DTSA 5301: COVID Data

ZK

8/11/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Data

For this, we use the Johns Hopkins COVID data that is publically available on Github.

```
url_base <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_c

file_names <- c("time_series_covid19_confirmed_US.csv",
                "time_series_covid19_confirmed_global.csv",
                "time_series_covid19_deaths_US.csv",
                "time_series_covid19_deaths_global.csv")
urls <- str_c( url_base, file_names)
confirmed_us <- read_csv(urls[1])
confirmed_global <- read_csv(urls[2])
deaths_us <- read_csv(urls[3])
deaths_global <- read_csv(urls[4])
```

```r
# we can look at the data:
head(confirmed_us)[1:12]
```

```
## # A tibble: 6 x 12
##       UID iso2  iso3  code3  FIPS Admin2 Province_State Country_Region   Lat
##     <dbl> <chr> <chr> <dbl> <dbl> <chr>  <chr>          <chr>          <dbl>
## 1 8.40e7 US    USA     840  1001 Autau~ Alabama        US              32.5
## 2 8.40e7 US    USA     840  1003 Baldw~ Alabama        US              30.7
## 3 8.40e7 US    USA     840  1005 Barbo~ Alabama        US              31.9
## 4 8.40e7 US    USA     840  1007 Bibb   Alabama        US              33.0
## 5 8.40e7 US    USA     840  1009 Blount Alabama        US              34.0
## 6 8.40e7 US    USA     840  1011 Bullo~ Alabama        US              32.1
## # ... with 3 more variables: Long_ <dbl>, Combined_Key <chr>, `1/22/20` <dbl>
```

The data is in a WIDE format, so we will need to tidy it up. I am only going to use the **US** data, so I will not tidy the global datasets.

```r
# vars we don't want
vars_to_drop <- c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2")
# vars to keep along with the numeric data
vars_not_to_pivot <- c("Province_State", "Country_Region", "Lat", "Long_", "Combined_Key")

# make them long
long_death_us <- deaths_us %>%
  select(-vars_to_drop) %>%
  pivot_longer(cols = -c(vars_not_to_pivot, "Population"), values_to = "deaths", names_to = "date")
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(vars_to_drop)` instead of `vars_to_drop` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(vars_not_to_pivot)` instead of `vars_not_to_pivot` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```r
long_confirmed_us <- confirmed_us %>%
  select(-vars_to_drop) %>%
  pivot_longer(cols = -(vars_not_to_pivot), values_to = "confirmed", names_to = "date")

df_tidy <- full_join(long_confirmed_us, long_death_us)
```

```
## Joining, by = c("Province_State", "Country_Region", "Lat", "Long_", "Combined_Key", "date")
```

```r
head(df_tidy)
```

```
## # A tibble: 6 x 9
##   Province_State Country_Region   Lat Long_ Combined_Key date  confirmed
##   <chr>          <chr>          <dbl> <dbl> <chr>        <chr>     <dbl>
```

```
## 1 Alabama        US              32.5 -86.6 Autauga, Al~ 1/22~         0
## 2 Alabama        US              32.5 -86.6 Autauga, Al~ 1/23~         0
## 3 Alabama        US              32.5 -86.6 Autauga, Al~ 1/24~         0
## 4 Alabama        US              32.5 -86.6 Autauga, Al~ 1/25~         0
## 5 Alabama        US              32.5 -86.6 Autauga, Al~ 1/26~         0
## 6 Alabama        US              32.5 -86.6 Autauga, Al~ 1/27~         0
## # ... with 2 more variables: Population <dbl>, deaths <dbl>
```

Now that we have tidied our data, let's deal with missing data.

```
summary(df_tidy) # there are several rows of missing data, but it appears to
```

```
##  Province_State     Country_Region          Lat              Long_
##  Length:2086084     Length:2086084     Min.   :-14.27   Min.   :-174.16
##  Class :character   Class :character   1st Qu.: 33.99   1st Qu.: -99.48
##  Mode  :character   Mode  :character   Median : 38.12   Median : -90.66
##                                        Mean   : 36.93   Mean   : -89.99
##                                        3rd Qu.: 41.73   3rd Qu.: -82.80
##                                        Max.   : 69.31   Max.   : 145.67
##
##  Combined_Key           date             confirmed         Population
##  Length:2086084     Length:2086084     Min.   :      0   Min.   :       0
##  Class :character   Class :character   1st Qu.:     23   1st Qu.:    9917
##  Mode  :character   Mode  :character   Median :    519   Median :   24892
##                                        Mean   :   4559   Mean   :   99604
##                                        3rd Qu.:   2416   3rd Qu.:   64979
##                                        Max.   :1350370   Max.   :10039107
##                                        NA's   :174460    NA's   :174460
##      deaths
##  Min.   :    0.00
##  1st Qu.:    0.00
##  Median :    9.00
##  Mean   :   89.38
##  3rd Qu.:   47.00
##  Max.   :24911.00
##  NA's   :174460
```

```
# be consistent acroos confirmed, population and deaths. Perhaps from reporting
# issues? i.e. weekends? We will simply drop the data.
df_tidy <- df_tidy %>% drop_na()
```

Finally, we will clean up some of the variables and do some feature engineering.

```
head(df_tidy)
```

```
## # A tibble: 6 x 9
##   Province_State Country_Region   Lat Long_ Combined_Key date  confirmed
##   <chr>          <chr>          <dbl> <dbl> <chr>        <chr>     <dbl>
## 1 Alabama        US              32.5 -86.6 Autauga, Al~ 1/22~         0
## 2 Alabama        US              32.5 -86.6 Autauga, Al~ 1/23~         0
## 3 Alabama        US              32.5 -86.6 Autauga, Al~ 1/24~         0
## 4 Alabama        US              32.5 -86.6 Autauga, Al~ 1/25~         0
```

```
## 5 Alabama        US              32.5 -86.6 Autauga, Al~ 1/26~         0
## 6 Alabama        US              32.5 -86.6 Autauga, Al~ 1/27~         0
## # ... with 2 more variables: Population <dbl>, deaths <dbl>
```

```
df <-df_tidy %>%
  mutate(date = floor_date(mdy(date), unit = "day")) %>%
  # group and summarize by state
  group_by(Province_State, date) %>%
  summarize(Population = sum(Population),
            confirmed = sum(confirmed),
            deaths = sum(deaths)) %>%
  # add some features
  mutate(week_start = floor_date(date, unit = "week"), # not sure if we will use this
         pct_deaths = deaths / Population, # note that deaths is cumulative
         pct_confirmed = confirmed / Population, # note confirmed is cumulative
         #  we try to extract an approximation of active cases using the 14 day
         # guideline provided by the CDC as an "active period"
         active = (confirmed - lag(confirmed, 14)) - (deaths - lag(deaths, 14)),
         pct_active = active / Population, # active is NOT cumulative (for
         # reasons that I hope are obvious!)
         pct_change_deaths = deaths / lag(deaths) - 1,
         pct_change_confirmed = confirmed / lag(confirmed) - 1,
         pct_change_active = active / lag(active) - 1)
```

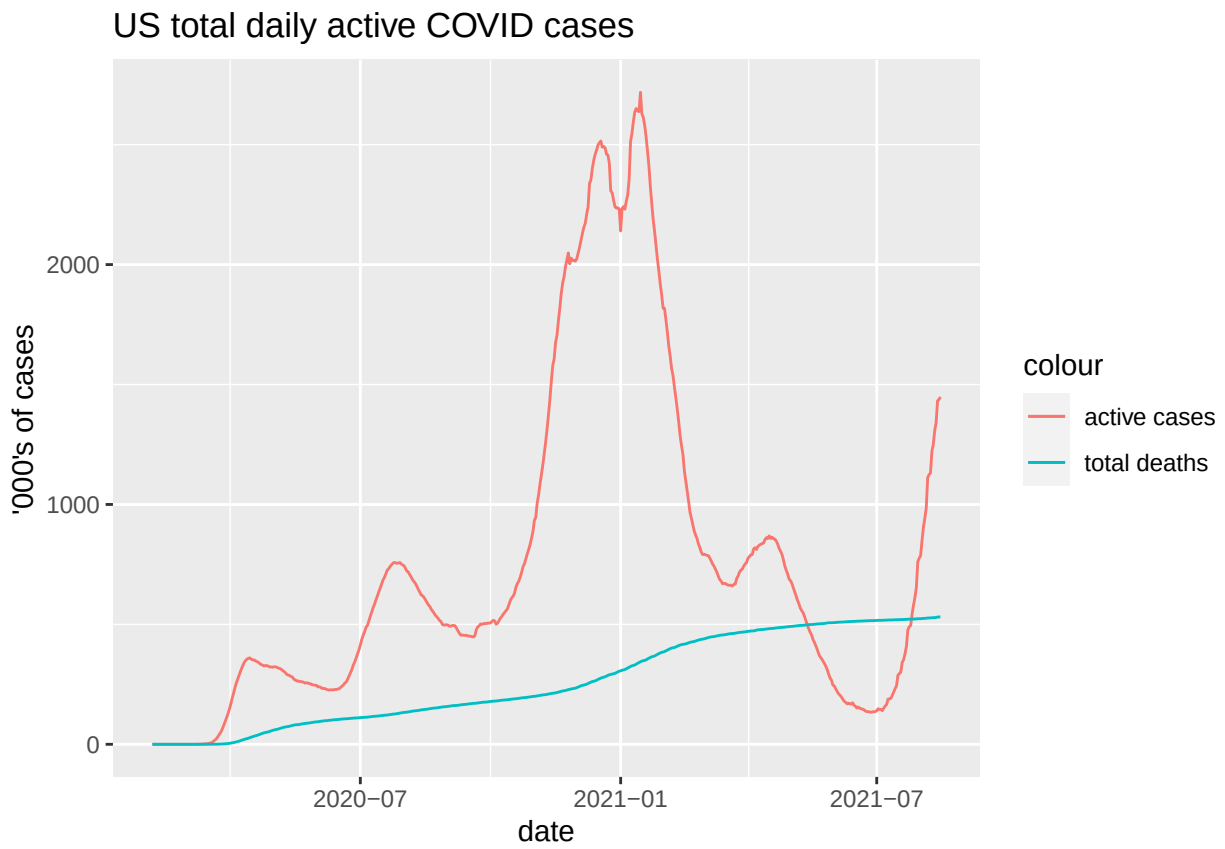
```
## `summarise()` regrouping output by 'Province_State' (override with `.groups` argument)
```

Exploratory Analysis

First, let's look at active cases against time. In the chart below, we can see a massive spike in active cases towards the end of 2020, and another large spike that is current in the summer of 2021.

```
df %>%
  group_by(date) %>%
  # let's generalize accross the country
  summarize(deaths = sum(deaths),
            confirmed = sum(confirmed),
            active = sum(active)) %>%
  drop_na() %>%
  ggplot(aes(x = date, y = active / 1000, color = "active cases")) +
  geom_line() +
  geom_line(aes(y = deaths/1000, color = "total deaths")) +
  labs(title = "US total daily active COVID cases",
       x = "date",
       y = "'000's of cases")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```




```
# for the technical anaslysts of the stock market among the class
# this appears to be a classic "head and shoulders" signal...
# sorry to make light of a morbid matter.
```

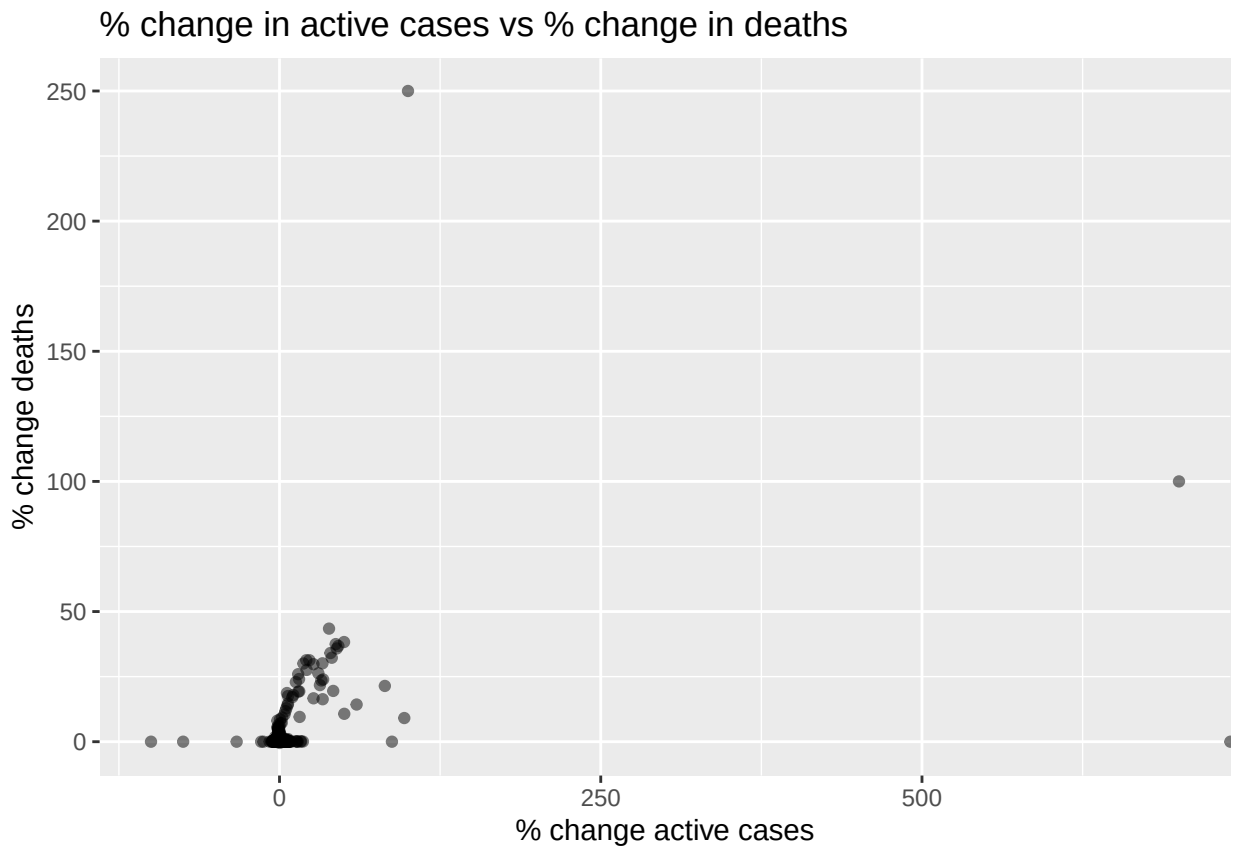
Is the number of active cases correlated to the number of deaths? We will plot the daily percent changes against each other. Looking at the chart below, we can see that something is not qutie right.

```
# let's make a new data frame
df_changes <- df %>%
  group_by(date) %>%
  summarize(deaths = sum(deaths),
            confirmed = sum(confirmed),
            active = sum(active)) %>%
  mutate(pct_change_deaths = deaths/lag(deaths) - 1,
         pct_change_active = active / lag(active) - 1) %>%
  drop_na()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
df_changes %>%
  ggplot(aes(x = pct_change_active*100, y = pct_change_deaths*100)) +
  geom_point( alpha = 0.5) +
  # geom_line(aes(y = total_deaths/1000, color = "total deaths")) +
  labs(title = "% change in active cases vs % change in deaths",
```

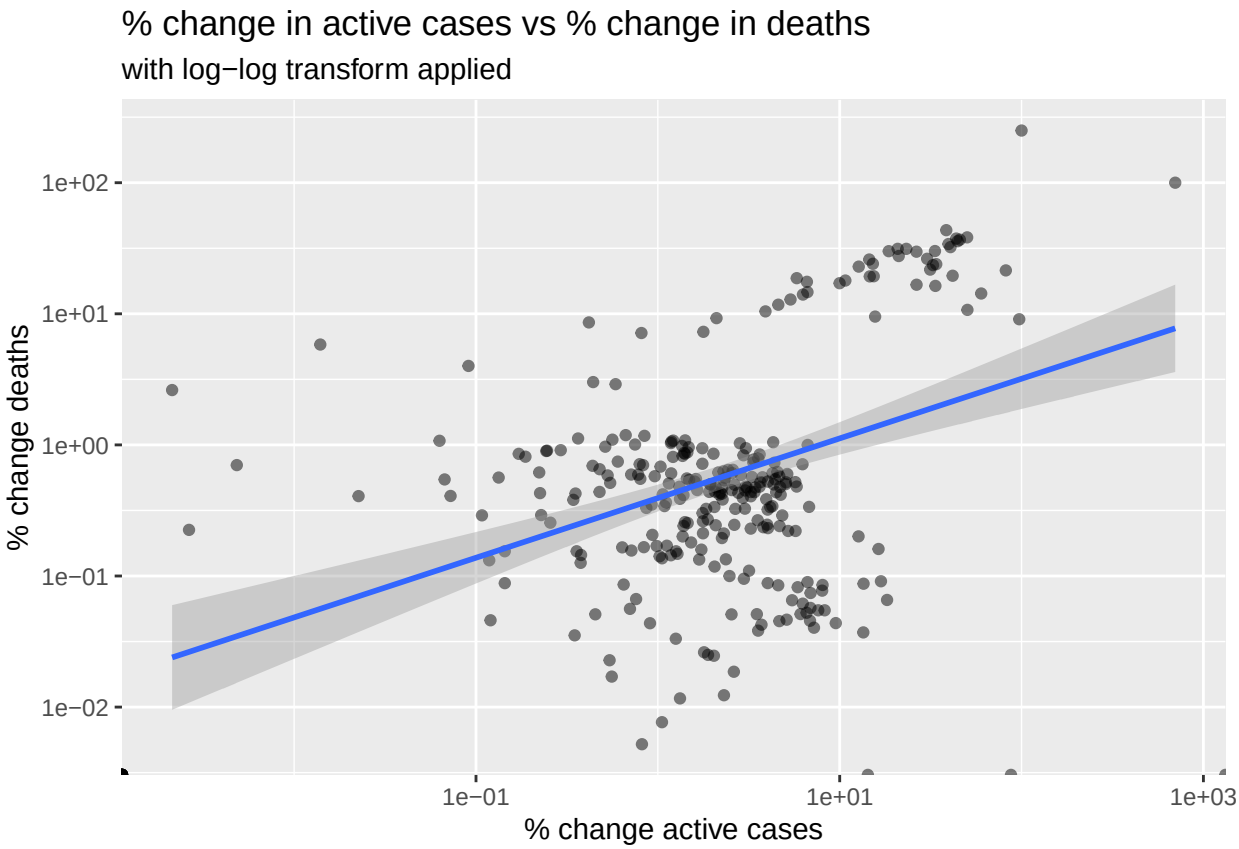
```
       x = "% change active cases",
       y = "% change deaths")
```



So we apply a log-log transform, and that helps normalize the data. In the chart below, we can see a linear relationship between the two variables once they have been transformed.

```
df_changes %>%
  ggplot(aes(x = pct_change_active*100, y = pct_change_deaths*100)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm") +
  # let's log transform to try and normalize
  scale_y_log10() +
  scale_x_log10() +
  labs(title = "% change in active cases vs % change in deaths",
       subtitle = "with log-log transform applied",
       x = "% change active cases",
       y = "% change deaths")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Modelling

We can model this effect via linear regression. We can see from the model below that percent change in active cases appears to have an effect and a positive correlation to percent change in deaths. That means that as more active cases occur, we can expect more deaths to occur which should not be suprising!

```
df_changes_log <- df_changes %>%
  mutate(pct_change_deaths = log(pct_change_deaths*100),
         pct_change_active = log(pct_change_active*100))
# there is probably a better way to do this next bit, but I pulled
# this code off the internet as-is: https://newbedev.com/how-to-remove-rows-with-inf-from-a-dataframe-i
# removes NaN and Inf row
df_changes_log <- df_changes_log[Reduce(`&`,
                                        lapply(df_changes_log,
                                               function(x) !is.na(x)  & is.finite(x))),]

fit_pct_change <- lm(pct_change_deaths ~ pct_change_active,
                     data = df_changes_log)

# here is the model
summary(fit_pct_change)
```

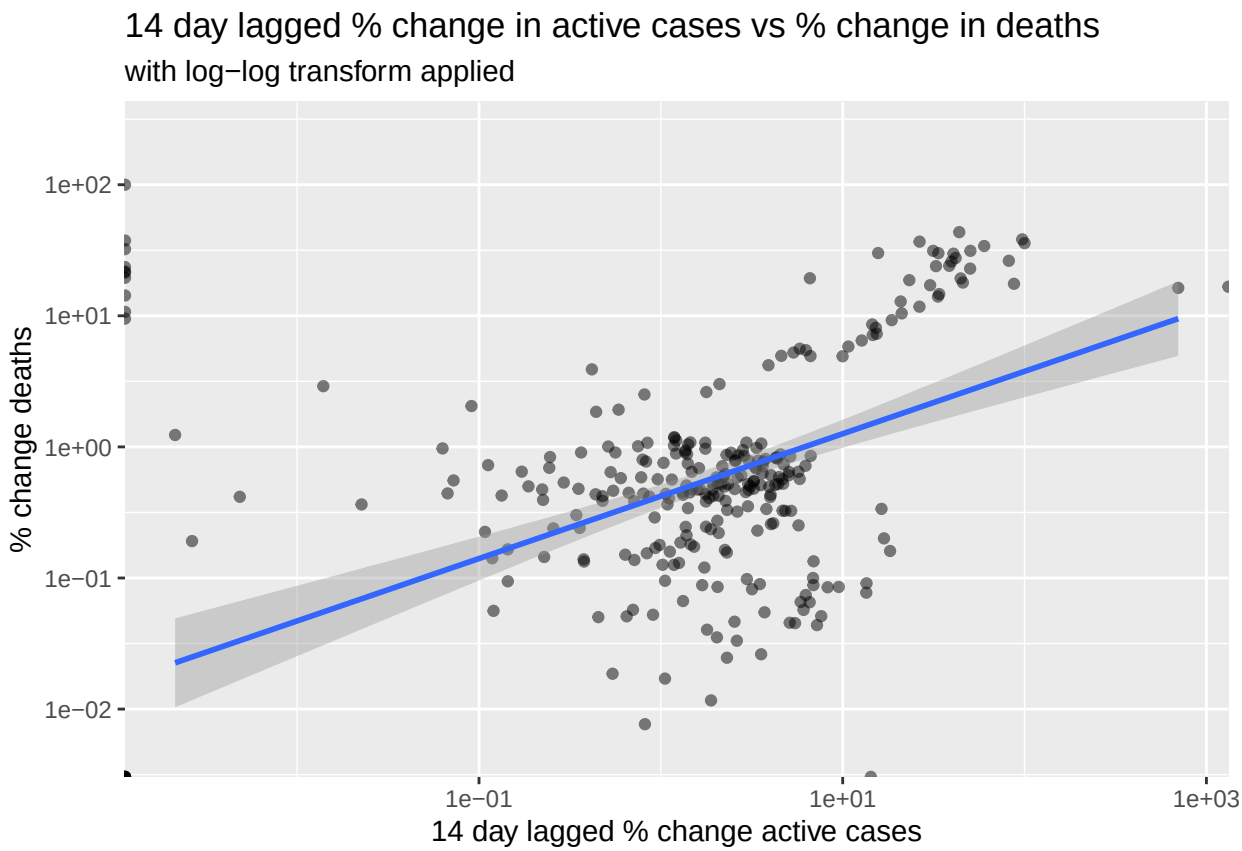
```
##
## Call:
```

```
## lm(formula = pct_change_deaths ~ pct_change_active, data = df_changes_log)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2317 -1.0026 -0.1432  0.9424  4.6967
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.93414    0.11979  -7.798 1.31e-13 ***
## pct_change_active  0.45518    0.06525   6.976 2.26e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.789 on 275 degrees of freedom
## Multiple R-squared:  0.1504, Adjusted R-squared:  0.1473
## F-statistic: 48.67 on 1 and 275 DF,  p-value: 2.263e-11
```

To take this one step further, let's look try adding a 14-day lag to the active cases % change. This way we can check to see whether an increase in active cases suggest more upcoming deaths. We make the same plot and run the same model as before, but with the 14-day lag.

```
df_changes %>%
  mutate(pct_change_active = lag(pct_change_active, 14)) %>%
  ggplot(aes(x = pct_change_active*100, y = pct_change_deaths*100)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm") +
  # let's log transform to try and normalize
  scale_y_log10() +
  scale_x_log10() +
  labs(title = "14 day lagged % change in active cases vs % change in deaths",
       subtitle = "with log-log transform applied",
       x = "14 day lagged % change active cases",
       y = "% change deaths")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
df_changes_log <- df_changes %>%
  mutate(pct_change_deaths = log(pct_change_deaths*100),
         pct_change_active = log(pct_change_active*100),
         pct_change_active = lag(pct_change_active, 14))
# there is probably a better way to do this next bit, but I pulled
# this code off the internet as-is: https://newbedev.com/how-to-remove-rows-with-inf-from-a-dataframe-i
# removes NaN and Inf row
df_changes_log <- df_changes_log[Reduce(`&`,
                                        lapply(df_changes_log,
                                               function(x) !is.na(x)  & is.finite(x))),]

fit_pct_change_14d_lag <- lm(pct_change_deaths ~ pct_change_active,
                     data = df_changes_log)

# here is the model
summary(fit_pct_change_14d_lag)
```

```
##
## Call:
## lm(formula = pct_change_deaths ~ pct_change_active, data = df_changes_log)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9096 -0.8136 -0.0565  0.9697  4.0018
##
```

```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.86513    0.10366  -8.346 4.05e-15 ***
## pct_change_active  0.47598    0.05545   8.584 8.11e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.525 on 263 degrees of freedom
## Multiple R-squared:  0.2189, Adjusted R-squared:  0.2159
## F-statistic: 73.69 on 1 and 263 DF,  p-value: 8.114e-16
```

As a comparison, we normalize the coefficient and we find that the results are not too different.

```
cat("0-day lag model\n")
```

```
## 0-day lag model
```

```
# normalizing the coefficients we get:
exp(fit_pct_change$coefficients)
```

```
##       (Intercept) pct_change_active
##         0.3929255         1.5764600
```

```
cat("\n\n14-day lag model\n")
```

```
##
##
## 14-day lag model
```

```
# normalizing the coefficients we get:
exp(fit_pct_change_14d_lag$coefficients)
```

```
##       (Intercept) pct_change_active
##         0.4209968         1.6095885
```

Bias

There are a number of ways that bias could be present in this analysis:

1. **Data Collection:** The data the John Hopkins provides is aggregated from several different sources. It is very likely that the quality, accuracy, and timeliness of the data varies depending on the source.
2. **Reporting Issues:** COVID in may places has been inconsistently reported. In some instances COVID is identified along with comorbidities, and in other places the opposite is reported, and further there are surely cases that are misreported or not reported at all.
3. **Data transformation issues:** When we performed the log transforms of the data, several NaN and Inf values arose. It is quite possible that there is a non-random pattern to those instances and that could skew the results of our analysis.

Conclusion, Further Analysis

In this analysis we looked at US COVID data over time and attempted to explore some of the relationships between active cases and subsequent deaths. Furhter analysis could go in many different directions and ask questions such as:

1. Do these relationships hold across different slices of time or regions?
2. Can we add in external data such as weather data that helps explain the various spikes in COVID?
3. Can we use the location and city level data to understand how COVID traverses the country and spreads?

There are many more questions that could be asked, examining these relationships has helped me gain a little bit of insight into the pandemic.

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.7.9.2 forcats_0.5.0     stringr_1.4.0     dplyr_1.0.2
##  [5] purrr_0.3.4       readr_1.4.0       tidyr_1.1.2       tibble_3.0.4
##  [9] ggplot2_3.3.2     tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.0  xfun_0.24         lattice_0.20-41   splines_4.0.3
##  [5] haven_2.3.1       colorspace_2.0-0  vctrs_0.3.5       generics_0.1.0
##  [9] htmltools_0.5.1.1 mgcv_1.8-33       yaml_2.2.1        utf8_1.1.4
## [13] rlang_0.4.9       pillar_1.4.7      glue_1.4.2        withr_2.3.0
## [17] DBI_1.1.0         dbplyr_2.0.0      modelr_0.1.8      readxl_1.3.1
## [21] lifecycle_0.2.0   munsell_0.5.0     gtable_0.3.0      cellranger_1.1.0
## [25] rvest_0.3.6       evaluate_0.14     labeling_0.4.2    knitr_1.30
## [29] curl_4.3          fansi_0.4.1       broom_0.7.2       Rcpp_1.0.5
## [33] scales_1.1.1      backports_1.2.0   jsonlite_1.7.1    farver_2.0.3
## [37] fs_1.5.0          hms_0.5.3         digest_0.6.27     stringi_1.5.3
## [41] grid_4.0.3        cli_2.2.0         tools_4.0.3       magrittr_2.0.1
## [45] crayon_1.3.4      pkgconfig_2.0.3   Matrix_1.2-18     ellipsis_0.3.1
## [49] xml2_1.3.2        reprex_0.3.0      assertthat_0.2.1  rmarkdown_2.5
## [53] httr_1.4.2        rstudioapi_0.13   R6_2.5.0          nlme_3.1-149
## [57] compiler_4.0.3
```