Untitled

Gabriella Veytsel

October 8, 2021

processing script this script loads the raw data, processes and cleans it and saves it as Rds file in the processed\_data folder

Load packages

library(readxl) #for loading Excel files

## Warning: package 'readxl' was built under R version 4.1.1

library(here) #to set paths

## Warning: package 'here' was built under R version 4.1.1

## here() starts at C:/Users/Gabriella Veytsel/Desktop/MADA 2021/Veytsel-MADA-project

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.1

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.4 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.1 v forcats 0.5.1

## Warning: package 'tibble' was built under R version 4.1.1

## Warning: package 'tidyr' was built under R version 4.1.1

## Warning: package 'readr' was built under R version 4.1.1

## Warning: package 'purrr' was built under R version 4.1.1

## Warning: package 'stringr' was built under R version 4.1.1

## Warning: package 'forcats' was built under R version 4.1.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(sf)#st\_read to read shapefiles

## Warning: package 'sf' was built under R version 4.1.1

## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1

library(tmap) #plot map

## Warning: package 'tmap' was built under R version 4.1.1

library(scales)

## Warning: package 'scales' was built under R version 4.1.1

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

library(broom) #tidy

## Warning: package 'broom' was built under R version 4.1.1

library(ggpubr) #add equation and r-square

## Warning: package 'ggpubr' was built under R version 4.1.1

path to data note the use of the here() package and not absolute paths

data\_location <- here::here("data","raw\_data","CaseCountData.xlsx")  
  
#the first row of data is not column names, skip  
rawdata\_Cases.and.Fatalities <- read\_excel(data\_location, sheet = 1, skip = 1) #skip first 2 rows  
  
#the first two rows of data are not column names, skip  
rawdata\_Trends <- read\_excel(data\_location, sheet = 2, skip = 3)

Take a look at the data

#2 obs with missing probable cases, 2 obs with missing confirmed cases  
summary(rawdata\_Cases.and.Fatalities)

## County Confirmed Cases Probable Cases Fatalities   
## Length:256 Min. : 5 Min. : 0.0 Min. : 0.0   
## Class :character 1st Qu.: 617 1st Qu.: 138.2 1st Qu.: 23.0   
## Mode :character Median : 2063 Median : 522.5 Median : 61.5   
## Mean : 13365 Mean : 2652.8 Mean : 505.7   
## 3rd Qu.: 5234 3rd Qu.: 1662.2 3rd Qu.: 173.2   
## Max. :558282 Max. :58990.0 Max. :64732.0   
## NA's :2 NA's :2

#examine those 2 obs using filter(is.na):   
 #county = unknown & county = grand total  
#remove them  
cleandata\_Cases.and.Fatalities <- rawdata\_Cases.and.Fatalities %>%   
 filter(!is.na(`Confirmed Cases`))   
  
summary(rawdata\_Trends) #Date should not be a character

## Date Cumulative Confirmed Cases New Confirmed Cases  
## Length:579 Min. : 0 Min. : 0   
## Class :character 1st Qu.: 383790 1st Qu.: 1361   
## Mode :character Median :1384476 Median : 3621   
## Mean :1453512 Mean : 5560   
## 3rd Qu.:2491423 3rd Qu.: 8337   
## Max. :3394815 Max. :26990   
##   
## Cumulative Probable Cases New Probable Cases Cumulative Fatalities  
## Min. : 67475 Min. : 7.0 Min. : 0   
## 1st Qu.:280465 1st Qu.: 454.8 1st Qu.: 9328   
## Median :401672 Median :1003.0 Median :28307   
## Mean :372537 Mean :1628.1 Mean :29682   
## 3rd Qu.:447291 3rd Qu.:2570.0 3rd Qu.:50388   
## Max. :673804 Max. :9241.0 Max. :64732   
## NA's :242 NA's :243 NA's :4   
## Fatalities by Date of Death  
## Min. : 0.0   
## 1st Qu.: 32.0   
## Median : 73.0   
## Mean :111.8   
## 3rd Qu.:189.5   
## Max. :386.0   
##

head(rawdata\_Trends$Date) #Format is year/month/day

## [1] "2020/03/04" "2020/03/05" "2020/03/06" "2020/03/07" "2020/03/08"  
## [6] "2020/03/09"

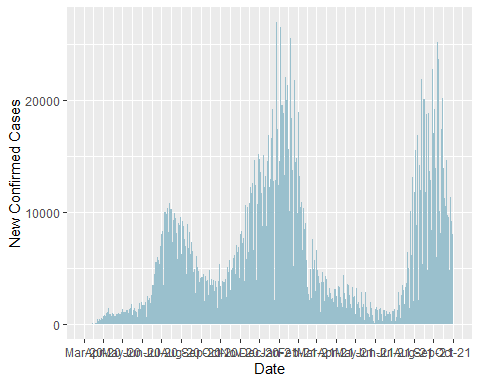
cleandata\_Trends <- rawdata\_Trends %>%  
 mutate(Date = as.Date(rawdata\_Trends$Date, '%Y/%m/%d'))  
head(cleandata\_Trends$Date) #looks good

## [1] "2020-03-04" "2020-03-05" "2020-03-06" "2020-03-07" "2020-03-08"  
## [6] "2020-03-09"

Epidemic curve

epicurve <- ggplot(cleandata\_Trends, aes(x=Date, y = `New Confirmed Cases`)) +  
 geom\_bar(stat = "identity", fill="lightblue3") +  
 scale\_x\_date(date\_breaks = "months" ,   
 date\_labels = "%b-%y",   
 limits = c(min(cleandata\_Trends$Date), max = max(cleandata\_Trends$Date)))  
epicurve

## Warning: Removed 2 rows containing missing values (geom\_bar).



#Save figure  
figure\_file\_epicurve = here("results","epicurve.png")  
ggsave(filename = figure\_file\_epicurve, plot=epicurve, height = 5, width = 8)

## Warning: Removed 2 rows containing missing values (geom\_bar).

Map

#Shapefiles came from: https://www.depts.ttu.edu/geospatial/center/TexasGISData.html  
shapefile\_location <- here::here("data","raw\_data","Tx\_CntyBndry\_Jurisdictional\_TIGER", "Tx\_CntyBndry\_Jurisdictional\_TIGER.shp")  
TX\_shapefile <- st\_read(shapefile\_location) #254 features (counties)

## Reading layer `Tx\_CntyBndry\_Jurisdictional\_TIGER' from data source   
## `C:\Users\Gabriella Veytsel\Desktop\MADA 2021\Veytsel-MADA-project\data\raw\_data\Tx\_CntyBndry\_Jurisdictional\_TIGER\Tx\_CntyBndry\_Jurisdictional\_TIGER.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 254 features and 16 fields  
## Geometry type: POLYGON  
## Dimension: XY  
## Bounding box: xmin: -106.6457 ymin: 25.83717 xmax: -93.50805 ymax: 36.50071  
## Geodetic CRS: WGS 84

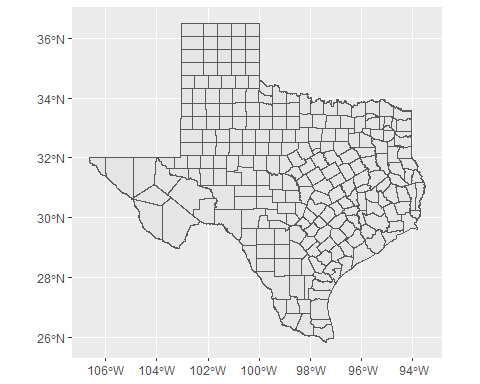
#Merge shapefile with metadata  
#Concatenate county name with "County" to match shapefile county names  
cleandata\_Cases.and.Fatalities$County <- paste(cleandata\_Cases.and.Fatalities$County, "County", sep = " ")  
merge\_shp <- merge(TX\_shapefile, cleandata\_Cases.and.Fatalities, by.x = "NAME", by.y = "County")   
#the order matters! if shapefile is 2nd, merge\_shp is dataframe, not sf

There's a population variable (2010 census) in the shapefile! Calcualate population density

merge\_shp <- merge\_shp %>%  
 mutate(Population\_Density = POP\_2010/AREA\_SQKM) %>% #population/area (km^2)  
 mutate(Population\_Density = round(Population\_Density,2)) %>%  
 arrange(desc(`Confirmed Cases`)) #sort by confirmed cases

Examine shapefile

ggplot() + geom\_sf(data = TX\_shapefile)



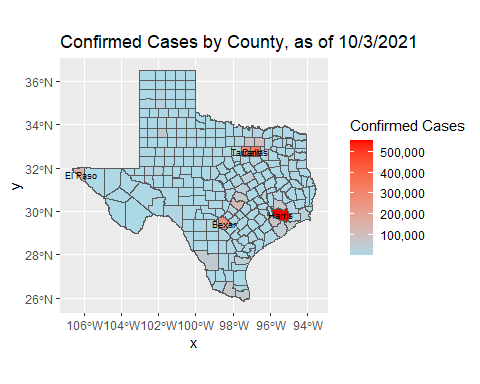
Top 5 counties, most confirmed cases

tail(sort(merge\_shp$`Confirmed Cases`),5)

## [1] 145433 257091 287546 331214 558282

map\_cases <- ggplot(merge\_shp) +  
 geom\_sf(aes(fill = `Confirmed Cases`)) +  
 ggtitle("Confirmed Cases by County, as of 10/3/2021") +  
 geom\_sf\_text(data = subset(merge\_shp, `Confirmed Cases` >= 145433), aes(label = COUNTY), size = 2.5) +  
 scale\_fill\_gradient(low = "light blue", high = "red", name = "Confirmed Cases", labels = comma)   
#labels = comma removes scientific notation (package 'scales')  
#label only top 5 counties (in terms of confirmed cases)  
#Harris County includes Houston  
#Later might want to use arrows  
  
map\_cases

## Warning in st\_point\_on\_surface.sfc(sf::st\_zm(x)): st\_point\_on\_surface may not  
## give correct results for longitude/latitude data



#Save figure  
figure\_file\_map\_cases = here("results","map\_cases.png")  
ggsave(filename = figure\_file\_map\_cases, plot=map\_cases, height = 12, width = 13)

## Warning in st\_point\_on\_surface.sfc(sf::st\_zm(x)): st\_point\_on\_surface may not  
## give correct results for longitude/latitude data

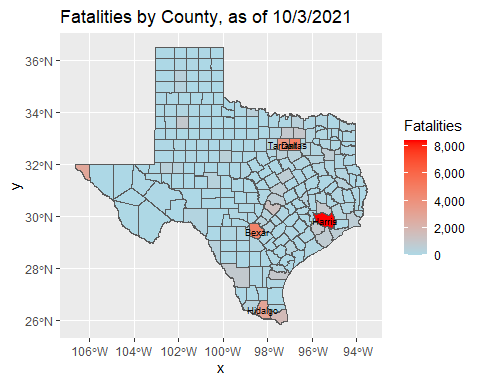
Top 5 counties, most fatalities

tail(sort(merge\_shp$Fatalities),5)

## [1] 2998 3825 4769 4931 8414

map\_deaths <- ggplot(merge\_shp) +  
 geom\_sf(aes(fill = `Fatalities`)) +  
 ggtitle("Fatalities by County, as of 10/3/2021") +  
 geom\_sf\_text(data = subset(merge\_shp, Fatalities >= 2998), aes(label = COUNTY), size = 2.5) +  
 scale\_fill\_gradient(low = "light blue", high = "red", name = "Fatalities", labels = comma)   
map\_deaths

## Warning in st\_point\_on\_surface.sfc(sf::st\_zm(x)): st\_point\_on\_surface may not  
## give correct results for longitude/latitude data



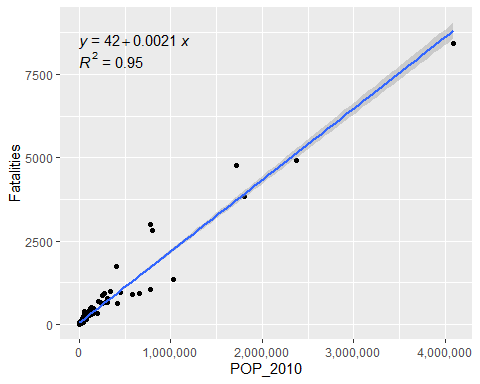
#Save figure  
figure\_file\_map\_deaths = here("results","map\_deaths.png")  
ggsave(filename = figure\_file\_map\_deaths, plot=map\_deaths, height = 12, width = 13)

## Warning in st\_point\_on\_surface.sfc(sf::st\_zm(x)): st\_point\_on\_surface may not  
## give correct results for longitude/latitude data

Association between pop density, pop and # confirmed cases, Fatalities?

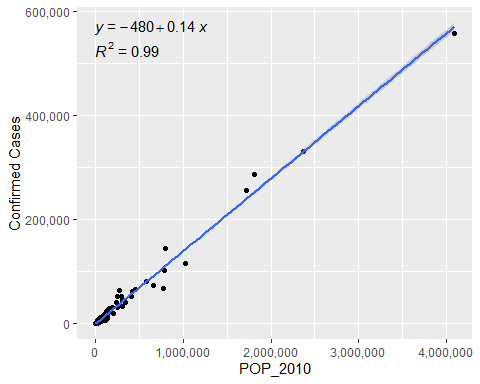
ggplot(merge\_shp, aes(x=POP\_2010, y=Fatalities)) +  
 geom\_point() +  
 geom\_smooth(method='lm') + #add linear regression line   
 stat\_regline\_equation(label.y = 8500, aes(label = ..eq.label..)) +  
 stat\_regline\_equation(aes(label = ..rr.label..)) +  
 scale\_x\_continuous(labels = comma) #remove scientific notation

## `geom\_smooth()` using formula 'y ~ x'



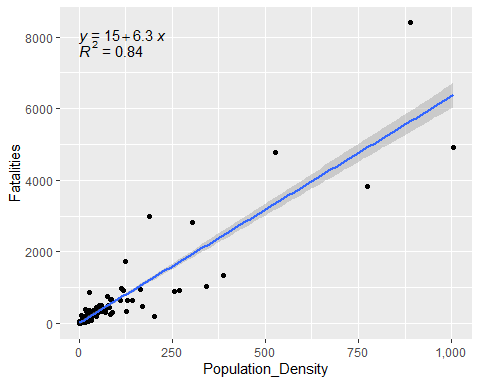
ggplot(merge\_shp, aes(x=POP\_2010, y=`Confirmed Cases`)) +  
 geom\_point() +  
 geom\_smooth(method='lm') + #add linear regression line   
 stat\_regline\_equation(label.y = 570000, aes(label = ..eq.label..)) +  
 stat\_regline\_equation(aes(label = ..rr.label..)) +  
 scale\_x\_continuous(labels = comma) + #remove scientific notation  
 scale\_y\_continuous(labels = comma)

## `geom\_smooth()` using formula 'y ~ x'



ggplot(merge\_shp, aes(x=Population\_Density, y=Fatalities)) +  
 geom\_point() +  
 geom\_smooth(method='lm') + #add linear regression line   
 stat\_regline\_equation(aes(label = ..eq.label..)) +  
 stat\_regline\_equation(label.y = 7700, aes(label = ..rr.label..)) +  
 scale\_x\_continuous(labels = comma) #remove scientific notation

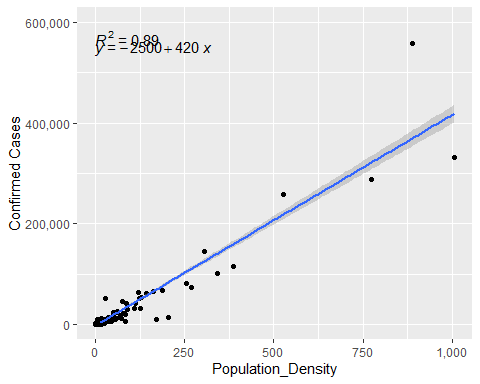
## `geom\_smooth()` using formula 'y ~ x'



ggplot(merge\_shp, aes(x=Population\_Density, y=`Confirmed Cases`)) +  
 geom\_point() +  
 geom\_smooth(method='lm') + #add linear regression line   
 stat\_regline\_equation(label.y = 550000, aes(label = ..eq.label..)) +  
 stat\_regline\_equation(aes(label = ..rr.label..)) +  
 scale\_x\_continuous(labels = comma) + #remove scientific notation  
 scale\_y\_continuous(labels = comma, limits = c(0, 600000))

## `geom\_smooth()` using formula 'y ~ x'

## Warning: Removed 1 rows containing missing values (geom\_smooth).



Linear regression

lm\_deaths\_pop <- lm(Fatalities~POP\_2010, data = merge\_shp)  
fit\_deaths\_pop <- tidy(lm\_deaths\_pop)  
print(fit\_deaths\_pop)

## # A tibble: 2 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 42.5 11.2 3.79 1.90e- 4  
## 2 POP\_2010 0.00214 0.0000307 70.0 4.36e-167

lm\_deaths\_density <- lm(Fatalities~Population\_Density, data = merge\_shp)  
fit\_deaths\_density <- tidy(lm\_deaths\_density)  
print(fit\_deaths\_density)

## # A tibble: 2 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 15.0 20.8 0.719 4.73e- 1  
## 2 Population\_Density 6.33 0.176 36.0 3.14e-101

lm\_cases\_pop <- lm(`Confirmed Cases`~POP\_2010, data = merge\_shp)  
fit\_cases\_pop <- tidy(lm\_cases\_pop)  
print(fit\_cases\_pop)

## # A tibble: 2 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -478. 368. -1.30 1.95e- 1  
## 2 POP\_2010 0.140 0.00100 139. 2.53e-240

lm\_cases\_density <- lm(`Confirmed Cases`~Population\_Density, data = merge\_shp)  
fit\_cases\_density <- tidy(lm\_cases\_density)  
print(fit\_cases\_density)

## # A tibble: 2 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -2494. 1075. -2.32 2.12e- 2  
## 2 Population\_Density 419. 9.10 46.0 1.35e-124

#Save results tables  
table\_file\_deaths\_pop = here("results", "resulttable\_deaths\_pop.rds")  
saveRDS(fit\_deaths\_pop, file = table\_file\_deaths\_pop)  
  
table\_file\_deaths\_density = here("results", "resulttable\_deaths\_density.rds")  
saveRDS(fit\_deaths\_density, file = table\_file\_deaths\_density)  
  
table\_file\_cases\_pop = here("results", "resulttable\_cases\_pop.rds")  
saveRDS(fit\_cases\_pop, file = table\_file\_cases\_pop)  
  
table\_file\_cases\_density = here("results", "resulttable\_cases\_density.rds")  
saveRDS(fit\_cases\_density, file = table\_file\_cases\_density)

save data as RDS I suggest you save your processed and cleaned data as RDS or RDA/Rdata files. This preserves coding like factors, characters, numeric, etc. If you save as CSV, that information would get lost. See here for some suggestions on how to store your processed data: <http://www.sthda.com/english/wiki/saving-data-into-r-data-format-rds-and-rdata>

location to save file

#save\_data\_location\_spatial <- here::here("data","merge\_shp","processeddata\_spatial.rds")  
#save\_data\_location\_trends <- here::here("data","cleandata\_Trends","processeddata\_trends.rds")  
  
#saveRDS(merge\_shp, file = save\_data\_location\_spatial)  
#saveRDS(cleandata\_Trends, file = save\_data\_location\_trends)