Preliminary Analysis

Team Ninja
Department of Applied Mathematics & Statistics
State University of New York at Stony Brook

2020/11/18

```
rm(list=ls())
if (!require(readxl)) install.packages('readxl')
```

```
## Loading required package: readxl
```

```
if (!require(ggplot2)) install.packages('ggplot2')
```

```
## Loading required package: ggplot2
```

```
if (!require(devtools)) install.packages('devtools')
```

```
## Loading required package: devtools
```

```
## Loading required package: usethis
```

```
if (!require(covidcast)) install.packages('covidcast')
```

```
## Loading required package: covidcast
```

```
## We encourage COVIDcast API users to register on our mailing list:
## https://lists.andrew.cmu.edu/mailman/listinfo/delphi-covidcast-api
## We'll send announcements about new data sources, package updates,
## server maintenance, and new features.
```

```
if (!require(dunn.test)) install.packages('dunn.test')
```

```
## Loading required package: dunn.test
```

```
if (!require(ggpubr)) install.packages('ggpubr')
```

```
## Loading required package: ggpubr
```

```
if (!require(openxlsx)) install.packages('openxlsx')
```

```
## Loading required package: openxlsx
```

```
if (!require(hqreg)) install.packages('hqreg')
```

```
## Loading required package: hqreg
```

```
library(readxl)
library(ggplot2)
library(devtools)
library(covidcast)
library(dunn.test)
library(ggpubr)
library(openxlsx)
library(hqreg)
```

Preprocessing

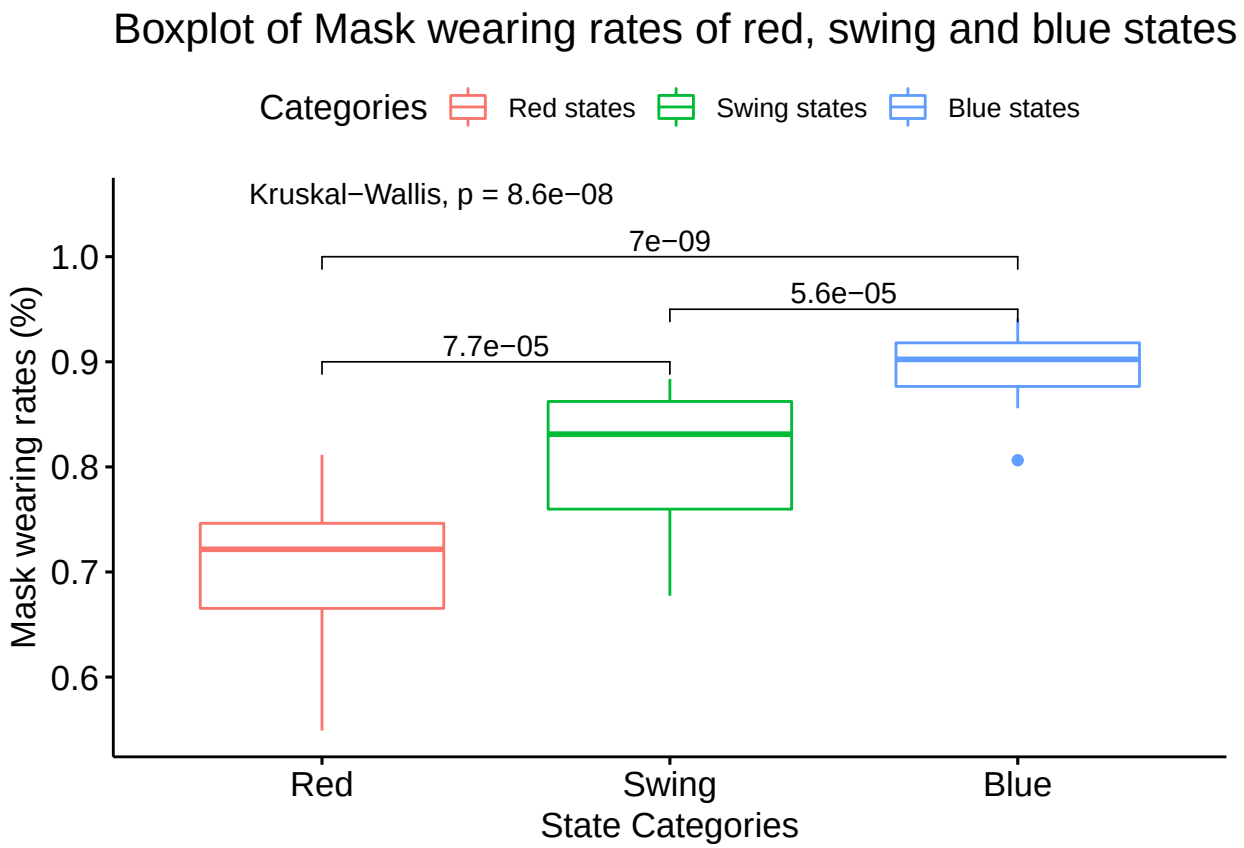
```
# Read data
mask = read.csv("mask-use-by-county.csv")
population = read_excel("co-est2019-annres.xlsx", skip = 4)[1:3142, c(1,13)]
rb = read_excel("rb_cont.xlsx")
tpolicy = read.csv("policy_date.csv")[1:51,]

colnames(population) = c("state", "population")
population$state = sub(".*,()","\\1",population$state)
population$state = trimws(population$state)
data = cbind(mask, population) # mask and population
data$total = ave(data$population, data$state, FUN = sum)
data$weight = data$population/data$total
data$mask = data$ALWAYS + data$FREQUENTLY # treat "always" and "frequently" as wearing mask
data$point = data$mask*data$weight
smask = tapply(data$point, data$state, FUN = sum) # population weighted
smask = as.data.frame(smask)[-9,]
data1 = cbind(smask, rb) # mask and rb
data1$population = unique(data$total)[-9]
data1$rep = data1$red - data1$blue
# Continuous variable of Red (vs. Blue)
data1$reps = "Swing"
data1$reps[(data1$red - data1$blue) > 0.1] = "Red"
data1$reps[(data1$blue - data1$red) > 0.1] = "Blue"
data1$reps = factor(data1$reps, levels = c("Red", "Swing", "Blue"), ordered = TRUE)
tpolicy = tpolicy[-9, 3:4]
for (i in 1:2){
  tpolicy[,i] = as.Date(tpolicy[,i], format = "%m/%d/%y")
}
data1 = cbind(data1, tpolicy)
```

Box-plots and Regressions

```
my_comparisons <- list( c("Red", "Blue"), c("Red", "Swing"), c("Blue", "Swing") )
ggboxplot(data1, x = "reps", y = "smask",
          color = "reps", palette = "jco")+
  stat_compare_means(comparisons = my_comparisons, label.y = c(1, 0.9, 0.95))+
  stat_compare_means(label.y = 1.05) +
  labs(x = "State Categories", y = "Mask wearing rates (%)", title = "Boxplot of Mask wearing rates of r
  scale_color_discrete(name = "Categories", labels = c("Red states","Swing states", "Blue states"))+
  theme(text = element_text(size = 13), axis.text.x = element_text(size = 13), axis.text.y = element_tex
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```




```
#ggsave("mask_rbs.png", width = 10, height = 7)
mean(data1$smask[data1$reps == "Red"])
```

```
## [1] 0.6980287
```

```
mean(data1$smask[data1$reps == "Blue"])
```

```
## [1] 0.8940097
```

```
shapiro.test(data1$smask)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data1$smask
## W = 0.9444, p-value = 0.02018
```

```
wilcox.test(smask ~ reps, data = data1[data1$reps != "Swing", c("smask","reps")])
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  smask by reps
## W = 1, p-value = 6.979e-09
## alternative hypothesis: true location shift is not equal to 0
```

```
kruskal.test(smask ~ reps, data = data1)
```

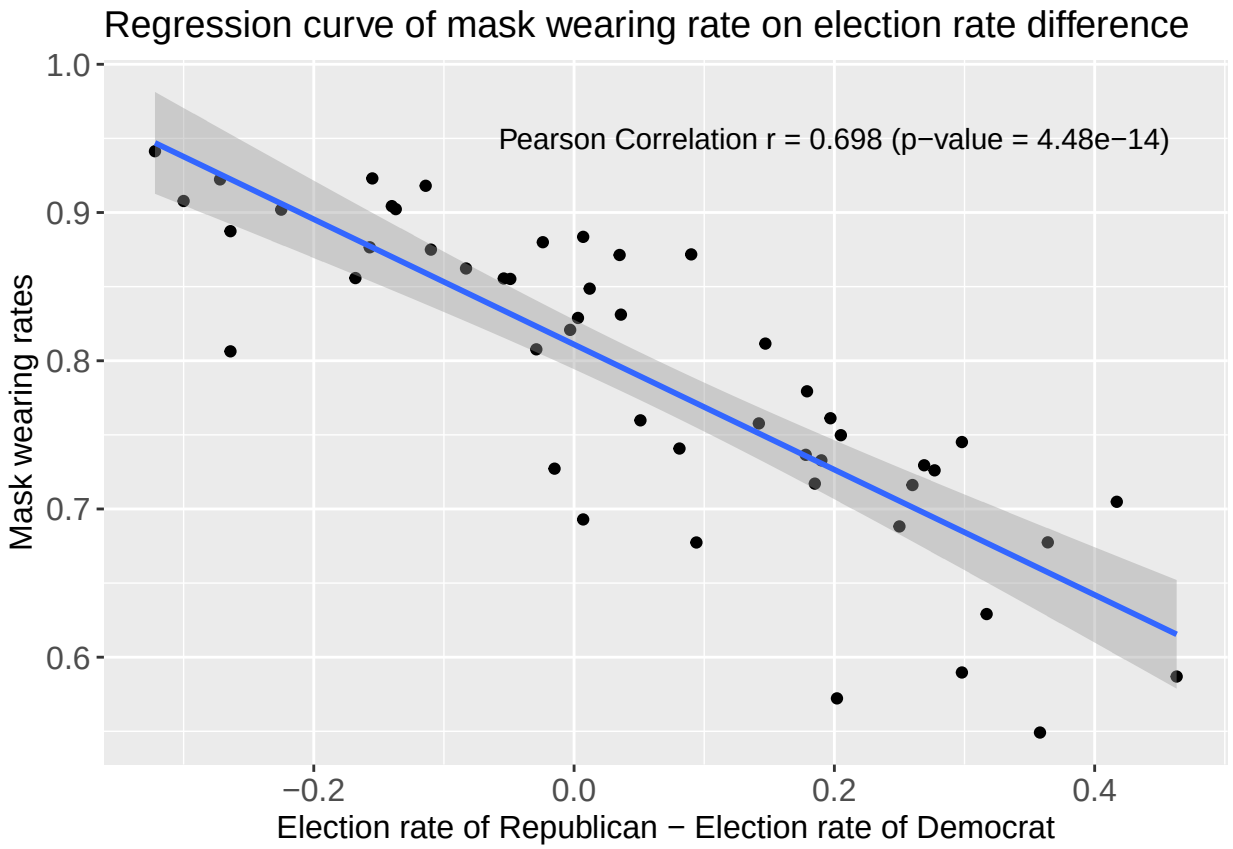
```
##
##  Kruskal-Wallis rank sum test
##
## data:  smask by reps
## Kruskal-Wallis chi-squared = 32.535, df = 2, p-value = 8.612e-08
```

```
dunn.test(data1$smask, data1$reps)
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 32.535, df = 2, p-value = 0
##
##
##                             Comparison of x by group
##                                   (No adjustment)
## Col Mean-|
## Row Mean |       Blue        Red
## ---------+----------------------
##      Red |   5.662538
##          |    0.0000*
##          |
##    Swing |   2.762518  -3.029955
##          |    0.0029*    0.0012*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

```
# Red vs. Blue (Continuous)
ggplot(data1, aes(x = rep, y = smask)) + geom_point() + geom_smooth(method=lm) +
  labs(x = "Election rate of Republican - Election rate of Democrat", y = "Mask wearing rates", title =
```

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



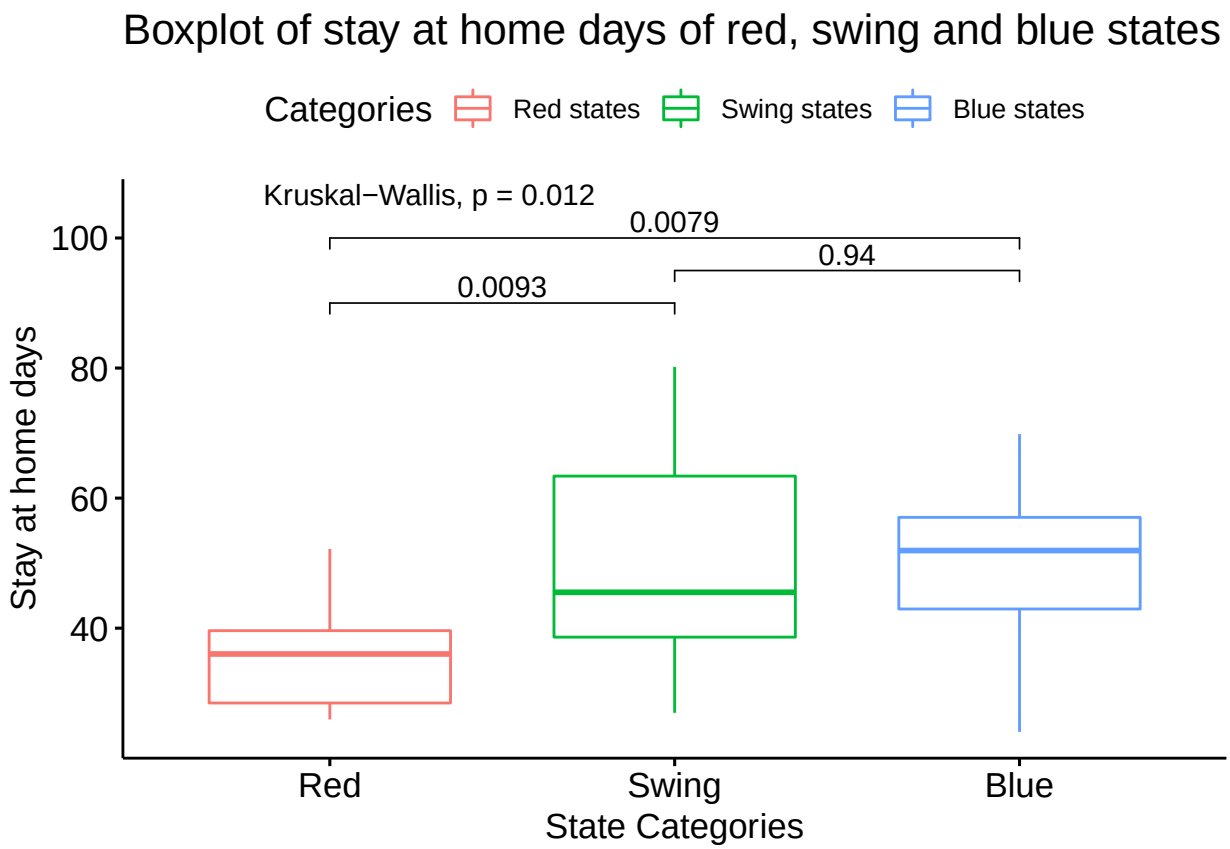

```
lm1 = lm(smask~rep, data1)
summary(lm1)
```

```
##
## Call:
## lm(formula = smask ~ rep, data = data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15346 -0.02790  0.01177  0.03401  0.09877
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.810988   0.008304   97.67  < 2e-16 ***
## rep         -0.422389   0.040095  -10.54 4.48e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05664 on 48 degrees of freedom
## Multiple R-squared:  0.6981, Adjusted R-squared:  0.6918
## F-statistic:   111 on 1 and 48 DF,  p-value: 4.478e-14
```

```
data1$stay_at_home_date = as.numeric(data1$stay_at_home_date) - as.numeric(as.Date("2020-01-01"))
data1$stay_at_home_expire_date = as.numeric(data1$stay_at_home_expire_date) - as.numeric(as.Date("2020-
# Stay at home period
data1$home = data1$stay_at_home_expire_date - data1$stay_at_home_date
```

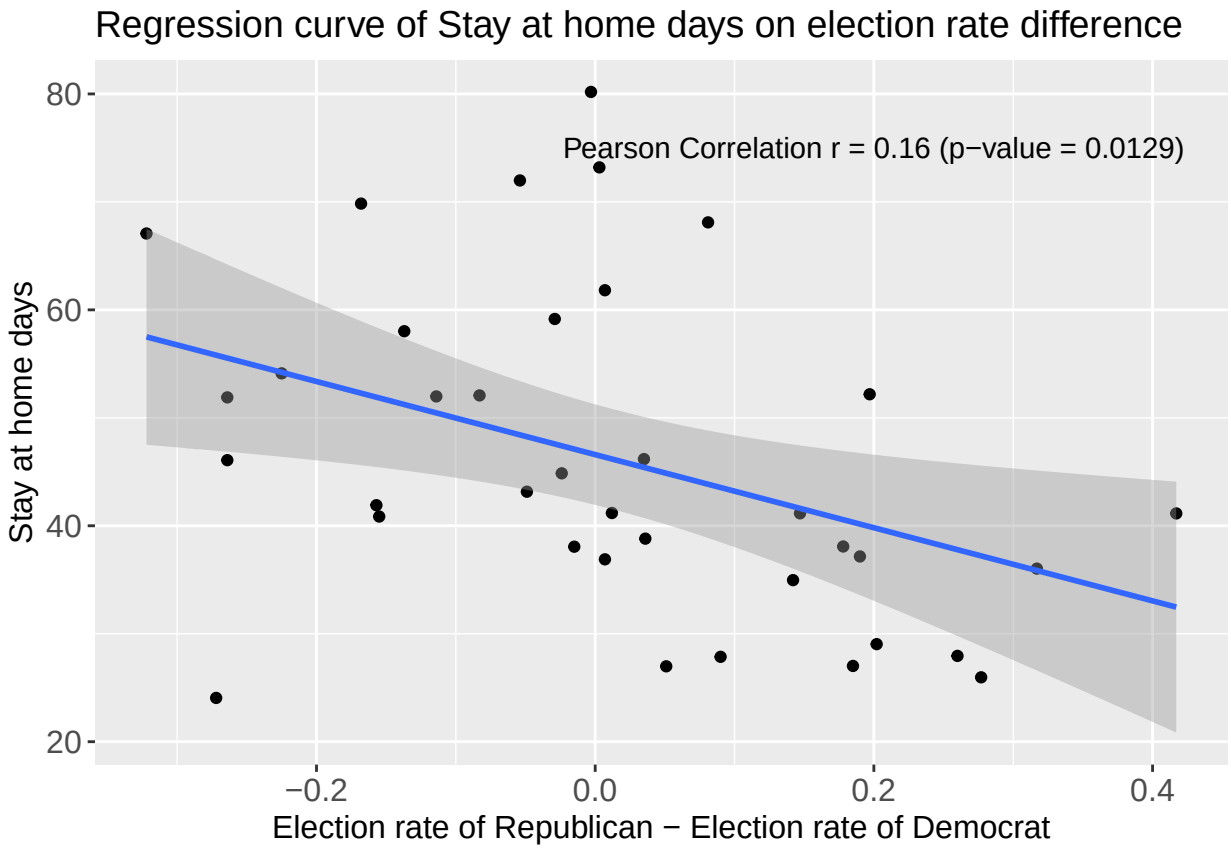
```
data1$home = jitter(data1$home)
ggboxplot(na.omit(data1), x = "reps", y = "home",
          color = "reps", palette = "jco")+
  stat_compare_means(comparisons = my_comparisons, label.y = c(100, 90, 95))+
  stat_compare_means(label.y = 105) + labs(x = "State Categories", y = "Stay at home days", title = "Bo
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```




```
ggplot(na.omit(data1), aes(x = rep, y = home)) + geom_point() + geom_smooth(method=lm) + labs(x = "Elec
```

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




```
lm1 = lm(home~rep, data1)
summary(lm1)
```

```
##
## Call:
## lm(formula = home ~ rep, data = data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.748  -9.462  -2.546   8.674  33.491
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   46.592      2.295  20.304   <2e-16 ***
## rep          -33.867     13.003  -2.605   0.0134 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.92 on 35 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.1624, Adjusted R-squared:  0.1384
## F-statistic: 6.784 on 1 and 35 DF,  p-value: 0.01341
```

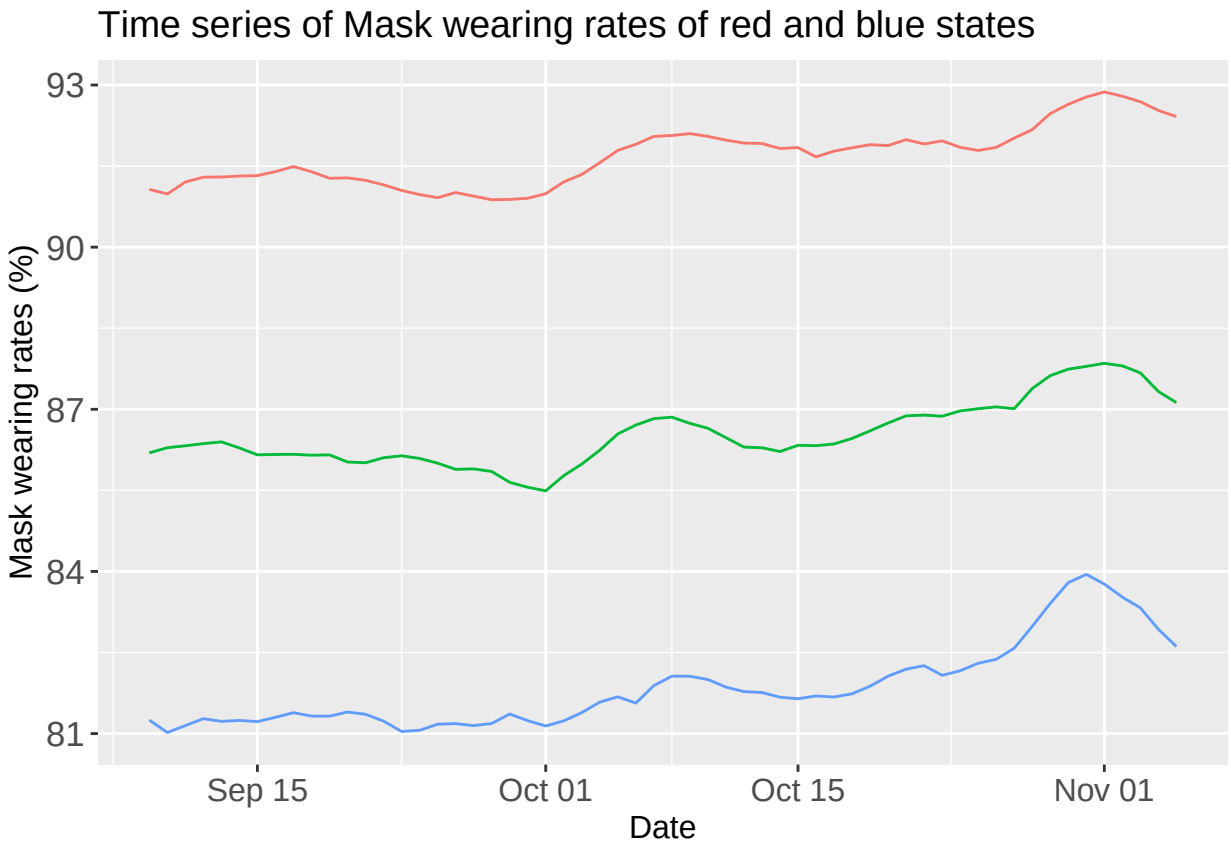
Time Series (Mask)

```
devtools::install_github("cmu-delphi/covidcast", ref = "main",subdir = "R-packages/covidcast")
```

```
## Skipping install of 'covidcast' from a github remote, the SHA1 (397ebb40) has not changed since last
##   Use `force = TRUE` to force installation
```

```
cip_mask <- suppressMessages(
  covidcast_signal(data_source = "fb-survey", signal = "smoothed_wearing_mask",
                   start_day = "2020-09-09", end_day = "2020-11-05",
                   geo_type = "state")
)
cip_mask = subset(cip_mask, select = c("geo_value","time_value","value"))
cip_mask <- tidyr::spread(cip_mask, geo_value, value)
mask_time = cip_mask[,2:52]
mask_time = unname(as.matrix(t(mask_time)))
mask_time = mask_time[c(2,1,4,3,5,6,7,9,10,11,12,14,15,16,13,17,18,19,22,21,20,23,24,26,25,27,30,34,31,3
population_state = unique(data$total)[-9]
sum1 = sum(population_state[data1$reps == "Red"])
sum2 = sum(population_state[data1$reps == "Blue"])
sum3 = sum(population_state[data1$reps == "Swing"])
weight_population = vector()
for(i in 1:50){
  if(data1$reps[i] == "Red"){
    weight_population[i] = population_state[i]/sum1
  }else if(data1$reps[i] == "Blue"){
    weight_population[i] = population_state[i]/sum2
  }else{
    weight_population[i] = population_state[i]/sum3
  }
}
red_prop = rep(0,58)
blue_prop = rep(0,58)
swing_prop = rep(0,58)
for(i in 1:50){
  if(data1$reps[i] == "Red"){
    red_prop = red_prop + mask_time[i,]*weight_population[i]
  }else if(data1$reps[i] == "Blue"){
    blue_prop = blue_prop + mask_time[i,]*weight_population[i]
  }else{
    swing_prop = swing_prop + mask_time[i,]*weight_population[i]
  }
}
red_prop = as.vector(as.matrix(red_prop))
blue_prop = as.vector(as.matrix(blue_prop))
swing_prop = as.vector(as.matrix(swing_prop))
date = seq(as.Date("2020-09-09"), as.Date("2020-11-05"), by="days")
mask_timeseries = as.data.frame(cbind(red_prop, blue_prop, swing_prop, date))
class(mask_timeseries$date) = "Date"

ggplot(mask_timeseries, aes(x = date)) + geom_line(aes(y=red_prop, colour = 'red')) + geom_line(aes(y=bl
```




```
ggsave("mask_rbs_ts.png", width = 10, height = 7)
```

Time Series(Daily confirmed proportion)

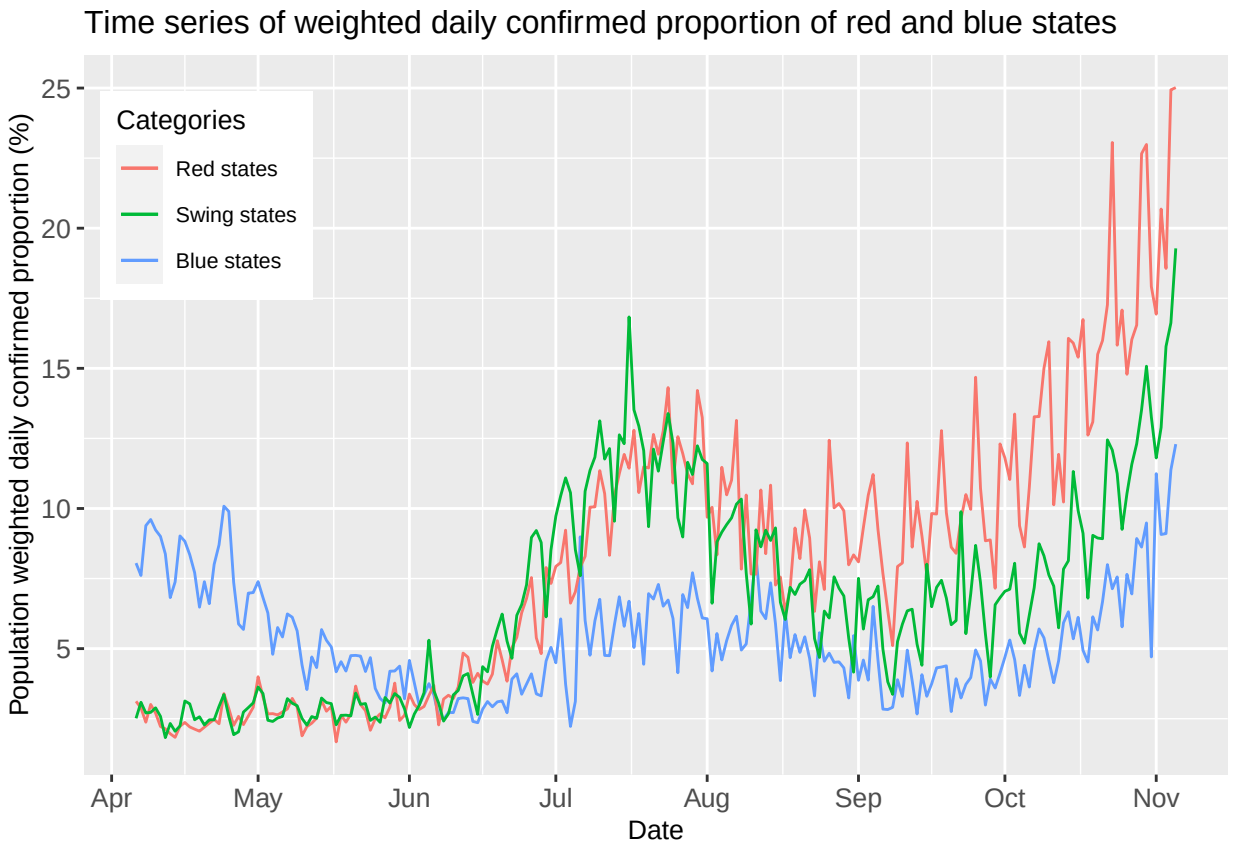
```
# Daily confirmed proportion
devtools::install_github("cmu-delphi/covidcast", ref = "main",
                         subdir = "R-packages/covidcast")
```

```
## Skipping install of 'covidcast' from a github remote, the SHA1 (397ebb40) has not changed since last
##   Use `force = TRUE` to force installation
```

```
cip_state <- suppressMessages(
  covidcast_signal(data_source = "indicator-combination", signal = "confirmed_incidence_prop",
                   start_day = "2020-04-06", end_day = "2020-11-05",
                   geo_type = "state")
)
cip_state <- subset(cip_state, select = c("geo_value","time_value","value"))
cip_state <- tidyr::spread(cip_state, geo_value, value)
confirmed_prop = cip_state[,2:53]
confirmed_prop = unname(as.matrix(t(confirmed_prop)))
confirmed_prop = confirmed_prop[c(2,1,4,3,5,6,7,9,10,11,12,14,15,16,13,17,18,19,22,21,20,23,24,26,25,27
population_state = unique(data$total)[-9]
sum1 = sum(population_state[data1$reps == "Red"])
sum2 = sum(population_state[data1$reps == "Blue"])
sum3 = sum(population_state[data1$reps == "Swing"])
```

```
weight_population = vector()
for(i in 1:50){
  if(data1$reps[i] == "Red"){
    weight_population[i] = population_state[i]/sum1
  }else if(data1$reps[i] == "Blue"){
    weight_population[i] = population_state[i]/sum2
  }else{
    weight_population[i] = population_state[i]/sum3
  }
}
red_prop = rep(0,214)
blue_prop = rep(0,214)
swing_prop = rep(0,214)
for(i in 1:50){
  if(data1$reps[i] == "Red"){
    red_prop = red_prop + confirmed_prop[i,]*weight_population[i]
  }else if(data1$reps[i] == "Blue"){
    blue_prop = blue_prop + confirmed_prop[i,]*weight_population[i]
  }else{
    swing_prop = swing_prop + confirmed_prop[i,]*weight_population[i]
  }
}
red_prop = as.vector(as.matrix(red_prop))
blue_prop = as.vector(as.matrix(blue_prop))
swing_prop = as.vector(as.matrix(swing_prop))
date = seq(as.Date("2020-04-06"), as.Date("2020-11-05"), by="days")
confirmed_time_prop = as.data.frame(cbind(red_prop, blue_prop, swing_prop, date))
class(confirmed_time_prop$date) = "Date"

ggplot(confirmed_time_prop, aes(x = date)) + geom_line(aes(y=blue_prop, colour = 'red')) + geom_line(aes
    legend.justification = c("right", "top"),
    legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6)) + scale_x_date(breaks = "1 month", date_labels = "%b")
```



** Correlation** ## Step 1: Daily Cases Data Manipulation

Download the 7-day average daily cases between 2020-7-2 to 2020-10-31.

```
devtools::install_github("cmu-delphi/covidcast", ref = "main",
                         subdir = "R-packages/covidcast")
```

```
## Skipping install of 'covidcast' from a github remote, the SHA1 (397ebb40) has not changed since last
##   Use `force = TRUE` to force installation
```

```
Date <- seq.Date(from = as.Date("2020/07/02",format = "%Y/%m/%d"), by = "day",
                 length.out = 122)

sevendaysaverage<- suppressMessages(
  covidcast_signal(data_source = "indicator-combination",
                   signal = "confirmed_7dav_incidence_prop",
                   start_day = "2020-07-2", end_day = "2020-10-31",
                   geo_type = "state")
)

A<-matrix(nrow=52,ncol=123)
dfsevenaverage<-(A)
colnames(dfsevenaverage)[2:123]<-as.character(Date)
colnames(dfsevenaverage)[1]<-'state'
dfsevenaverage[1:52,1]<-sevendaysaverage[1:52,3]
for (i in 1:52){
  for(j in 1:122){
```

```
    dfsevenaverage[i,j+1]=sevendaysaverage[i+(j-1)*52,7]

  }

}

dfseven50<-data.frame(dfsevenaverage[-c(8,40),])
print(head(dfseven50[,1:5]))
```

```
##   state     X2020.07.02     X2020.07.03     X2020.07.04     X2020.07.05
## 1    ak  1.786824948763  2.040690788478 2.4703099018418  2.333612911226
## 2    al 10.059059279107 11.193890152159 11.341024835315 12.413214209598
## 3    ar 9.5078597360389 9.1788598596861 9.2190972546357 9.4581547187482
## 4    az 23.938677261784 24.924918577193 24.044661403141 23.724746309784
## 5    ca 8.4617576981902 8.0613391094555 7.6356118966022 7.5493817987122
## 6    co 2.3244188109534 2.2512380693065 2.2847275612466 2.1830187338729
```

Step 2: Read Mask Data

Read the mask wearing percantage from local. It is got from NYtimes. https://www.nytimes.com/interactive/2020/07/17/upshot/coronavirus-face-mask-map.html

The mask wearing percentage for all states was collected between July 2 and July 14

```
covidfeatures<-read.xlsx('Mask wearing percentage.xlsx')
```

Step 3 Normalize the Mask Wearing Percantage and Numeric the Daily Cases

Normalize the mask wearing percantage and numeric the daily cases

```
covidlableuse<-as.matrix(dfseven50[2:length(dfseven50)])
covidfeaturesuse<-as.matrix(covidfeatures[2:length(covidfeatures)])

for(i in 1:dim(covidfeaturesuse)[2]){
  covidfeaturesuse[,i]=scale(covidfeaturesuse[,i],center=TRUE,scale=TRUE)
}

covidlableuse2<-matrix(nrow=50,ncol=122)

for (i in 1:50){
  covidlableuse2[i,]<-as.numeric(covidlableuse[i,])
}
```

Step 4 Get the Correlations and Draw the Correlations

Get the Correlations and Draw the Correlations The mask wearing percentage for all states was collected between July 2 and July 14 (in green shadow)

```
correlationvector<-c()

for(i in 1:122){correlationvector<-c(correlationvector,cor(covidlableuse2[,i],covidfeaturesuse[,1]))}


df<-data.frame(Date=Date,Correlation=correlationvector)

shadow <- data.frame (xmin=Date[1], xmax=Date[13], ymin=-Inf, ymax=Inf)


ggplot(data = df, mapping = aes(x = Date, y = Correlation, group = 1)) + geom_line() +
  xlab('Date')+
  labs(title = "Correlation between the mask wearing rate and daily confirmed cases")+
  geom_rect(data=shadow, aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill="green", alpha=0.1, inherit.aes = FALSE)
```