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: ##—— Fri Apr 17 14:42:17 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"

# TODO: customize get_google_mobility.py script, add arguments
```

SET UP ENVIORNMENT

Load libraries and set global variables

```
# timestamp start
timestamp()
## ##------ Fri Apr 17 14:42:17 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/css
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_data
Corona_Deaths.source_url <- "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/cs

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

function_name	description
make_long	converts input data to long format (specialized cases)
name_overlaps	outputs the column names intersection and set diffs of two data frame

```
##------------------------------------------
## 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.log<-m*days+b
  total_cases<-10^total_cases.log
  prediction<-data.frame(Days_since_100=days,Total_confirmed_cases=total_cases,Total_confirmed_cases.log
  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"))
```

```r
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))
}
```

READ IN DATA

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

```r
# 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)

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: X4.16.20"
```

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

```
## [1] "US total: X4.16.20"
```

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

```
## [1] "World deaths: X4.16.20"
```

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

```
## [1] "World total: X4.16.20"
```

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	86	86
Albania	86	86
Algeria	86	86
Andorra	86	86
Angola	86	86
Antigua and Barbuda	86	86

	death	total
Argentina	86	86
Armenia	86	86
Australia	688	688
Austria	86	86
Azerbaijan	86	86
Bahamas	86	86
Bahrain	86	86
Bangladesh	86	86
Barbados	86	86
Belarus	86	86
Belgium	86	86
Belize	86	86
Benin	86	86
Bhutan	86	86
Bolivia	86	86
Bosnia and Herzegovina	86	86
Botswana	86	86
Brazil	86	86
Brunei	86	86
Bulgaria	86	86
Burkina Faso	86	86
Burma	86	86
Burundi	86	86
Cabo Verde	86	86
Cambodia	86	86
Cameroon	86	86
Canada	1290	1290
Central African Republic	86	86
Chad	86	86
Chile	86	86
China	2838	2838
Colombia	86	86
Congo (Brazzaville)	86	86
Congo (Kinshasa)	86	86
Costa Rica	86	86
Cote d'Ivoire	86	86
Croatia	86	86
Cuba	86	86
Cyprus	86	86
Czechia	86	86
Denmark	258	258
Diamond Princess	86	86
Djibouti	86	86
Dominica	86	86
Dominican Republic	86	86
Ecuador	86	86
Egypt	86	86
El Salvador	86	86
Equatorial Guinea	86	86
Eritrea	86	86
Estonia	86	86
Eswatini	86	86

	death	total
Ethiopia	86	86
Fiji	86	86
Finland	86	86
France	946	946
Gabon	86	86
Gambia	86	86
Georgia	86	86
Germany	86	86
Ghana	86	86
Greece	86	86
Grenada	86	86
Guatemala	86	86
Guinea	86	86
Guinea-Bissau	86	86
Guyana	86	86
Haiti	86	86
Holy See	86	86
Honduras	86	86
Hungary	86	86
Iceland	86	86
India	86	86
Indonesia	86	86
Iran	86	86
Iraq	86	86
Ireland	86	86
Israel	86	86
Italy	86	86
Jamaica	86	86
Japan	86	86
Jordan	86	86
Kazakhstan	86	86
Kenya	86	86
Korea, South	86	86
Kosovo	86	86
Kuwait	86	86
Kyrgyzstan	86	86
Laos	86	86
Latvia	86	86
Lebanon	86	86
Liberia	86	86
Libya	86	86
Liechtenstein	86	86
Lithuania	86	86
Luxembourg	86	86
Madagascar	86	86
Malawi	86	86
Malaysia	86	86
Maldives	86	86
Mali	86	86
Malta	86	86
Mauritania	86	86
Mauritius	86	86

	death	total
Mexico	86	86
Moldova	86	86
Monaco	86	86
Mongolia	86	86
Montenegro	86	86
Morocco	86	86
Mozambique	86	86
MS Zaandam	86	86
Namibia	86	86
Nepal	86	86
Netherlands	430	430
New Zealand	86	86
Nicaragua	86	86
Niger	86	86
Nigeria	86	86
North Macedonia	86	86
Norway	86	86
Oman	86	86
Pakistan	86	86
Panama	86	86
Papua New Guinea	86	86
Paraguay	86	86
Peru	86	86
Philippines	86	86
Poland	86	86
Portugal	86	86
Qatar	86	86
Romania	86	86
Russia	86	86
Rwanda	86	86
Saint Kitts and Nevis	86	86
Saint Lucia	86	86
Saint Vincent and the Grenadines	86	86
San Marino	86	86
Sao Tome and Principe	86	86
Saudi Arabia	86	86
Senegal	86	86
Serbia	86	86
Seychelles	86	86
Sierra Leone	86	86
Singapore	86	86
Slovakia	86	86
Slovenia	86	86
Somalia	86	86
South Africa	86	86
South Sudan	86	86
Spain	86	86
Sri Lanka	86	86
Sudan	86	86
Suriname	86	86
Sweden	86	86
Switzerland	86	86

	death	total
Syria	86	86
Taiwan*	86	86
Tanzania	86	86
Thailand	86	86
Timor-Leste	86	86
Togo	86	86
Trinidad and Tobago	86	86
Tunisia	86	86
Turkey	86	86
Uganda	86	86
Ukraine	86	86
United Arab Emirates	86	86
United Kingdom	946	946
Uruguay	86	86
US	86	86
US_state	279930	279930
Uzbekistan	86	86
Venezuela	86	86
Vietnam	86	86
West Bank and Gaza	86	86
Western Sahara	86	86
Yemen	86	86
Zambia	86	86
Zimbabwe	86	86

```
# 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_death

##------------------------------------------
## 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)
##-------------------------------------------
# TODO Add population data for non US cities/regions
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$Populati
Corona_Cases$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	1586
Alaska	227
Arizona	476
Arkansas	1501
California	1928
Colorado	1502
Connecticut	277

Var1	Freq
Delaware	103
Diamond Princess	31
District of Columbia	32
Florida	1849
Georgia	3767
Grand Princess	32
Guam	32
Hawaii	171
Idaho	712
Illinois	1846
Indiana	2237
Iowa	1706
Kansas	1231
Kentucky	1936
Louisiana	1680
Maine	417
Maryland	713
Massachusetts	525
Michigan	1861
Minnesota	1564
Mississippi	2065
Missouri	1873
Montana	622
Nebraska	799
Nevada	289
New Hampshire	314
New Jersey	753
New Mexico	576
New York	1771
North Carolina	2329
North Dakota	605
Northern Mariana Islands	17
Ohio	2056
Oklahoma	1367
Oregon	851
Pennsylvania	1720
Puerto Rico	32
Rhode Island	184
South Carolina	1259
South Dakota	810
Tennessee	2192
Texas	3965
Utah	479
Vermont	405
Virgin Islands	32
Virginia	2690
Washington	1240
West Virginia	816
Wisconsin	1484
Wyoming	451

```
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?

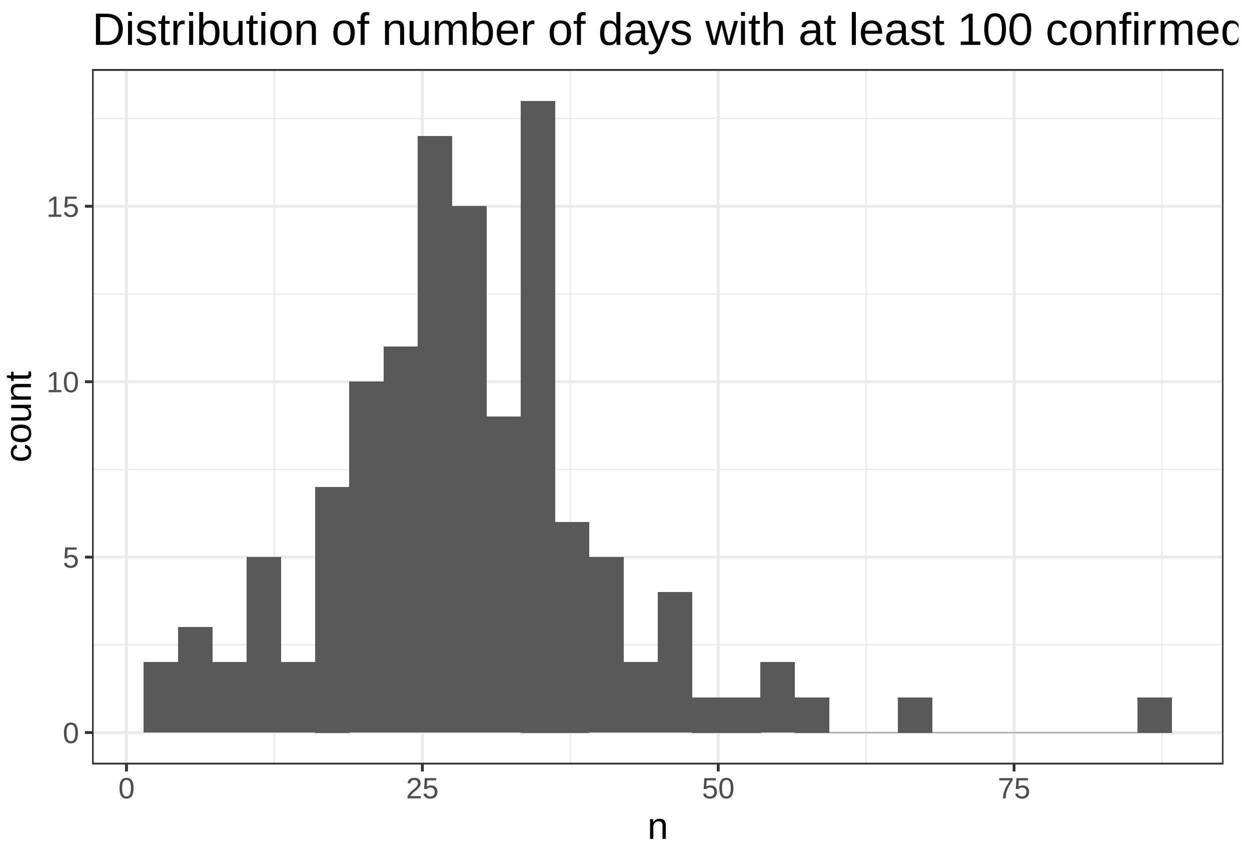
Plot # of cases vs time
* For each geographical set:
* comparative longitudinal case trend (absolute & log scale)
* comparative longitudinal mortality trend
* death vs total correlation

question	dataset	x	y	color	facet	pch	dimentions
comparative_longitudinal_case_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

```
# 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
```

```r
#total/death on same plot-  diffeer 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`.
```




```r
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	7196
China	86
Diamond Princess	67
Korea, South	57
Japan	56
Italy	54
Iran	51
Singapore	48

Country.Region	n
France	47
Germany	47
Spain	46
US	45
Switzerland	43
United Kingdom	43
Belgium	42
Netherlands	42
Norway	42
Sweden	42
Austria	40
Malaysia	39
Australia	38
Bahrain	38
Denmark	38
Canada	37
Qatar	37
Iceland	36
Brazil	35
Czechia	35
Finland	35
Greece	35
Iraq	35
Israel	35
Portugal	35
Slovenia	35
Egypt	34
Estonia	34
India	34
Ireland	34
Kuwait	34
Philippines	34
Poland	34
Romania	34
Saudi Arabia	34
Indonesia	33
Lebanon	33
San Marino	33
Thailand	33
Chile	32
Pakistan	32
Luxembourg	31
Peru	31
Russia	31
Ecuador	30
Slovakia	30
South Africa	30
United Arab Emirates	30
Armenia	29
Colombia	29
Croatia	29
Mexico	29

Country.Region	n
Panama	29
Serbia	29
Taiwan*	29
Turkey	29
Argentina	28
Bulgaria	28
Latvia	28
Algeria	27
Costa Rica	27
Dominican Republic	27
Hungary	27
Uruguay	27
Andorra	26
Bosnia and Herzegovina	26
Jordan	26
Lithuania	26
Morocco	26
New Zealand	26
North Macedonia	26
Vietnam	26
Albania	25
Cyprus	25
Malta	25
Moldova	25
Brunei	24
Burkina Faso	24
Sri Lanka	24
Tunisia	24
Ukraine	23
Azerbaijan	22
Ghana	22
Kazakhstan	22
Oman	22
Senegal	22
Venezuela	22
Afghanistan	21
Cote d'Ivoire	21
Cuba	20
Mauritius	20
Uzbekistan	20
Cambodia	19
Cameroon	19
Honduras	19
Nigeria	19
West Bank and Gaza	19
Belarus	18
Georgia	18
Bolivia	17
Kosovo	17
Kyrgyzstan	17
Montenegro	17
Congo (Kinshasa)	16

Country.Region	n
Kenya	15
Niger	14
Guinea	13
Rwanda	13
Trinidad and Tobago	13
Paraguay	12
Bangladesh	11
Djibouti	9
El Salvador	8
Guatemala	7
Madagascar	6
Mali	5
Congo (Brazzaville)	2
Jamaica	2

```
# 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

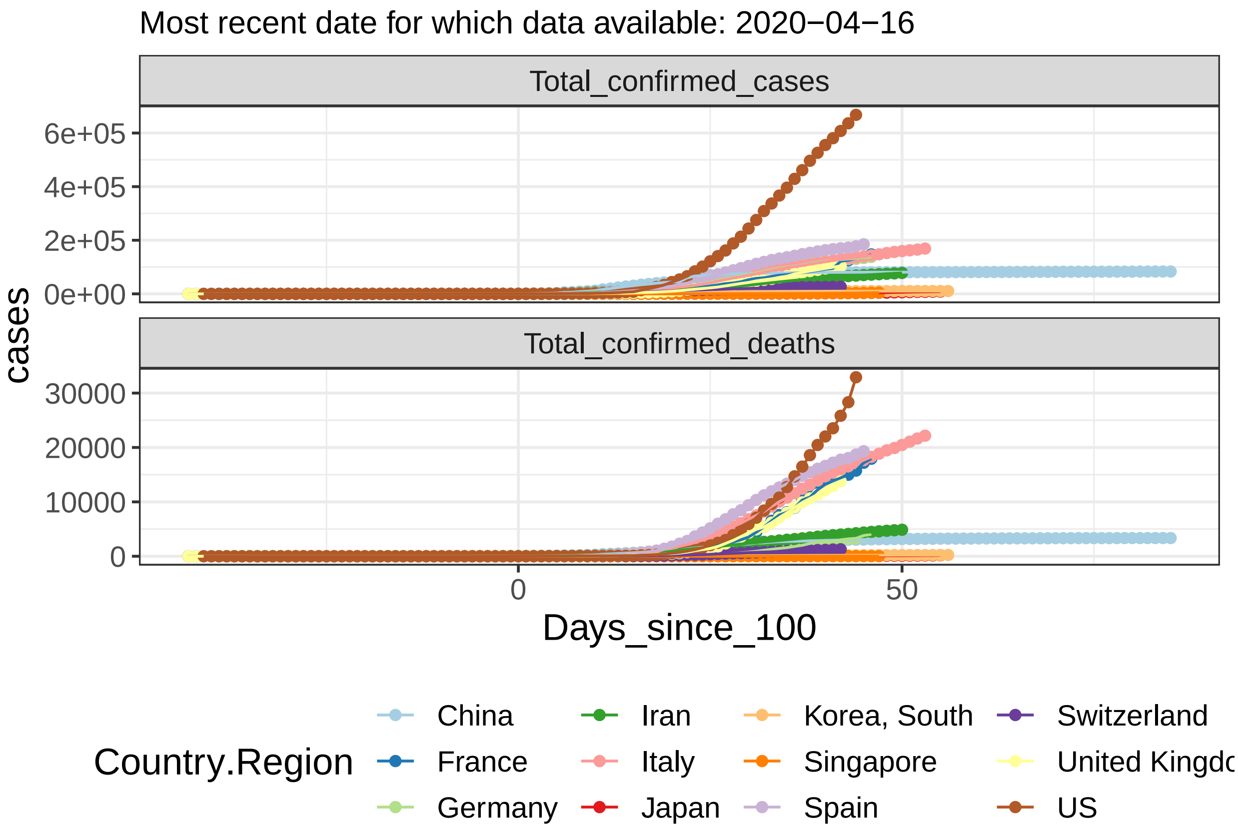
Country.Region	n
China	86
Korea, South	57
Japan	56
Italy	54
Iran	51
Singapore	48
France	47
Germany	47
Spain	46
US	45
Switzerland	43
United Kingdom	43

```
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
```

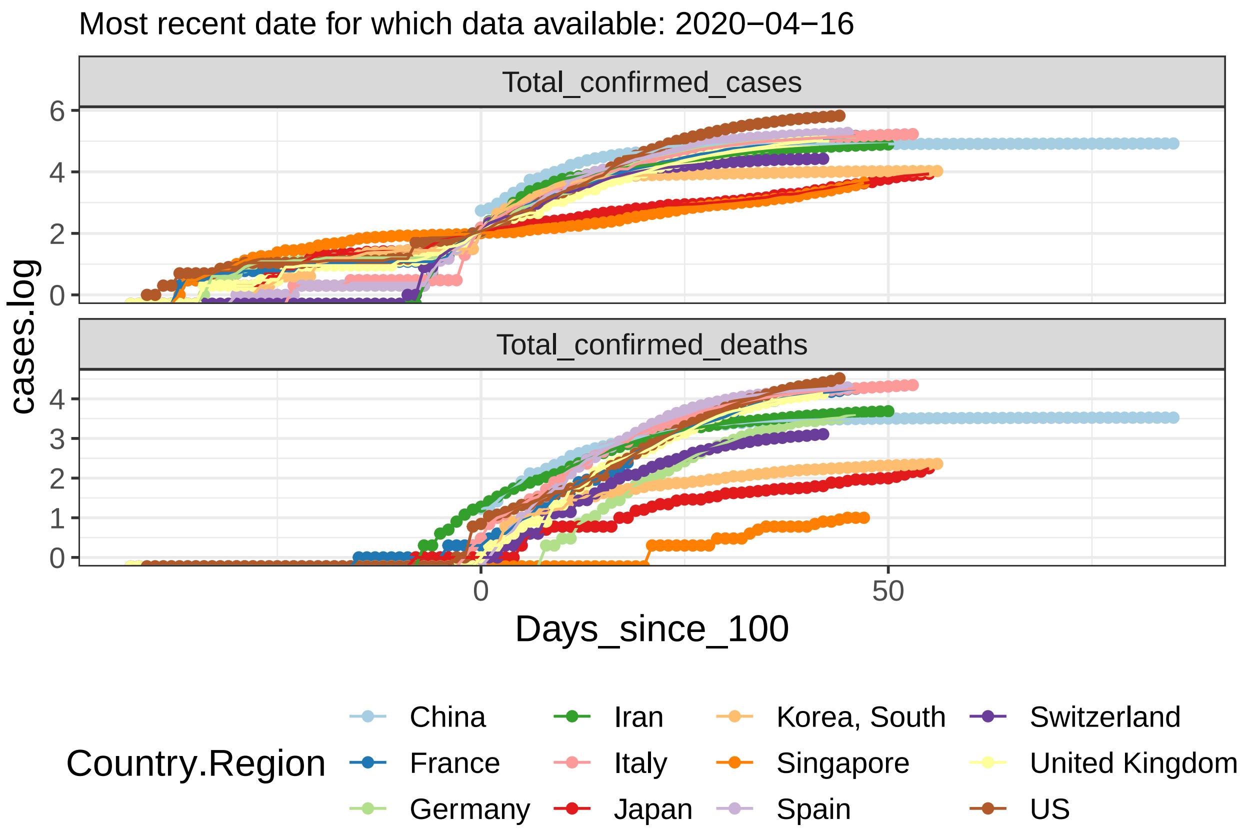
```r
melt_arg_list<-list(variable.name = "case_type",value.name = "cases",measure.vars = c("Total_confirmed_c
melt_arg_list$data=NULL


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"))
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)
    )
```



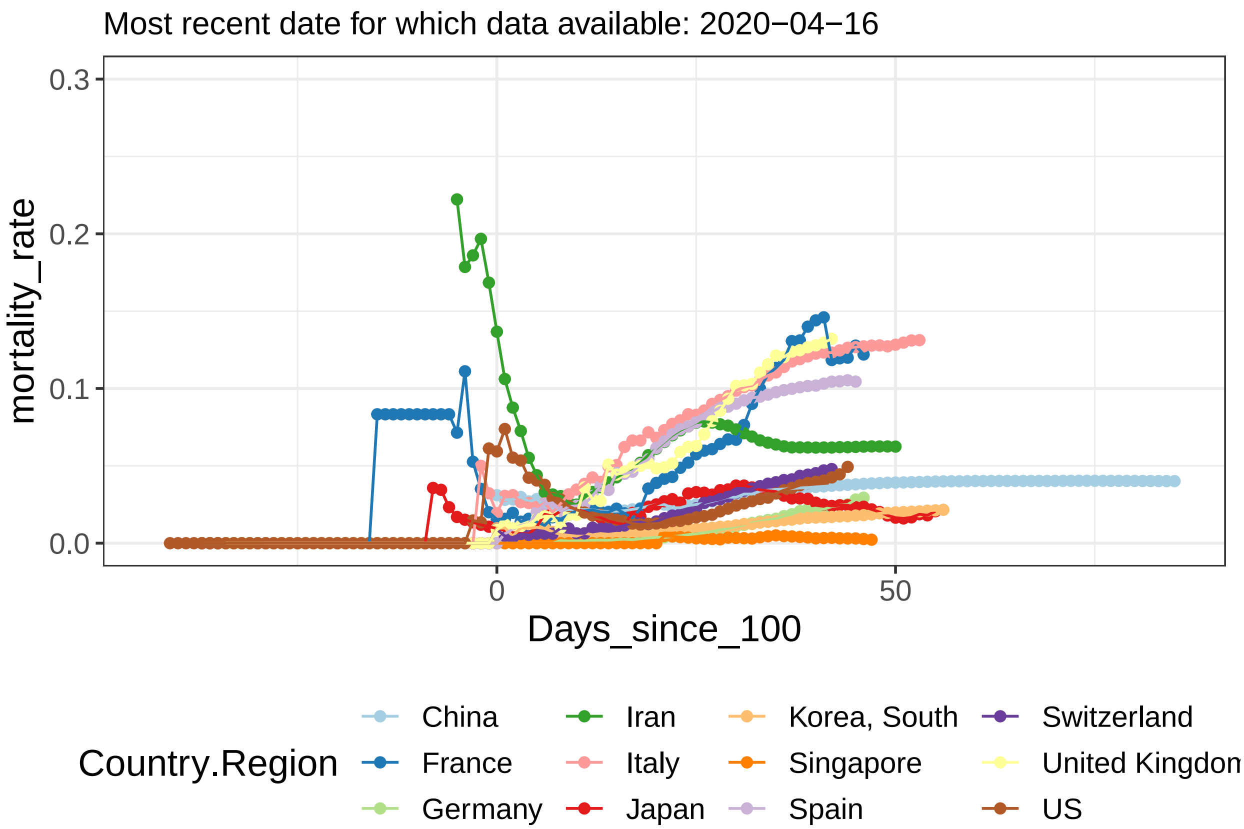

```
(Corona_Cases.world.loglong.plot<-baseplot.world+
    geom_point(data=Corona_Cases.world.long,aes(y=cases.log))+
    geom_line(data=Corona_Cases.world.long,aes(y=cases.log))+
    facet_wrap(~case_type,scales = "free_y",ncol=1)+
    ggtitle(timestamp_plot.world))
```



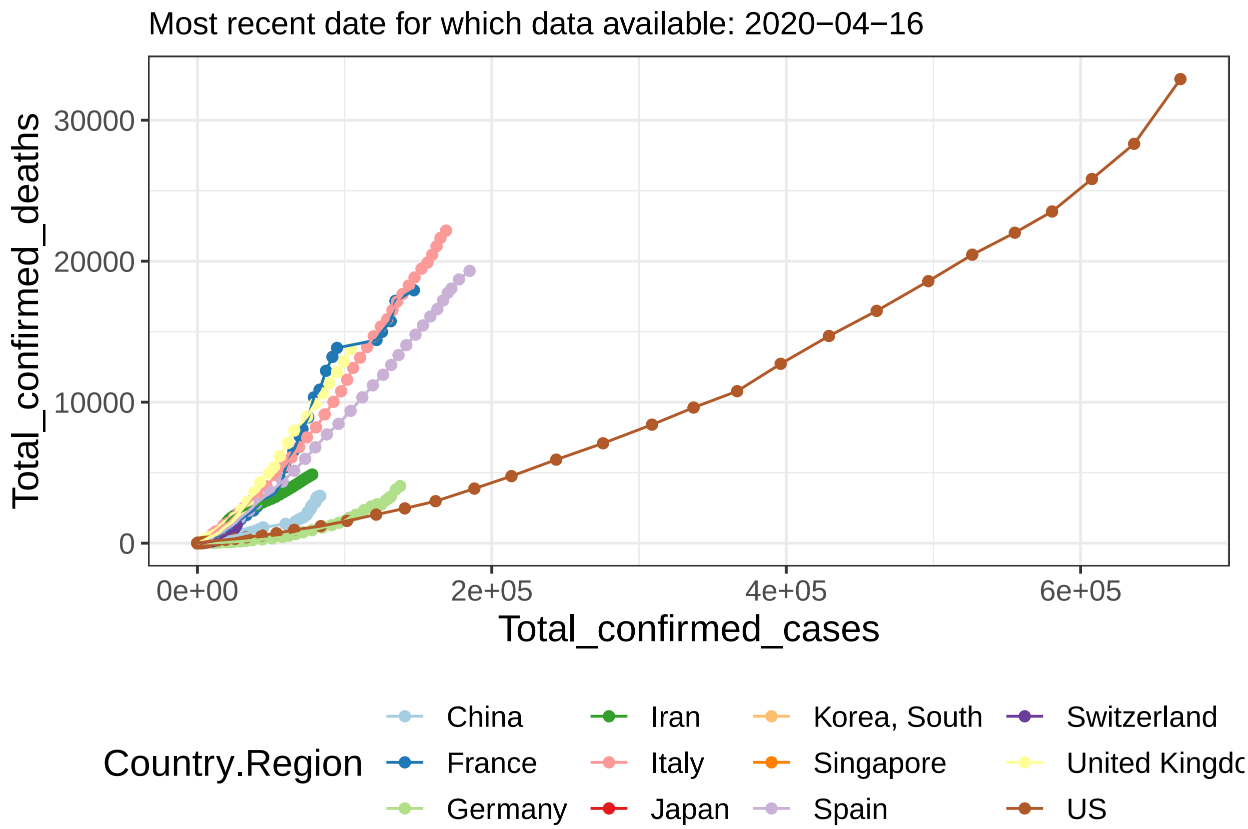

```
##//////////////////////////
### Plot Longitudinal mortality rate

(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_confirme
  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)
```

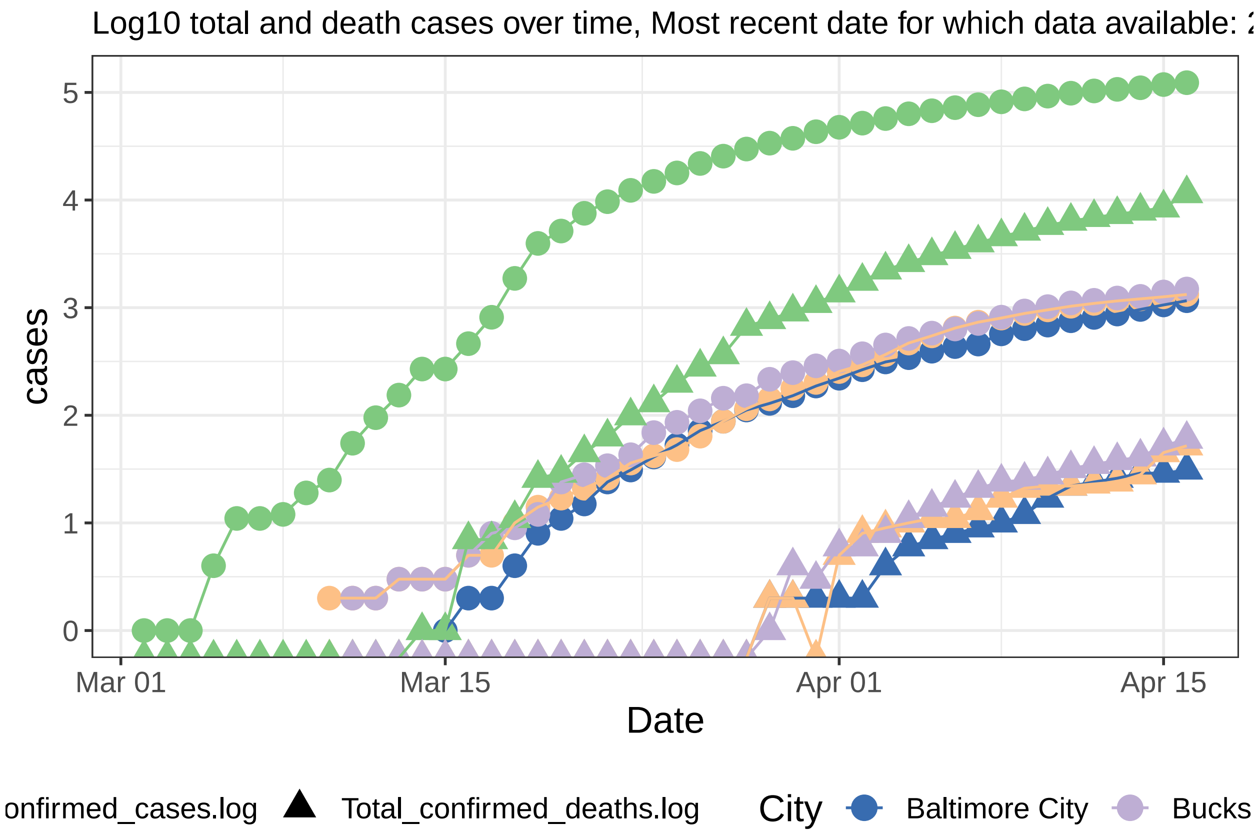
```
## Warning: Removed 100 rows containing missing values (geom_point).

## 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"
```

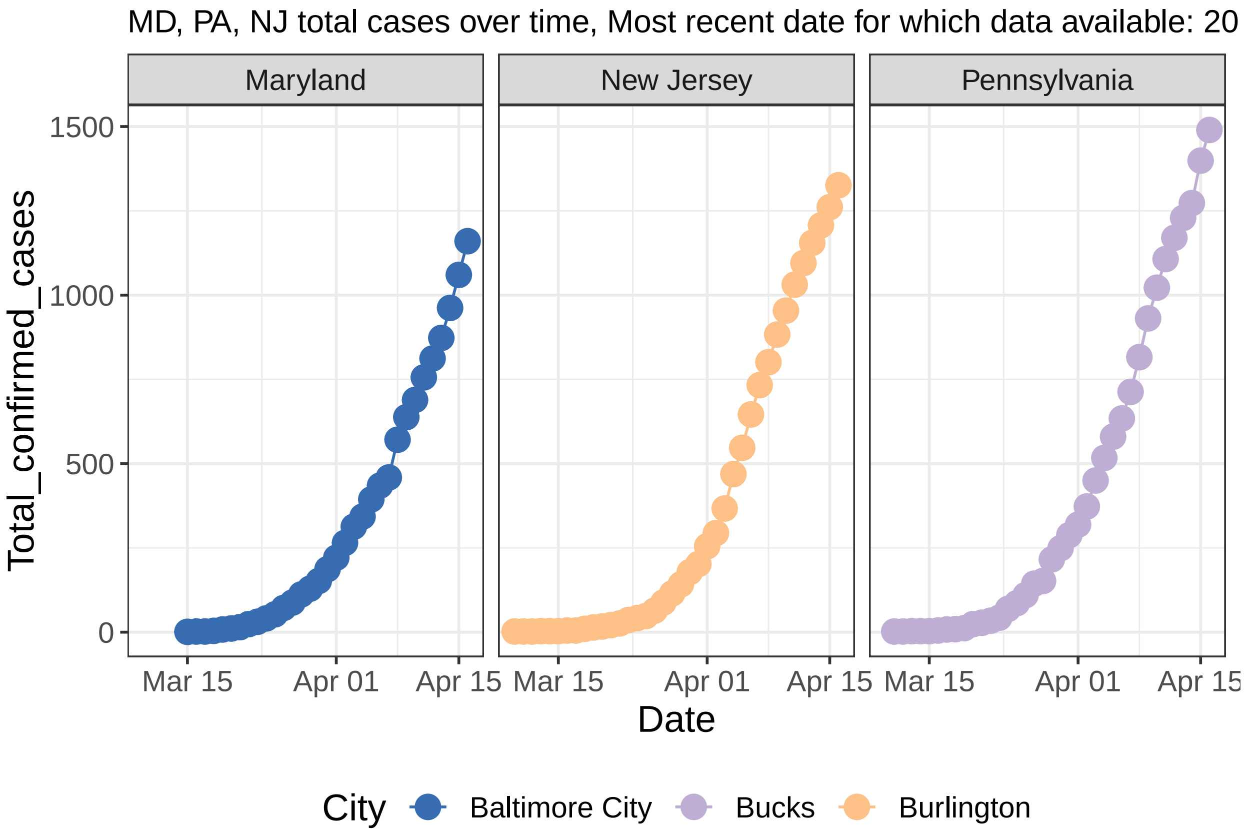
```
##-----------------------------------------
## Plot US State Data
##-----------------------------------------

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))
```

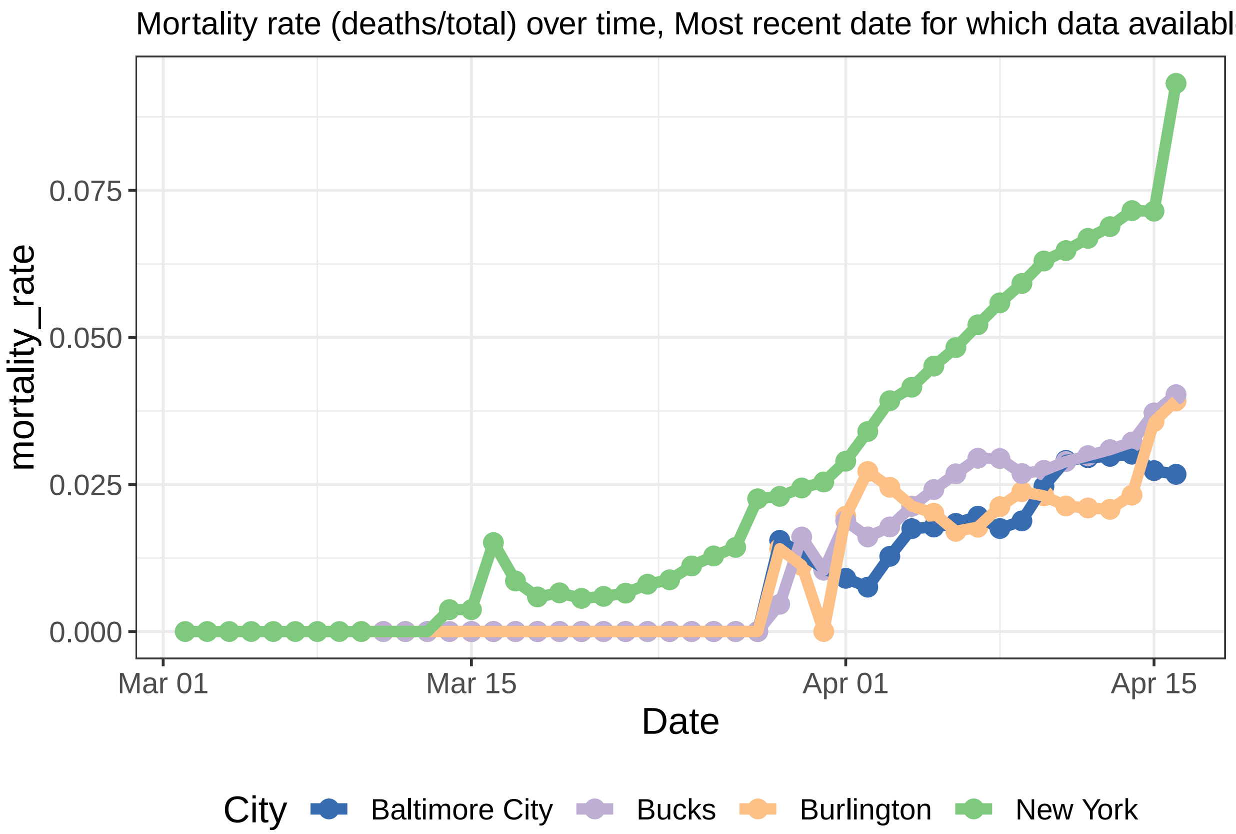



```
(Corona_Cases.city.long.plot<-ggplot(filter(Corona_Cases.US.plotdata,Province.State !="New York"),aes(x
  geom_point(size=4)+
  geom_line()+
```

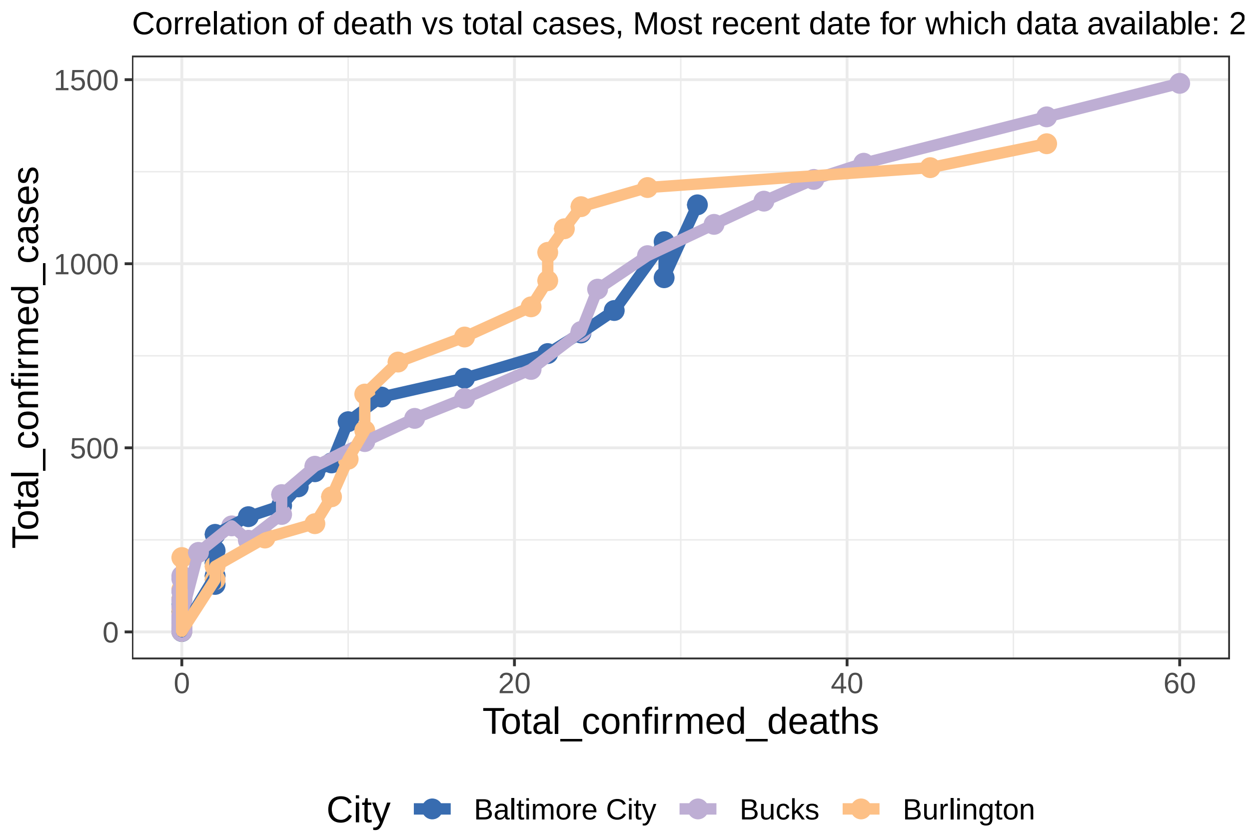
```
  default_theme+
  facet_grid(~Province.State,scales = "free_y")+
  ggtitle(paste("MD, PA, NJ total cases over time,",timestamp_plot))+
  theme(legend.position = "bottom",plot.title = element_text(size=12))+
  scale_color_manual(values = city_colors))
```



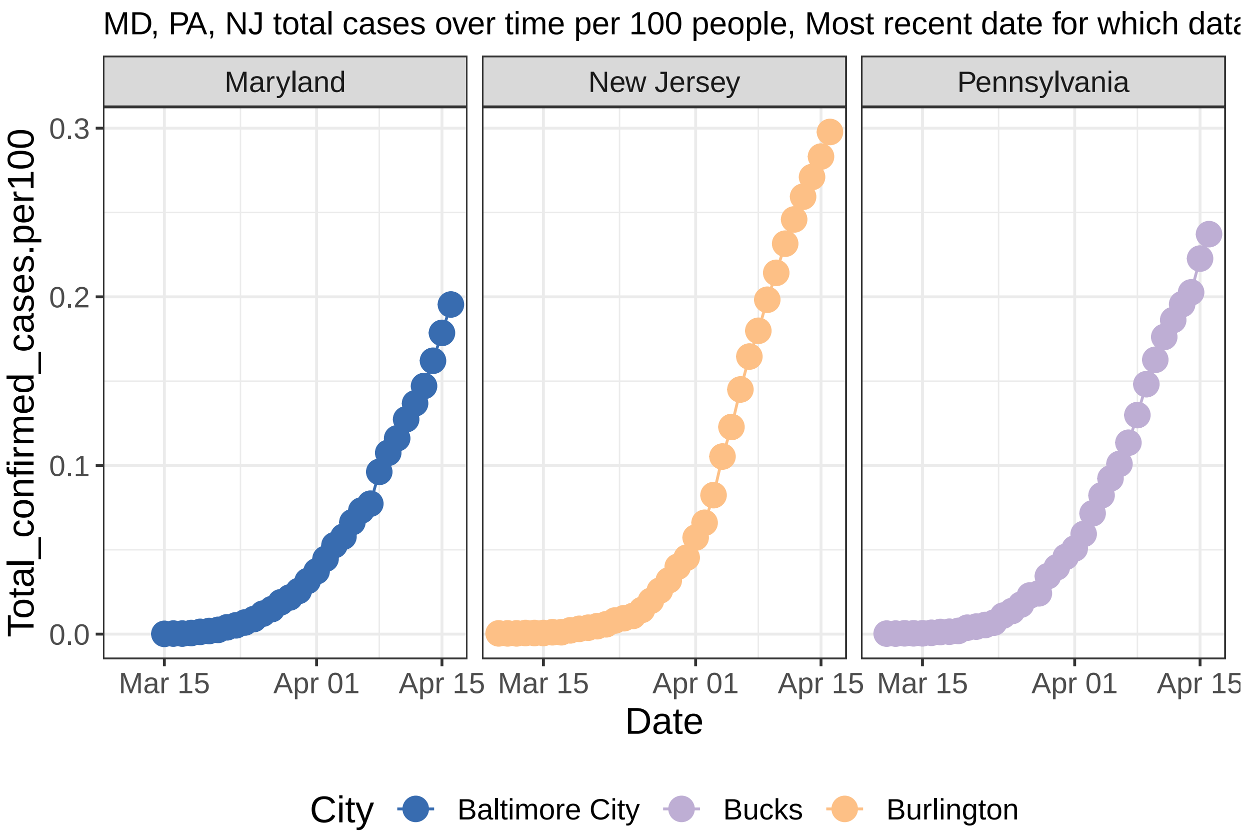

```
(Corona_Cases.city.mortality.plot<-ggplot(Corona_Cases.US.plotdata,aes(x=Date,y=mortality_rate,col=City
  geom_point(size=3)+
  geom_line(size=2)+
  default_theme+
  ggtitle(paste("Mortality rate (deaths/total) over time,",timestamp_plot))+
  theme(legend.position = "bottom",plot.title = element_text(size=12))+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

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
##--------------------------------------------
```

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

# Formula for # of cases by x days
paste0("log10_total_cases = ",slope,"*days + ",intercept)
paste0("total_cases = 10^(",slope,"*days + ",intercept,")")
#Days untill... cases:
# 2.5k, 5k and 1M:
paste0("2.5k cases is ",(log(2.5E5,10) - intercept)/slope," days")
paste0("5k cases is ",(log(5E5,10)- intercept)/slope," days")
paste0("1M cases is ",(log(1E6,10)- intercept)/slope," days")

head(filter(Corona_Cases.raw,Country.Region=="US"))
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"
names(Corona_Cases)

Corona_Cases_wprediction<-rbind.fill(Corona_Cases.US,data.frame(Code="USA",type="MAR26_prediction",predi

Corona_Cases.US.prediction<-Corona_Cases_wprediction
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,Total
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_confirmed_
prediction_day_x_nextweek<-prediction_model(m = model_fit_x$coefficients[2],b = model_fit_x$coefficient
prediction_day_x_nextweek$type<-"Predicted"
acutal_day_x_nextweek<-filter(Corona_Cases.US,Days_since_100 %in% day_x_nextweek) %>% select(c(Days_sinc
acutal_day_x_nextweek$type<-"Historical"
historical_model_predictions.i<-data.frame(day_x=day_x,rbind(acutal_day_x_nextweek,prediction_day_x_next
historical_model_predictions<-rbind(historical_model_predictions.i,historical_model_predictions)
}

historical_model_predictions.withHx<-rbind.fill(historical_model_predictions,data.frame(Corona_Cases.US
historical_model_predictions.withHx$Total_confirmed_cases.log2<-log(historical_model_predictions.withHx$
#TODO: fix case_type.. are we predicting deaths too?
#TODO: better analysis of death rate!
(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")+
```

```r
      #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)
```

```r
##--------------------------------------------
## filter input_data1
##--------------------------------------------
input_data1.filter<-fitler(input_data1,col1=="foo")
##--------------------------------------------

##--------------------------------------------
## sub question 1
##--------------------------------------------
table(input_data1.filter$col<5)
##--------------------------------------------

##--------------------------------------------
## sub question 2
##--------------------------------------------
table(input_data1.filter$col<10)
##--------------------------------------------

##--------------------------------------------
## plot data
##--------------------------------------------
(input_data1.filter.plot<-ggplot(input_data1.filter,aes(x=col1,y=col2.log))+
   geom_point()+
   default_plot_theme)
write_plot(input_data1.filter.plot,wd=results_dir)
##--------------------------------------------
results_dir
```

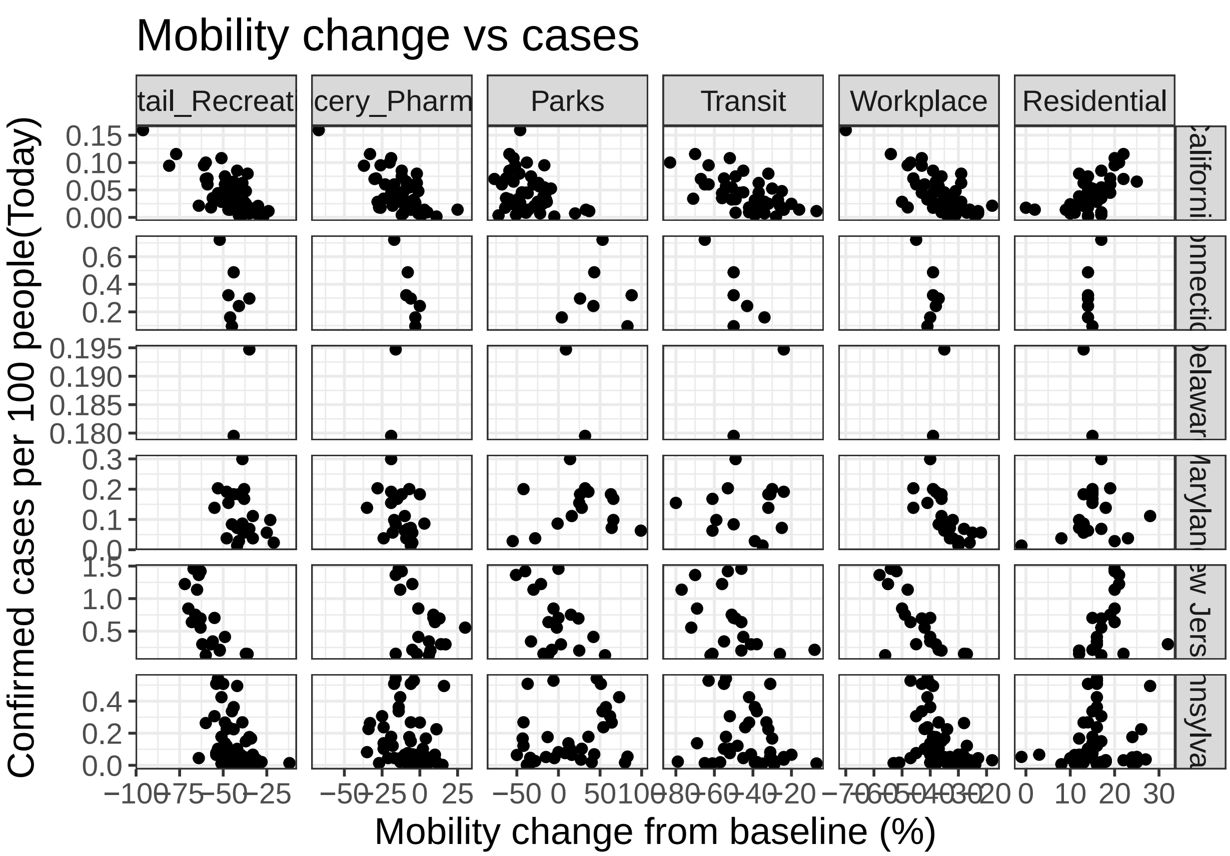
Q3: What is the effect on social distancing, descreased mobility on case load?

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/

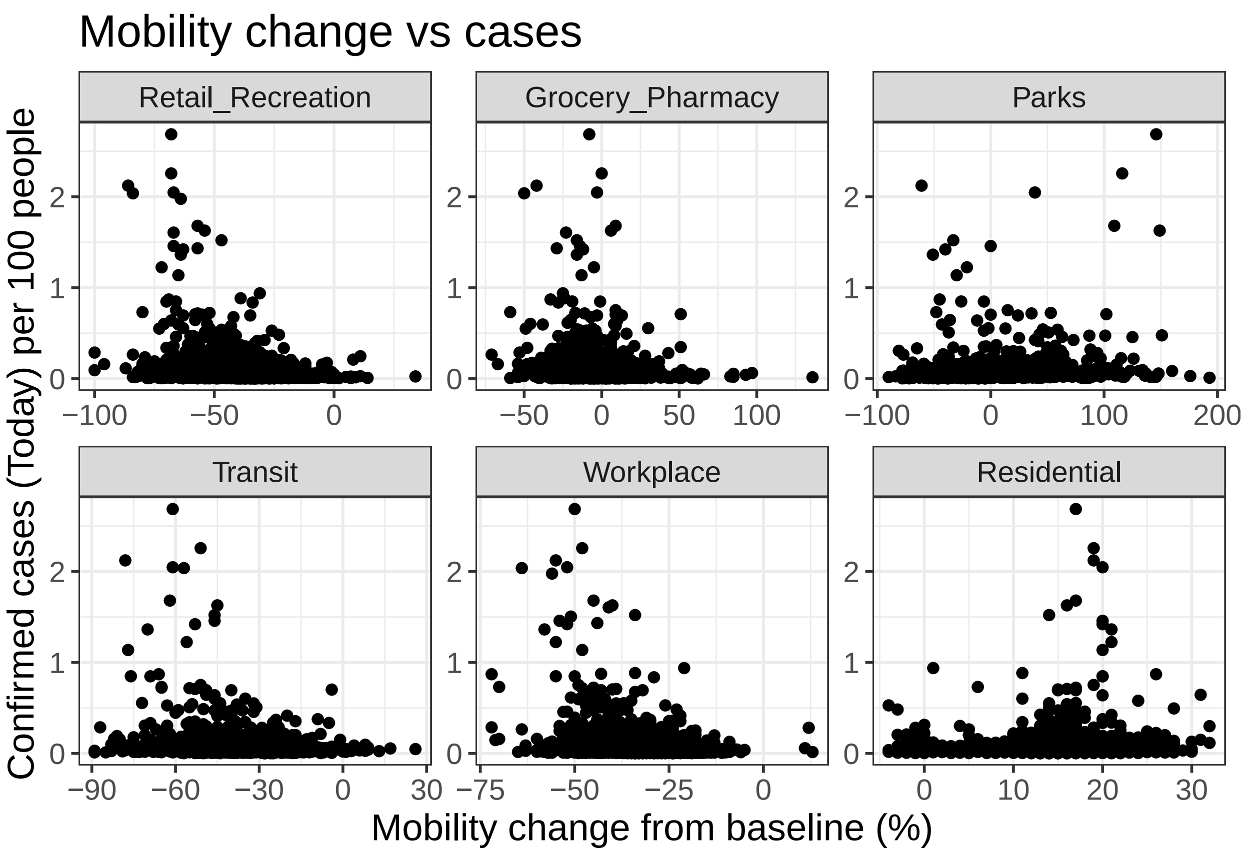
```r
# See pre-processing section for script on gathering mobility data

# UNDER DEVELOPMENT
# TODO convert % to numeric in mobility data
# TODO standardize headers in mobility data
# TODO standardize counties in mobility data to JHU source
# TODO normalize case load to population for mobility data
# TODO automate get_mobility.py script so most recent data is availble
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 = ""))
```

```r
##--------------------------------------------
## 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
#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","Ca
  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 8: 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
Alaska	Residential	0.9751827	13.0
Vermont	Parks	0.9369429	-35.5
South Dakota	Parks	0.9113715	-26.0
Connecticut	Grocery_Pharmacy	-0.8921920	-6.0
New Hampshire	Parks	0.8894686	-20.0
Hawaii	Transit	0.8678700	-89.0
Alaska	Grocery_Pharmacy	-0.8491403	-7.0
Utah	Workplace	-0.8254821	-33.0
Massachusetts	Workplace	-0.8246232	-39.0

Province.State	variable	cor	median_change
Hawaii	Parks	0.8240971	-72.0
Rhode Island	Workplace	-0.7899509	-39.5
Connecticut	Transit	-0.7846991	-50.0
North Dakota	Residential	-0.7671769	17.0
New Mexico	Parks	0.7648296	-31.5
Hawaii	Workplace	-0.7535008	-46.0
Utah	Retail_Recreation	-0.7510837	-36.0
Utah	Grocery_Pharmacy	-0.7378303	-3.0
New Jersey	Workplace	-0.7243558	-44.0
California	Retail_Recreation	-0.7198130	-44.0
Kansas	Parks	0.7142568	72.0
Maryland	Workplace	-0.7139322	-35.0
Utah	Transit	-0.7090547	-18.0
California	Workplace	-0.7032442	-36.0
New Jersey	Retail_Recreation	-0.6684091	-62.5
Massachusetts	Retail_Recreation	-0.6642412	-44.0
New York	Workplace	-0.6627340	-34.5
Vermont	Grocery_Pharmacy	-0.6599694	-25.0
North Dakota	Retail_Recreation	-0.6561931	-43.5
Nevada	Transit	-0.6553021	-20.0
California	Grocery_Pharmacy	-0.6485176	-12.0
Connecticut	Residential	0.6464052	14.0
New York	Retail_Recreation	-0.6227696	-46.0
California	Residential	0.6132272	14.0
Maine	Transit	-0.6038220	-50.0
California	Transit	-0.6031857	-42.0
Rhode Island	Residential	-0.5961314	18.5
North Dakota	Transit	0.5699363	-48.0
Montana	Workplace	-0.5683986	-40.5
Nevada	Retail_Recreation	-0.5681384	-43.0
Montana	Retail_Recreation	-0.5640849	-51.0
West Virginia	Parks	0.5629441	-27.0
Connecticut	Workplace	-0.5582775	-39.0
Massachusetts	Grocery_Pharmacy	-0.5494303	-7.0
Alaska	Workplace	-0.5487306	-34.0
Rhode Island	Retail_Recreation	-0.5461570	-45.0
Montana	Transit	-0.5326846	-41.0
Utah	Residential	-0.5179984	12.0
Idaho	Workplace	-0.5178822	-29.5
Montana	Parks	-0.5112347	-58.0
New Jersey	Parks	-0.5019243	-6.0
Hawaii	Residential	0.4967913	19.0
Maine	Workplace	-0.4942295	-30.0
Kansas	Grocery_Pharmacy	-0.4902249	-14.0
Maine	Parks	0.4730078	-31.0
Nebraska	Grocery_Pharmacy	-0.4726597	0.0
Minnesota	Parks	0.4690157	-3.5
New Jersey	Grocery_Pharmacy	-0.4632024	2.5
Idaho	Transit	-0.4597707	-30.0
Connecticut	Retail_Recreation	-0.4594678	-45.0
Massachusetts	Transit	-0.4531303	-45.0
Montana	Residential	0.4519305	14.0

Province.State	variable	cor	median_change
Vermont	Residential	0.4518083	11.5
New York	Parks	0.4419450	20.0
Virginia	Transit	-0.4413524	-33.0
Arkansas	Parks	-0.4364127	-12.0
Pennsylvania	Workplace	-0.4305236	-36.0
Colorado	Workplace	-0.4265285	-39.0
Idaho	Grocery_Pharmacy	-0.4256137	-4.0
New Jersey	Transit	-0.4242579	-50.5
New Mexico	Residential	0.4193898	13.5
Virginia	Retail_Recreation	-0.4188956	-35.0
North Dakota	Grocery_Pharmacy	-0.4182110	-9.5
Colorado	Residential	0.4158610	14.0
New York	Transit	-0.4150487	-48.0
Rhode Island	Parks	0.4091070	52.0
Michigan	Workplace	-0.4067618	-40.0
Pennsylvania	Retail_Recreation	-0.4023266	-45.0
Florida	Parks	-0.3937549	-43.0
Arizona	Grocery_Pharmacy	-0.3919405	-15.0
Hawaii	Grocery_Pharmacy	0.3846654	-34.0
Oregon	Parks	0.3842822	16.5
Idaho	Retail_Recreation	-0.3836365	-41.0
Montana	Grocery_Pharmacy	-0.3798480	-16.0
Kansas	Retail_Recreation	-0.3761978	-39.0
Utah	Parks	-0.3654861	0.0
Colorado	Retail_Recreation	-0.3643580	-44.0
Rhode Island	Grocery_Pharmacy	0.3637254	-7.5
Colorado	Transit	-0.3600682	-36.0
Vermont	Retail_Recreation	0.3547722	-57.0
Arizona	Transit	0.3524131	-38.0
Maryland	Retail_Recreation	-0.3520347	-39.0
Illinois	Transit	-0.3501367	-31.0
South Dakota	Transit	-0.3443918	-40.0
Mississippi	Parks	0.3434226	-25.0
Alaska	Retail_Recreation	0.3392193	-39.0
Colorado	Parks	-0.3350256	2.0
Maryland	Grocery_Pharmacy	-0.3333357	-10.0
Idaho	Parks	0.3314101	-22.0
New Mexico	Retail_Recreation	-0.3304166	-42.0
Virginia	Workplace	-0.3270881	-31.5
Alabama	Workplace	-0.3257901	-29.0
Florida	Transit	-0.3244274	-49.0
Texas	Transit	0.3232985	-42.0
Maine	Grocery_Pharmacy	-0.3226854	-10.5
Washington	Transit	-0.3219955	-33.5
Colorado	Grocery_Pharmacy	-0.3190343	-17.0
New Hampshire	Grocery_Pharmacy	-0.3180236	-6.0
Florida	Residential	0.3158151	14.0
Arizona	Residential	0.3110955	13.0
Maine	Retail_Recreation	-0.3061413	-41.5
California	Parks	-0.3051809	-38.0
Arkansas	Retail_Recreation	-0.2930966	-30.0
North Carolina	Retail_Recreation	-0.2904058	-33.0

Province.State	variable	cor	median_change
North Dakota	Parks	0.2900720	-34.0
New Jersey	Residential	0.2891961	18.0
Iowa	Residential	-0.2891354	13.0
South Carolina	Residential	0.2890789	12.0
Oregon	Residential	0.2884220	10.5
Florida	Workplace	-0.2855630	-33.0
New Mexico	Grocery_Pharmacy	-0.2853571	-12.0
Virginia	Grocery_Pharmacy	-0.2848121	-8.0
New York	Grocery_Pharmacy	-0.2847600	8.0
Tennessee	Retail_Recreation	-0.2845661	-30.0
Arkansas	Residential	0.2817073	12.0
Pennsylvania	Parks	0.2758113	13.0
Indiana	Grocery_Pharmacy	-0.2739350	-5.5
Wisconsin	Transit	-0.2640239	-23.5
Nevada	Workplace	-0.2613095	-40.0
Hawaii	Retail_Recreation	0.2610592	-56.0
Mississippi	Grocery_Pharmacy	-0.2597076	-8.0
Georgia	Grocery_Pharmacy	-0.2575650	-10.0
Illinois	Workplace	-0.2572536	-30.0
New Hampshire	Retail_Recreation	-0.2551458	-41.0
Iowa	Workplace	-0.2544543	-29.0
Massachusetts	Residential	0.2536914	15.0
Maryland	Residential	0.2505475	15.0
Nebraska	Residential	0.2455199	14.0
Nevada	Residential	0.2445709	17.0
Arizona	Retail_Recreation	-0.2346055	-42.5
West Virginia	Retail_Recreation	0.2282075	-38.5
North Dakota	Workplace	0.2245498	-33.5
Michigan	Retail_Recreation	-0.2237441	-53.0
Pennsylvania	Grocery_Pharmacy	-0.2218285	-6.0
Georgia	Retail_Recreation	-0.2216571	-41.0
Washington	Workplace	-0.2186041	-38.0
Tennessee	Grocery_Pharmacy	-0.2163418	6.0
Alabama	Residential	0.2161366	11.0
Michigan	Grocery_Pharmacy	-0.2125593	-11.0
Georgia	Workplace	-0.2061976	-33.5
Rhode Island	Transit	-0.2059088	-56.0
Oklahoma	Residential	0.2057739	15.0
Washington	Parks	0.2045647	-3.5
Texas	Parks	0.2010288	-42.0
Oklahoma	Retail_Recreation	0.1999559	-31.0
Wisconsin	Workplace	-0.1991779	-31.0
Oklahoma	Grocery_Pharmacy	0.1982537	-0.5
Texas	Residential	-0.1970688	15.0
Nebraska	Retail_Recreation	-0.1957405	-37.5
Kentucky	Workplace	-0.1925535	-34.0
West Virginia	Grocery_Pharmacy	-0.1924103	-6.0
Wisconsin	Parks	0.1860787	51.5
Oklahoma	Workplace	-0.1815528	-30.0
Arizona	Parks	0.1809408	-44.5
Arizona	Workplace	-0.1789244	-35.0
Indiana	Retail_Recreation	-0.1785956	-38.0

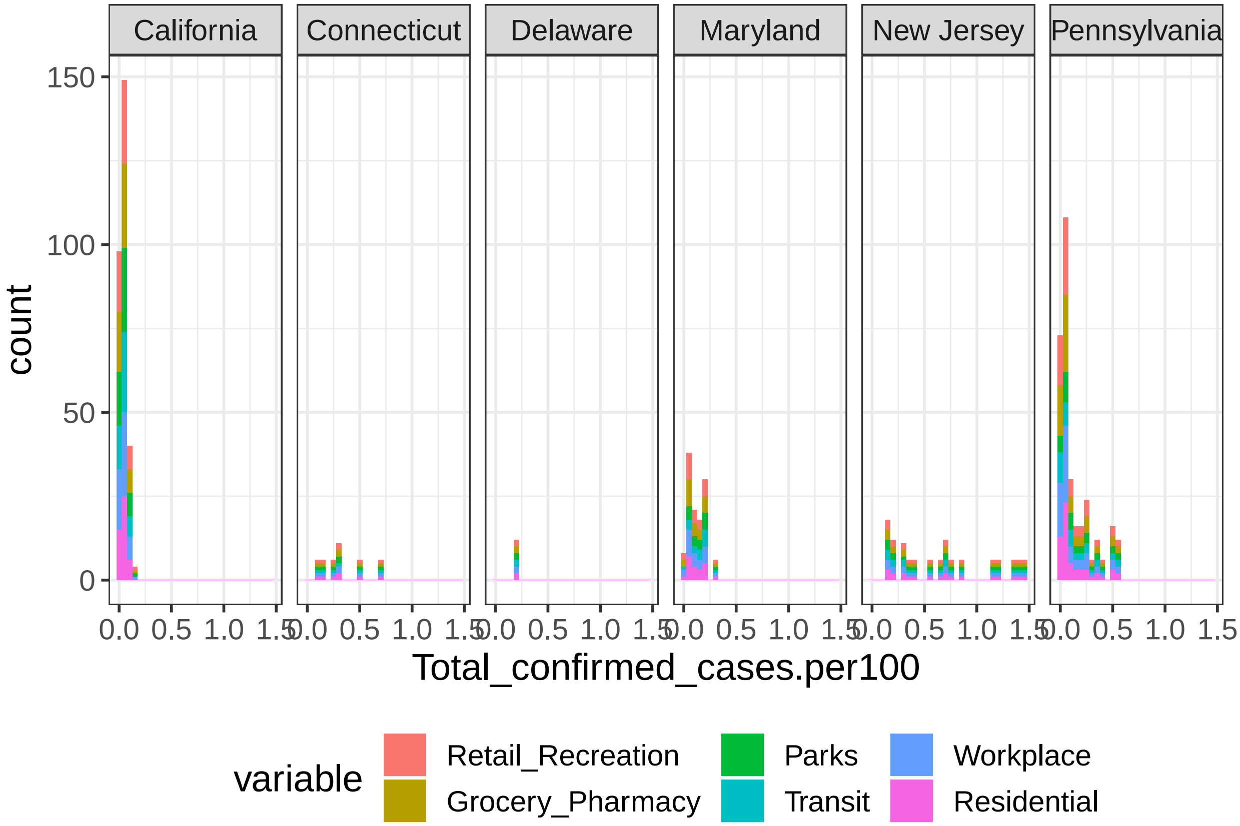
Province.State	variable	cor	median_change
Florida	Grocery_Pharmacy	-0.1781858	-14.0
South Dakota	Retail_Recreation	-0.1776820	-38.5
Kentucky	Parks	0.1764580	28.5
Mississippi	Workplace	-0.1755623	-33.0
Alabama	Transit	-0.1742344	-36.5
New Hampshire	Residential	-0.1715010	14.0
Minnesota	Workplace	0.1707732	-33.0
Alabama	Grocery_Pharmacy	-0.1706799	-2.0
Iowa	Parks	0.1650330	28.5
South Carolina	Retail_Recreation	-0.1647717	-35.0
North Carolina	Transit	0.1620756	-32.0
Kentucky	Residential	0.1617046	12.0
Kansas	Residential	0.1609134	13.0
Missouri	Transit	-0.1586251	-23.0
Pennsylvania	Transit	-0.1578144	-41.5
South Carolina	Workplace	0.1557223	-30.0
Tennessee	Workplace	-0.1557013	-31.0
Oregon	Grocery_Pharmacy	0.1552171	-7.0
New Hampshire	Workplace	-0.1499564	-37.0
Illinois	Residential	0.1480607	14.0
North Carolina	Residential	0.1468998	13.0
South Dakota	Grocery_Pharmacy	0.1460286	-9.0
Florida	Retail_Recreation	-0.1453130	-43.0
Michigan	Parks	0.1452597	33.0
Nebraska	Transit	0.1438132	-11.5
West Virginia	Workplace	0.1401526	-32.5
Idaho	Residential	-0.1399533	11.0
Vermont	Workplace	-0.1373228	-43.0
Tennessee	Parks	0.1340093	10.5
Virginia	Residential	0.1334182	14.0
Wisconsin	Grocery_Pharmacy	0.1313039	-1.5
Missouri	Retail_Recreation	-0.1311510	-37.0
Kentucky	Retail_Recreation	-0.1309355	-30.0
Arkansas	Grocery_Pharmacy	0.1299502	3.5
Minnesota	Retail_Recreation	0.1292107	-41.0
Ohio	Transit	0.1277846	-28.0
South Carolina	Parks	-0.1258213	-23.0
Wisconsin	Residential	-0.1232691	14.0
Alabama	Parks	0.1231807	-1.0
New Mexico	Workplace	-0.1206169	-34.0
Oklahoma	Parks	-0.1177523	-23.0
Minnesota	Grocery_Pharmacy	-0.1177003	-4.0
Pennsylvania	Residential	0.1174616	15.0
Illinois	Retail_Recreation	-0.1161616	-40.0
Minnesota	Transit	-0.1160218	-28.5
Washington	Residential	0.1158118	13.0
Kansas	Transit	-0.1153641	-26.5
Wisconsin	Retail_Recreation	-0.1151781	-44.0
Indiana	Workplace	-0.1146622	-34.0
Georgia	Residential	-0.1122811	13.0
Tennessee	Residential	0.1115267	12.0
New Mexico	Transit	0.1111206	-38.0

Province.State	variable	cor	median_change
Ohio	Workplace	-0.1086989	-35.0
Missouri	Grocery_Pharmacy	-0.1083840	2.0
Arkansas	Workplace	-0.1080422	-26.0
Illinois	Grocery_Pharmacy	-0.1069454	2.0
North Carolina	Parks	0.1056213	7.0
Washington	Retail_Recreation	-0.1041358	-42.0
West Virginia	Transit	-0.1036597	-45.0
Illinois	Parks	0.0997447	26.5
Nebraska	Workplace	-0.0990387	-32.5
Maine	Residential	-0.0955213	11.0
Kansas	Workplace	-0.0928691	-31.0
Virginia	Parks	0.0923779	6.0
Michigan	Residential	0.0921375	15.0
Ohio	Residential	0.0915657	14.0
Maryland	Parks	0.0872619	27.0
Missouri	Workplace	0.0860893	-28.5
Texas	Retail_Recreation	-0.0852700	-40.0
South Carolina	Transit	-0.0842450	-45.0
Georgia	Transit	-0.0827364	-35.0
Kentucky	Transit	0.0809790	-31.0
Iowa	Transit	-0.0808529	-25.0
Mississippi	Residential	0.0802950	13.0
New Hampshire	Transit	-0.0761430	-57.0
Oregon	Transit	-0.0738295	-28.0
West Virginia	Residential	0.0722752	11.0
New York	Residential	0.0715833	17.5
North Carolina	Grocery_Pharmacy	0.0694109	1.0
Kentucky	Grocery_Pharmacy	-0.0631344	4.0
Nevada	Grocery_Pharmacy	-0.0624108	-11.0
Arkansas	Transit	0.0597156	-27.0
Massachusetts	Parks	-0.0596276	39.0
North Carolina	Workplace	-0.0587233	-31.0
Texas	Workplace	0.0573335	-31.0
Georgia	Parks	-0.0491592	-6.0
Indiana	Parks	-0.0483382	29.0
Michigan	Transit	0.0472364	-46.0
Iowa	Grocery_Pharmacy	-0.0454772	4.0
South Carolina	Grocery_Pharmacy	-0.0433828	1.0
Missouri	Residential	-0.0422062	13.0
Washington	Grocery_Pharmacy	-0.0374377	-7.0
Connecticut	Parks	0.0355938	43.0
Alabama	Retail_Recreation	-0.0300013	-39.0
Indiana	Residential	0.0295390	12.0
Missouri	Parks	-0.0292843	0.5
Ohio	Retail_Recreation	-0.0272361	-36.0
Ohio	Grocery_Pharmacy	0.0272273	0.0
Iowa	Retail_Recreation	-0.0252021	-37.0
Oregon	Workplace	-0.0242419	-32.0
Tennessee	Transit	0.0227562	-32.0
Oregon	Retail_Recreation	0.0199623	-41.0
Indiana	Transit	-0.0187454	-29.0
Mississippi	Retail_Recreation	-0.0170864	-40.0

Province.State	variable	cor	median_change
South Dakota	Workplace	0.0135420	-35.0
South Dakota	Residential	0.0132234	15.0
Nevada	Parks	-0.0114763	-12.5
Ohio	Parks	0.0113255	67.5
Texas	Grocery_Pharmacy	-0.0105044	-13.0
Vermont	Transit	0.0100846	-63.0
Mississippi	Transit	-0.0093391	-38.5
Maryland	Transit	-0.0081246	-39.0
Nebraska	Parks	-0.0076250	55.5
Oklahoma	Transit	-0.0037925	-26.0
Minnesota	Residential	-0.0015431	18.0
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`.
```



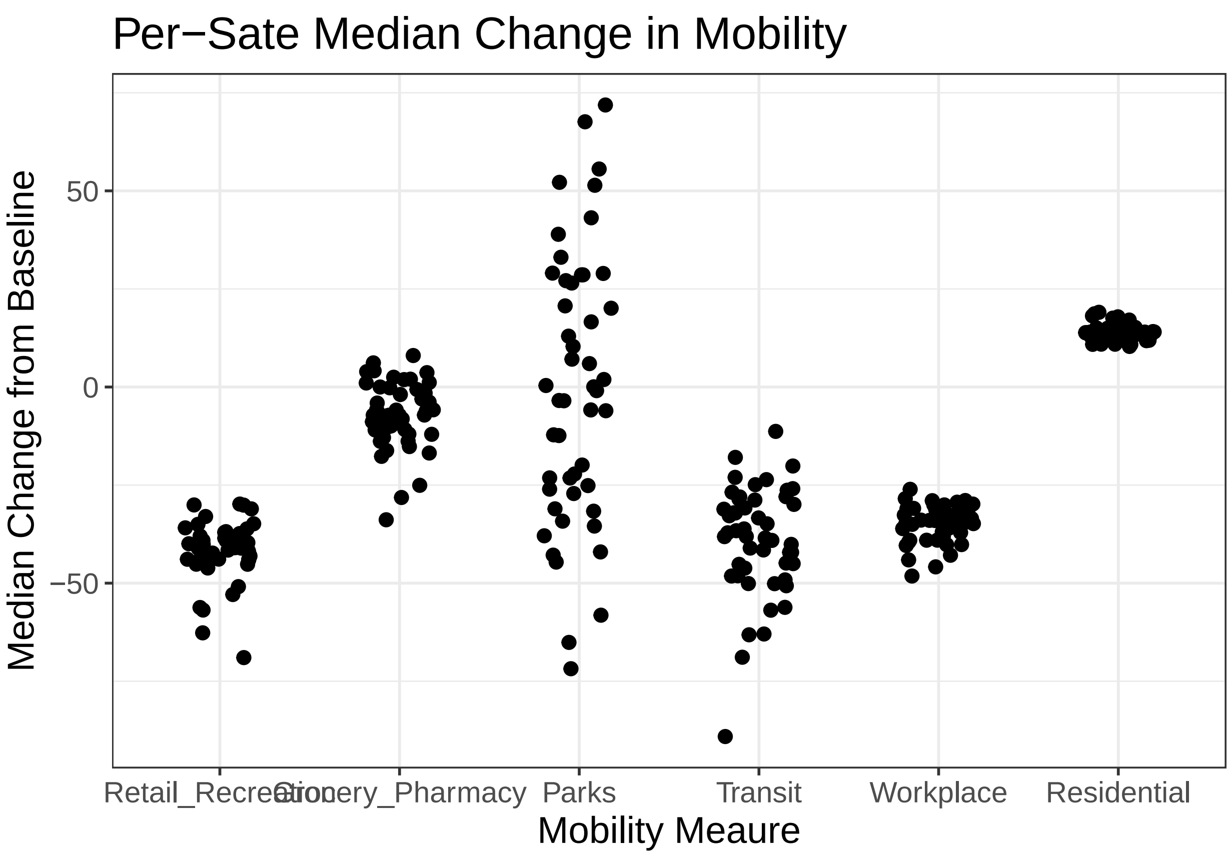

```
write_plot(mobility.plot,wd = results_dir)
```

```
## [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"
```

```
# TODO secondary question: rank greatest to least mobility

(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/"
link<-paste0(github_root,"results/Corona_Cases.world.casecor.plot.png")
section_ref<-'Q3'
plot_handle<-c("Corona_Cases.world.casecor.plot","Corona_Cases.world.long.plot")
name<-"World total & death cases, correlation"
deliverable_manifest<-data.frame(
  name=c("World total & death cases, correlation",
         "World total & death 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, correlation | Corona_Cases.world.casecor.plot | https://github.com/sbs87
```

```
## 2      World total & death cases, longitudinal | Corona_Cases.world.long.plot | https://github.com/s
row_out<-apply(tmp, 2, paste, collapse="\t\n")
```

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
World total & death cases, longitudinal	Corona_Cases.world.long.plot	https://github.com/sbs87/coronavirus/blob/master/results/Corona_Cases.world.long.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: ##—— Fri Apr 17 14:43:29 2020 ——##

Cheatsheet: http://rmarkdown.rstudio.com> # TODO * mkdir the results dir if it doesn't exist * make ggplot a dependency for plot.utils?
* automated way of downloading daily data * fix plot_utils, add dataset and documentation * Auto git mv the new data?

Sandbox

```
##TODO:
# Geographical heatmap!
```

```r
install.packages("maps")
library(maps)
library
mi_counties <- map_data("county", "pennsylvania") %>%
  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()
```

smith research

https://stevenbsmith.net