Project 2

Cameron Farrugia

4/21/2020

require("COVID19")

## Loading required package: COVID19

require(dplyr)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

require(ggplot2)

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.6.3

require(tidyverse)

## Loading required package: tidyverse

## Warning: package 'tidyverse' was built under R version 3.6.3

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

## v tibble 2.1.3 v purrr 0.3.2  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## Warning: package 'tidyr' was built under R version 3.6.3

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

Making state a Factor because I want to group by that variable

covid=covid19("ITA", level = 2)  
# write.csv(covid, file="C:/Users/camer/Desktop/DAT512 Stat Big Data/covidOG.csv")  
str(covid)

## Classes 'grouped\_df', 'tbl\_df', 'tbl' and 'data.frame': 1218 obs. of 24 variables:  
## $ id : chr "ITA, Abruzzo" "ITA, Abruzzo" "ITA, Abruzzo" "ITA, Abruzzo" ...  
## $ date : Date, format: "2020-02-24" "2020-02-25" ...  
## $ deaths : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ confirmed : num 0 0 0 1 1 2 5 5 6 7 ...  
## $ tests : num 5 5 13 33 33 43 52 52 52 85 ...  
## $ recovered : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ hosp : num 0 0 0 1 1 2 3 3 5 7 ...  
## $ icu : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ vent : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ driving : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ walking : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ transit : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ country : chr "Italy" "Italy" "Italy" "Italy" ...  
## $ state : chr "Abruzzo" "Abruzzo" "Abruzzo" "Abruzzo" ...  
## $ city : chr NA NA NA NA ...  
## $ lat : num 42.4 42.4 42.4 42.4 42.4 ...  
## $ lng : num 13.4 13.4 13.4 13.4 13.4 ...  
## $ pop : int 1315196 1315196 1315196 1315196 1315196 1315196 1315196 1315196 1315196 1315196 ...  
## $ pop\_14 : num 0.126 0.126 0.126 0.126 0.126 0.126 0.126 0.126 0.126 0.126 ...  
## $ pop\_15\_64 : num 0.639 0.639 0.639 0.639 0.639 0.639 0.639 0.639 0.639 0.639 ...  
## $ pop\_65 : num 0.236 0.236 0.236 0.236 0.236 0.236 0.236 0.236 0.236 0.236 ...  
## $ pop\_age : num 45.9 45.9 45.9 45.9 45.9 45.9 45.9 45.9 45.9 45.9 ...  
## $ pop\_density : num 121 121 121 121 121 ...  
## $ pop\_death\_rate: num 0.0112 0.0112 0.0112 0.0112 0.0112 0.0112 0.0112 0.0112 0.0112 0.0112 ...  
## - attr(\*, "groups")=Classes 'tbl\_df', 'tbl' and 'data.frame': 21 obs. of 2 variables:  
## ..$ id : chr "ITA, Abruzzo" "ITA, Basilicata" "ITA, Calabria" "ITA, Campania" ...  
## ..$ .rows:List of 21  
## .. ..$ : int 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ : int 59 60 61 62 63 64 65 66 67 68 ...  
## .. ..$ : int 117 118 119 120 121 122 123 124 125 126 ...  
## .. ..$ : int 175 176 177 178 179 180 181 182 183 184 ...  
## .. ..$ : int 233 234 235 236 237 238 239 240 241 242 ...  
## .. ..$ : int 291 292 293 294 295 296 297 298 299 300 ...  
## .. ..$ : int 349 350 351 352 353 354 355 356 357 358 ...  
## .. ..$ : int 407 408 409 410 411 412 413 414 415 416 ...  
## .. ..$ : int 465 466 467 468 469 470 471 472 473 474 ...  
## .. ..$ : int 523 524 525 526 527 528 529 530 531 532 ...  
## .. ..$ : int 581 582 583 584 585 586 587 588 589 590 ...  
## .. ..$ : int 639 640 641 642 643 644 645 646 647 648 ...  
## .. ..$ : int 697 698 699 700 701 702 703 704 705 706 ...  
## .. ..$ : int 755 756 757 758 759 760 761 762 763 764 ...  
## .. ..$ : int 813 814 815 816 817 818 819 820 821 822 ...  
## .. ..$ : int 871 872 873 874 875 876 877 878 879 880 ...  
## .. ..$ : int 929 930 931 932 933 934 935 936 937 938 ...  
## .. ..$ : int 987 988 989 990 991 992 993 994 995 996 ...  
## .. ..$ : int 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 ...  
## .. ..$ : int 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 ...  
## .. ..$ : int 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 ...  
## ..- attr(\*, ".drop")= logi TRUE

covid$state=factor(covid$state)

Since the data is cumulative I want to eliminate all the entries that arn’t the most recent data.

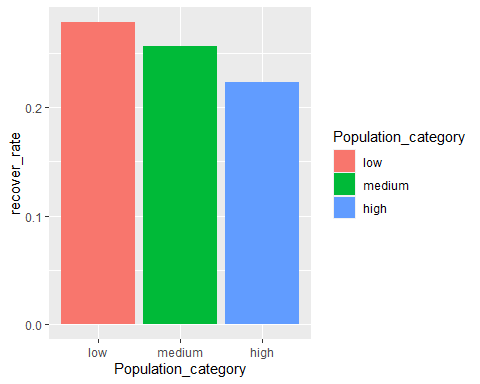
# Filter to show the most recent data since it is cumlative.   
covid2=covid  
covid2=covid2 %>%   
 filter(date == "2020-04-20")

Adding in some useful columns and making one a categorical variable

covid2=covid2 %>% group\_by(state) %>% mutate(recover\_rate=recovered/confirmed, Population\_category=pop\_density)  
covid2$Population\_category= cut(covid2$Population\_category, breaks=c(0,150,300,1000), labels=c("low", "medium", "high"))

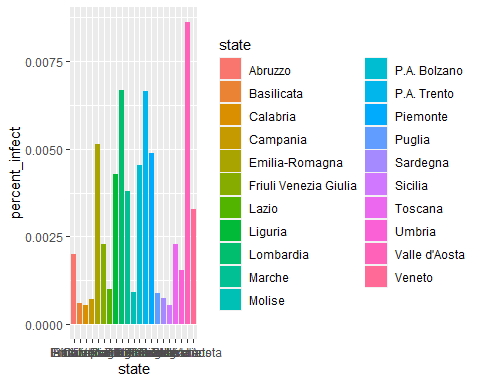
Shwoing recovery rate based on population area. More factors go into why it shows low density having the higher recovery rate, but I thought it was an interesting point to be made.

covid3=covid2 %>% group\_by(Population\_category) %>% mutate(recover\_rate=mean(recover\_rate)) %>% distinct(recover\_rate)  
View(covid3)  
ggplot(covid3, aes(x=Population\_category, y= recover\_rate, fill=Population\_category))+geom\_col()



This shows the percent of overall population in an area that have COVID-19

covid4= covid2 %>% mutate(percent\_infect= confirmed/pop)  
View(covid4)  
ggplot(covid4, aes(x=state, y=percent\_infect, fill=state, key=FALSE))+geom\_col()



First attempted linear model. y= confirmed x1=population, x2=population density, x3=Population Category, x4=tests I included these parameters because they seemed the most likely to affect the number of confirmed cases.

model<- lm(confirmed ~ pop+pop\_density+Population\_category+tests, data=covid4)  
summary(model)

##   
## Call:  
## lm(formula = confirmed ~ pop + pop\_density + Population\_category +   
## tests, data = covid4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17403 -3564 78 3618 18393   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.908e+03 5.163e+03 -0.370 0.7168   
## pop 2.791e-03 2.139e-03 1.305 0.2116   
## pop\_density -1.436e+01 5.187e+01 -0.277 0.7857   
## Population\_categorymedium -5.469e+03 7.716e+03 -0.709 0.4893   
## Population\_categoryhigh -3.840e+03 1.819e+04 -0.211 0.8356   
## tests 1.189e-01 4.746e-02 2.505 0.0243 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 8833 on 15 degrees of freedom  
## Multiple R-squared: 0.7352, Adjusted R-squared: 0.647   
## F-statistic: 8.331 on 5 and 15 DF, p-value: 0.0006128

This model isn’t very good because the F statistic is pretty low. This maybe because of the dataset size or the predictors just arn’t very good at predicting a confirmed case.