	-0.2479870	-29.0
Michigan	Grocery_Pharmacy	-0.2463488	-11.0
Massachusetts	Residential	0.2439284	15.0
Maine	Retail_Recreation	-0.2417156	-42.0
Texas	Residential	-0.2396631	15.0
Texas	Parks	0.2321230	-42.0
West Virginia	Grocery_Pharmacy	-0.2307729	-6.0
Pennsylvania	Grocery_Pharmacy	-0.2234022	-6.0
Rhode Island	Transit	-0.2223621	-56.0
South Carolina	Residential	0.2201064	12.0
Alabama	Residential	0.2196409	11.0
Michigan	Retail_Recreation	-0.2193041	-53.0
Iowa	Residential	-0.2147003	13.0
Georgia	Retail_Recreation	-0.2141297	-41.0
West Virginia	Retail_Recreation	0.2088251	-38.5
North Carolina	Retail_Recreation	-0.2070155	-33.0
Virginia	Residential	0.2067519	14.0
Washington	Workplace	-0.2043014	-38.0
Georgia	Workplace	-0.2031488	-33.5
Alabama	Transit	-0.2011211	-36.5
Wisconsin	Parks	0.2010301	51.5
Nevada	Residential	0.2004036	17.0
Tennessee	Grocery_Pharmacy	-0.1984328	6.0
New Hampshire	Retail_Recreation	-0.1977905	-41.0
West Virginia	Workplace	0.1975290	-32.5
Kentucky	Workplace	-0.1951285	-35.0
Washington	Parks	0.1943336	-3.5
South Dakota	Retail_Recreation	-0.1933995	-38.5
Michigan	Parks	0.1870128	30.0
South Carolina	Workplace	0.1862729	-30.0
Alabama	Grocery_Pharmacy	-0.1849553	-2.0
Arizona	Workplace	-0.1835159	-35.0
South Carolina	Retail_Recreation	-0.1827910	-35.0
Wisconsin	Workplace	-0.1817482	-31.0
Oklahoma	Residential	0.1806528	15.0
Hawaii	Workplace	-0.1786337	-46.0
Nevada	Workplace	-0.1742736	-40.0

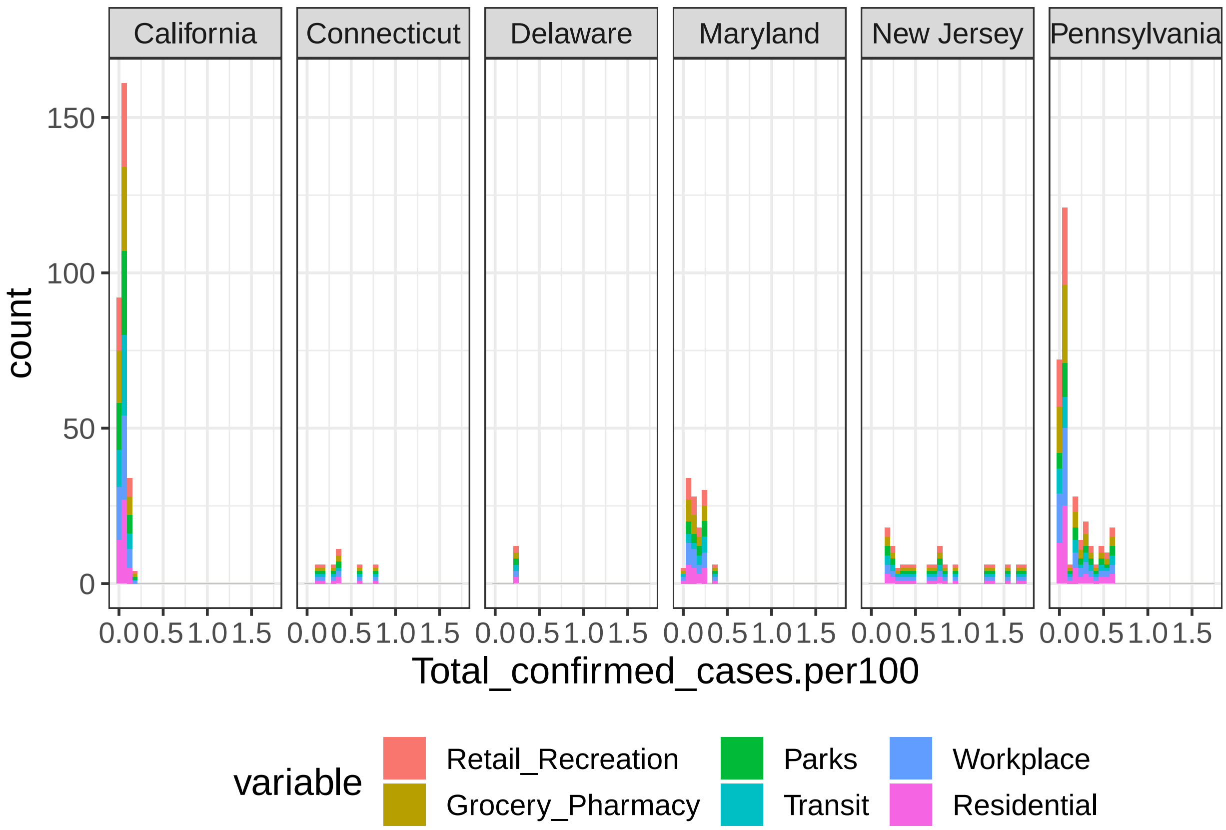
Province.State	variable	cor	median_change
Illinois	Residential	0.1731195	14.0
Ohio	Transit	0.1704157	-28.0
North Carolina	Transit	0.1677937	-32.0
South Carolina	Parks	-0.1660195	-23.0
Tennessee	Workplace	-0.1659508	-31.0
Tennessee	Residential	0.1646697	11.5
Oklahoma	Workplace	-0.1642756	-31.0
Alabama	Parks	0.1633032	-1.0
Missouri	Transit	-0.1630140	-23.0
Oklahoma	Retail_Recreation	0.1585736	-31.0
Arkansas	Workplace	-0.1565230	-26.0
Florida	Grocery_Pharmacy	-0.1559133	-14.0
Indiana	Retail_Recreation	-0.1557982	-38.0
Oregon	Grocery_Pharmacy	0.1535339	-7.0
Hawaii	Residential	-0.1532690	19.0
Texas	Workplace	0.1527706	-31.0
South Dakota	Grocery_Pharmacy	0.1524109	-9.0
Pennsylvania	Transit	-0.1523603	-41.5
Wisconsin	Residential	-0.1455945	14.0
Wisconsin	Grocery_Pharmacy	0.1430202	-1.5
Minnesota	Retail_Recreation	0.1420350	-41.0
Nebraska	Transit	0.1393742	-11.5
Arizona	Parks	0.1375308	-44.5
Idaho	Residential	-0.1359717	11.0
Kentucky	Residential	0.1343560	12.0
Illinois	Retail_Recreation	-0.1337023	-40.0
New Hampshire	Transit	-0.1332052	-57.0
Mississippi	Workplace	-0.1306779	-33.0
Florida	Retail_Recreation	-0.1300595	-43.0
Georgia	Residential	-0.1284274	13.0
Oklahoma	Parks	-0.1282496	-19.0
Maine	Residential	-0.1277053	11.0
Pennsylvania	Residential	0.1252114	15.0
Vermont	Workplace	-0.1168951	-43.0
Tennessee	Parks	0.1100873	10.5
Ohio	Residential	0.1081276	14.0
Illinois	Grocery_Pharmacy	-0.0998410	2.0
Illinois	Parks	0.0990821	26.5
Wisconsin	Retail_Recreation	-0.0961856	-44.0
New Mexico	Workplace	-0.0960656	-34.0
Washington	Retail_Recreation	-0.0960212	-42.0
Indiana	Workplace	-0.0959906	-34.0
Virginia	Parks	0.0956155	6.0
Washington	Residential	0.0921385	13.0
North Carolina	Parks	-0.0917782	7.0
Kansas	Transit	-0.0883159	-26.5
New Hampshire	Workplace	-0.0862113	-37.0
Michigan	Residential	0.0843610	15.0
Ohio	Workplace	-0.0838715	-35.0
Nebraska	Residential	-0.0817917	14.0
Nevada	Parks	-0.0813650	-12.5
New York	Residential	0.0805757	17.5

Province.State	variable	cor	median_change
Oregon	Transit	-0.0805134	-28.0
Kentucky	Transit	0.0796651	-31.0
North Carolina	Grocery_Pharmacy	0.0781098	1.0
Nebraska	Workplace	0.0729846	-33.0
Maryland	Parks	0.0709456	27.0
Arkansas	Transit	0.0694013	-27.0
Texas	Retail_Recreation	-0.0676429	-39.0
New Mexico	Transit	0.0676321	-37.0
Ohio	Retail_Recreation	0.0635497	-36.0
Missouri	Grocery_Pharmacy	-0.0629333	2.0
Georgia	Transit	-0.0623702	-35.0
Missouri	Retail_Recreation	-0.0614052	-36.5
Connecticut	Parks	0.0577733	43.0
North Carolina	Residential	0.0572740	13.0
Ohio	Grocery_Pharmacy	0.0564511	0.0
Indiana	Residential	0.0559441	12.0
Nebraska	Parks	-0.0552271	55.5
Kansas	Workplace	-0.0544930	-31.5
Arkansas	Grocery_Pharmacy	0.0542347	3.5
Nevada	Grocery_Pharmacy	-0.0502032	-11.0
South Carolina	Transit	-0.0480403	-45.0
Iowa	Parks	0.0474548	28.5
Indiana	Parks	-0.0467895	29.0
Missouri	Workplace	0.0459470	-28.5
Maryland	Transit	-0.0458910	-39.0
Michigan	Transit	0.0436864	-46.0
Iowa	Transit	-0.0432352	-25.0
Massachusetts	Parks	-0.0404740	39.0
Iowa	Grocery_Pharmacy	-0.0401015	4.0
Washington	Grocery_Pharmacy	-0.0400099	-7.0
Minnesota	Grocery_Pharmacy	-0.0396030	-5.0
Iowa	Retail_Recreation	-0.0392869	-37.0
Mississippi	Residential	0.0391384	13.0
West Virginia	Transit	0.0351261	-45.0
Oklahoma	Transit	0.0348631	-27.0
South Carolina	Grocery_Pharmacy	-0.0343276	1.0
North Dakota	Transit	-0.0325083	-48.0
Georgia	Parks	-0.0321167	-6.0
Oregon	Workplace	-0.0320910	-32.0
Kentucky	Retail_Recreation	0.0316665	-29.0
Tennessee	Transit	0.0282359	-32.0
Missouri	Parks	0.0264083	0.0
Minnesota	Residential	0.0247357	17.0
Vermont	Transit	0.0227965	-63.0
Mississippi	Transit	-0.0225227	-38.5
South Dakota	Residential	0.0216262	15.0
Texas	Grocery_Pharmacy	-0.0176207	-13.0
Minnesota	Workplace	-0.0155862	-33.0
Indiana	Transit	-0.0147170	-29.0
West Virginia	Residential	0.0135475	11.0
Mississippi	Retail_Recreation	0.0134457	-40.0
Ohio	Parks	0.0127799	67.5

Province.State	variable	cor	median_change
Kentucky	Grocery_Pharmacy	0.0118563	4.0
South Dakota	Workplace	0.0098588	-35.0
Oregon	Retail_Recreation	0.0074022	-41.0
North Carolina	Workplace	0.0051079	-31.0
Nebraska	Retail_Recreation	0.0048579	-37.5
Kansas	Residential	0.0045322	13.0
Missouri	Residential	-0.0042579	13.0
Alabama	Retail_Recreation	0.0007484	-39.0
Alaska	Parks	NA	29.0
District of Columbia	Retail_Recreation	NA	-69.0
District of Columbia	Grocery_Pharmacy	NA	-28.0
District of Columbia	Parks	NA	-65.0
District of Columbia	Transit	NA	-69.0
District of Columbia	Workplace	NA	-48.0
District of Columbia	Residential	NA	17.0

```
# sanity check
ggplot(filter(plot_data,Province.State %in% c("Pennsylvania","Maryland","New Jersey","California","Dela
  facet_grid(~Province.State)+
    default_theme+
  theme(legend.position = "bottom")
```

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



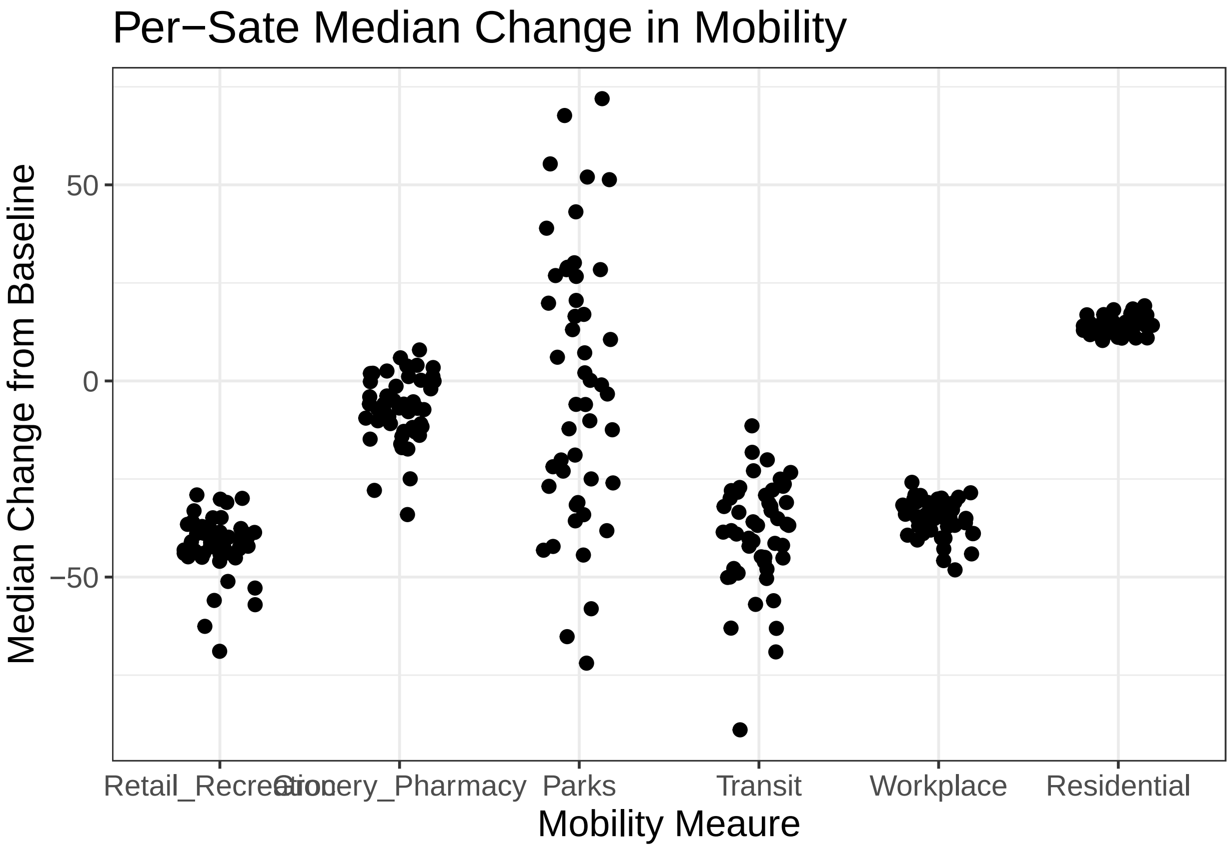

```
write_plot(mobility.plot,wd = results_dir)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/mobility.plot.png"
```

```
write_plot(mobility.global.plot,wd = results_dir)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/mobility.global.plot.png"
```

```
(plot_data.permobility_summary.plot<-ggplot(plot_data.permobility_summary,aes(x=variable,y=median_chang
  geom_jitter(size=2,width=.2)+
  #geom_jitter(data=plot_data.permobility_summary %>% arrange(-abs(median_change)) %>% head(n=15),aes(c
  default_theme+
  ggtitle("Per-Sate Median Change in Mobility")+
  xlab("Mobility Meaure")+
  ylab("Median Change from Baseline"))
```




```
write_plot(plot_data.permobility_summary.plot,wd = results_dir)
```

```
## [1] "/Users/stevensmith/Projects/coronavirus/results/plot_data.permobility_summary.plot.png"
```

DELIVERABLE MANIFEST

The following link to commited documents pushed to github. These are provided as a convienence, but note this is a manual process. The generation of reports, plots and tables is not coupled to the execution of this markdown. ## Report This report, html & pdf

Plots

```
github_root<-"https://github.com/sbs87/coronavirus/blob/master/"
```

```
plot_handle<-c("Corona_Cases.world.long.plot",
               "Corona_Cases.world.loglong.plot",
               "Corona_Cases.world.mortality.plot",
               "Corona_Cases.world.casecor.plot",
               "Corona_Cases.city.long.plot",
               "Corona_Cases.city.loglong.plot",
               "Corona_Cases.city.mortality.plot",
               "Corona_Cases.city.casecor.plot",
               "Corona_Cases.city.long.normalized.plot",
               "Corona_Cases.US_state.lm.plot",
               "Corona_Cases.US_state.summary.plot")


deliverable_manifest<-data.frame(
  name=c("World total & death cases, longitudinal",
         "World log total & death cases, longitudinal",
         "World mortality",
         "World total & death cases, correlation",
         "City total & death cases, longitudinal",
         "City log total & death cases, longitudinal",
         "City mortality",
         "City total & death cases, correlation",
         "City population normalized total & death cases, longitudinal",
         "State total cases (select) with linear model, longitudinal",
         "State total cases, longitudinal"),
  plot_handle=plot_handle,
  link=paste0(github_root,"results/",plot_handle,".png")
)


(tmp<-data.frame(row_out=apply(deliverable_manifest,MARGIN = 1,FUN = function(x) paste(x[1],x[2],x[3],se
```

```
##
## 1                                           World total & death cases, longitudinal | Corona_Cases.wo
## 2                                 World log total & death cases, longitudinal | Corona_Cases.world.lo
## 3                                                         World mortality | Corona_Cases.world.mortal
## 4                                    World total & death cases, correlation | Corona_Cases.world.ca
## 5                                            City total & death cases, longitudinal | Corona_Cases
## 6                                  City log total & death cases, longitudinal | Corona_Cases.city.l
## 7                                                          City mortality | Corona_Cases.city.morta
## 8                                          City total & death cases, correlation | Corona_Cases.city.c
## 9  City population normalized total & death cases, longitudinal | Corona_Cases.city.long.normalized.p
## 10                    State total cases (select) with linear model, longitudinal | Corona_Cases.US_s
## 11                                      State total cases, longitudinal | Corona_Cases.US_state.summa
```

```
row_out<-apply(tmp, 2, paste, collapse="\t\n")
```

name	handle	link
World total & death cases, longitudinal	Corona_Cases.world.long.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.world.long.plot.png
World log total & death cases, longitudinal	Corona_Cases.world.loglong.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.world.loglong.plot.png
World mortality	Corona_Cases.world.mortality.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.world.mortality.plot.png

name	handle	link
World total & death cases, correlation	Corona_Cases.world.casecor.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.world.casecor.plot.png
City total & death cases, longitudinal	Corona_Cases.city.long.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.city.long.plot.png
City log total & death cases, longitudinal	Corona_Cases.city.loglong.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.city.loglong.plot.png

name	handle	link
City mortality	Corona_Cases.city.mortality.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.city.mortality.plot.png
City total & death cases, correlation	Corona_Cases.city.casecor.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.city.casecor.plot.png
City population normalized total & death cases, longitudinal	Corona_Cases.city.long.normalized.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.city.long.normalized.plot.png

name	handle	link
State total cases (select) with linear model, longitudinal	Corona_Cases.US_state.lm.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.US_state.lm.plot.png
State total cases, longitudinal	Corona_Cases.US_state.summary.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.US_state.summary.plot.png

Tables

CONCLUSION

Overall, the trends of COVID-19 cases is no longer in log-linear phase for world or U.S. (but some regions like MD are still in the log-linear phase). Mortality rate (deaths/confirmed RNA-based cases) is >1%, with a range depending on region. Mobility is not a strong indicator of caseload (U.S. data).

See table below for detailed breakdown.

Question	Answer
What is the effect on social distancing, descreased mobility on case load?	There is not a strong apparent effect on decreased mobility (work, grocery, retail) or increased mobility (at residence, parks) on number of confirmed cases, either as a country (U.S.) or state level. California appears to have one of the best correlations, but this is a mixed bag
What is the trend in cases, mortality across geopgraphical regions?	The confirmed total casees and mortality is overall log-linear for most countries, with a trailing off beginning for most (inlcuding U.S.). On the state level, NY, NJ, PA starting to trail off; MD is still in log-linear phase. Mortality and case load are highly correlated for NY, NJ, PA, MD. The mortality rate flucutates for a given region, but is about 3% overall.

END

End: ##—— Mon Apr 20 00:45:41 2020 ——##

Cheatsheet: http://rmarkdown.rstudio.com>

Sandbox

```
# Geographical heatmap!
install.packages("maps")
library(maps)
library
mi_counties <- map_data("county", "pennsylvania") %>%
```

```r
  select(lon = long, lat, group, id = subregion)
head(mi_counties)

ggplot(mi_counties, aes(lon, lat)) +
  geom_point(size = .25, show.legend = FALSE) +
  coord_quickmap()
mi_counties$cases<-1:2226
name_overlaps(metadata,Corona_Cases.US_state)

tmp<-merge(Corona_Cases.US_state,metadata)
ggplot(filter(tmp,Province.State=="Pennsylvania"), aes(Long, Lat, group = as.factor(City))) +
  geom_polygon(aes(fill = Total_confirmed_cases), colour = "grey50") +
  coord_quickmap()


ggplot(Corona_Cases.US_state, aes(Long, Lat))+
  geom_polygon(aes(fill = Total_confirmed_cases ), color = "white")+
  scale_fill_viridis_c(option = "C")
dev.off()


require(maps)
require(viridis)

world_map <- map_data("world")
ggplot(world_map, aes(x = long, y = lat, group = group)) +
  geom_polygon(fill="lightgray", colour = "white")

head(world_map)
head(Corona_Cases.US_state)
unique(select(world_map,c("region","group"))) %>% filter()

some.eu.countries <- c(
  "US"
)
# Retrievethe map data
some.eu.maps <- map_data("world", region = some.eu.countries)

# Compute the centroid as the mean longitude and lattitude
# Used as label coordinate for country's names
region.lab.data <- some.eu.maps %>%
  group_by(region) %>%
  summarise(long = mean(long), lat = mean(lat))

unique(filter(some.eu.maps,subregion %in% Corona_Cases.US_state$Province.State) %>% select(subregion))
unique(Corona_Cases.US_state$Total_confirmed_cases.log)
ggplot(filter(Corona_Cases.US_state,Date=="2020-04-17") aes(x = Long, y = Lat)) +
  geom_polygon(aes( fill = Total_confirmed_cases.log))+
  #geom_text(aes(label = region), data = region.lab.data,  size = 3, hjust = 0.5)+
  #scale_fill_viridis_d()+
  #theme_void()+
  theme(legend.position = "none")
library("sf")
```

```
library("rnaturalearth")
library("rnaturalearthdata")

world <- ne_countries(scale = "medium", returnclass = "sf")
class(world)
ggplot(data = world) +
    geom_sf()

counties <- st_as_sf(map("county", plot = FALSE, fill = TRUE))
counties <- subset(counties, grepl("florida", counties$ID))
counties$area <- as.numeric(st_area(counties))
#install.packages("lwgeom")
class(counties)
head(counties)
ggplot(data = world) +
    geom_sf(data=Corona_Cases.US_state) +
    #geom_sf(data = counties, aes(fill = area)) +
  geom_sf(data = counties, aes(fill = area)) +
   # scale_fill_viridis_c(trans = "sqrt", alpha = .4) +
    coord_sf(xlim = c(-88, -78), ylim = c(24.5, 33), expand = FALSE)


head(counties)
tmp<-unique(select(filter(Corona_Cases.US_state,Date=="2020-04-17"),c(Lat,Long,Total_confirmed_cases.pe
st_as_sf(map("county", plot = FALSE, fill = TRUE))

join::inner_join.sf(Corona_Cases.US_state, counties)

library(sf)
library(sp)

nc <- st_read(system.file("shape/nc.shp", package="sf"))
class(nc)


spdf <- SpatialPointsDataFrame(coords = select(Corona_Cases.US_state,c("Lat","Long")), data = Corona_Ca
                               proj4string = CRS("+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,

head(spdf)
class(spdf)
st_cast(spdf)
```

https://stevenbsmith.net