Massachusetts_Case

Karl Grindal

6/8/2020

Case 1: Massachusetts Data Security Law

```
library(plyr)
library(here)
```

```
## here() starts at C:/Users/karl_000/Documents/SpiderOak Hive/Dissertation/Dissertation_Code
```

```
##
## Attaching package: 'here'
```

```
## The following object is masked from 'package:plyr':
##
##     here
```

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

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

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

```
library(lubridate)
```

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

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

```
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
```

```
##   as.zoo.data.frame zoo
```

```
library(TTR)

AllStateClean <- read.table(here::here("Data","Other_data","AllStateClean.txt"),sep=";")

AllStateClean$MA_affected_residents <- gsub("\\,.*","",AllStateClean$MA_affected_residents) # this sele
AllStateClean$NC_affected_residents <- gsub("\\,.*","",AllStateClean$NC_affected_residents) # this sele

AllStateClean$Massachusetts[!is.na(AllStateClean$Massachusetts)]<-1
AllStateClean$New_Hampshire[!is.na(AllStateClean$New_Hampshire)]<-1
AllStateClean$North_Carolina[!is.na(AllStateClean$North_Carolina)]<-1
AllStateClean$Massachusetts[is.na(AllStateClean$Massachusetts)]<-0
AllStateClean$New_Hampshire[is.na(AllStateClean$New_Hampshire)]<-0
AllStateClean$North_Carolina[is.na(AllStateClean$North_Carolina)]<-0
```

Produce Tables on Overlapping Co-occurance of Breach Incidents

```
CoveredDays <- AllStateClean

CoveredDays$reported_date <- substr(CoveredDays$reported_date, start=1, stop=10)

CoveredDays <- subset(CoveredDays, reported_date > as.Date("2008-03-22") )
CoveredDays <- subset(CoveredDays, reported_date < as.Date("2010-09-01") )

table(CoveredDays$Massachusetts) # 0 = 313, 1 = 858
```

```
##
##   0   1
## 313 858
```

```
table(CoveredDays$New_Hampshire) # 0 = 949, 1 = 222
```

```
##
##   0   1
## 949 222
```

```
table(CoveredDays$North_Carolina) # 0 = 819, 1 = 352
```

```
##
##   0   1
## 819 352
```

```
table(CoveredDays$Massachusetts,CoveredDays$New_Hampshire) # 0 = 205 + 108 ; 1 = 744 + 114
```

```
##
##       0   1
##   0 205 108
##   1 744 114
```

```
table(CoveredDays$Massachusetts,CoveredDays$North_Carolina) # 0 = 99 + 214 ; 1 = 720 + 138
```

```
##
##       0   1
##   0  99 214
```

```
##   1 720 138
```

```
CoveredDays$MA_affected_residents <- as.numeric(CoveredDays$MA_affected_residents)
CoveredDays$NC_affected_residents <- as.numeric(CoveredDays$NC_affected_residents)

CoveredDays$MA_BigBreach <- CoveredDays$MA_affected_residents > 1000
CoveredDays$NC_BigBreech <- CoveredDays$NC_affected_residents > 1000

# Need to resolve this R issue to get to a fixed solution

# table(CoveredDays$MA_BigBreach,CoveredDays$NC_BigBreach)
```

Create Population Time Series for Matching with Incident Frequncy

```
# Creating blank frequency starting with earliest date

dat2 <- data.frame(seq(as.Date("2006-06-01"), by="1 month", length.out=174)) # treatment date
names(dat2) <- "yearmonth"
dat2 <- format(dat2,  "%Y/%m")

# Population
pop <- read.csv(here::here("Data","Other_data","populations.csv")) # starts at 2000.04.01
pop <- pop[c(22,30,34),] # Massachusetts row 22, New Hampshire row 30, North Carolina row 34 # removes

datforpop <- data.frame(seq(as.Date("2000-04-01"), by="1 month", length.out=(length(pop)-1)))
names(datforpop) <- "yearmonth"
datforpop <- format(datforpop,  "%Y/%m")
datforpop <- rbind("yearmonth",datforpop)
row.names(datforpop) <- 1:nrow(datforpop)

pop <- cbind(datforpop,t(pop))
colnames(pop) <- pop[1,]
pop <- pop[-1,]
rownames(pop) <- seq(1:nrow(pop))
pop <- as.data.frame(pop)
```

Identifying treatment and control options

```
# Case 1: Experiment 1
massachusetts <- dplyr::filter(AllStateClean,Massachusetts==1)
massachusetts$reported_date <- substr(massachusetts$reported_date, start=1, stop=10)

new_hampshire <- dplyr::filter(AllStateClean,New_Hampshire==1)
new_hampshire$reported_date <- substr(new_hampshire$reported_date, start=1, stop=10)

# Case 1: Experiment 2
massachusetts$MA_affected_residents <- as.numeric(massachusetts$MA_affected_residents)
massachusetts1000 <- subset(massachusetts, massachusetts$MA_affected_residents > 1000)
```

```r
n_carolina <- dplyr::filter(AllStateClean,North_Carolina==1)
n_carolina$NC_affected_residents <- as.numeric(n_carolina$NC_affected_residents)
n_carolina1000 <- subset(n_carolina, n_carolina$NC_affected_residents > 1000)
```

Experiment 1: Create control and treatment populations (Massachusetts v New Hampshire) with Total Incidents

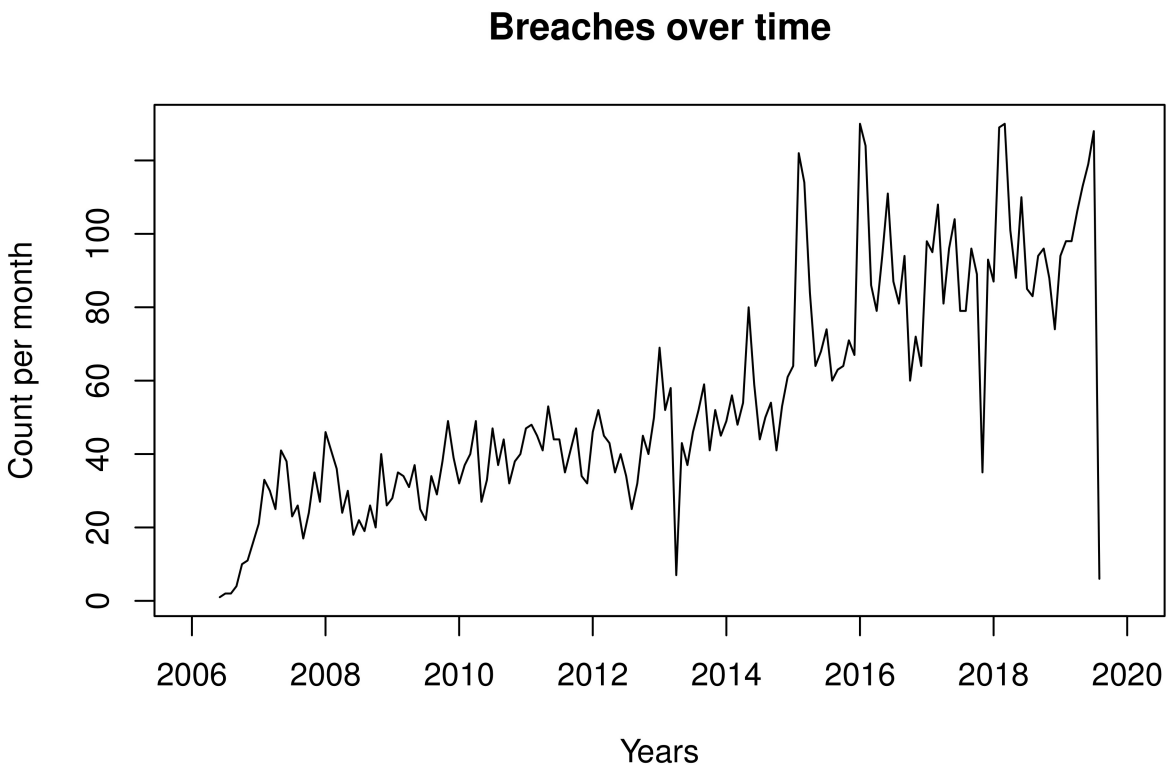
```r
treatment <- massachusetts # This Must Be Filled in to Work Properly!
control <- new_hampshire  # This Must Be Filled in to Work Properly!

# Format treatment dates into months
treatment$date_formatted <- format(as.Date(treatment$reported_date, "%Y-%m-%d"), "%Y/%m") # Alternative
treatment_freq <- treatment %>%
  dplyr::group_by(treatment$date_formatted) %>%
  dplyr::summarise(frequency = n(),)
```

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

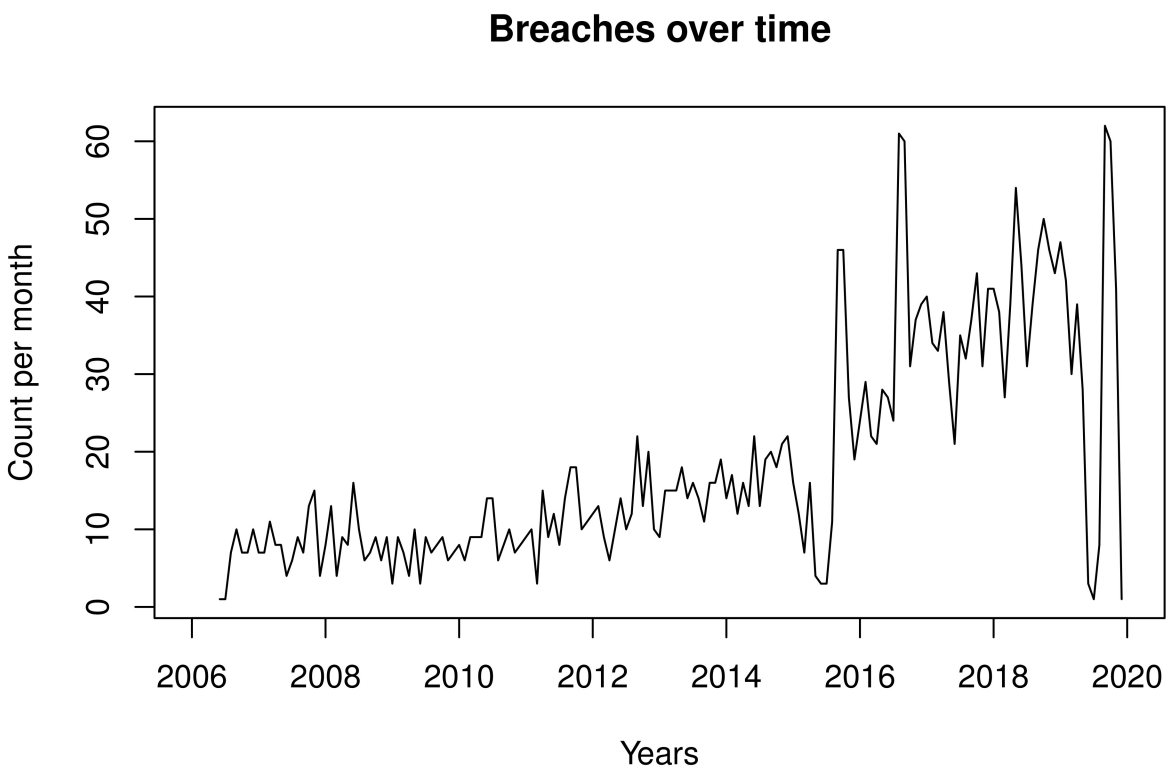
```r
names(treatment_freq)[1] <- "yearmonth"
treatment_freq$frequency[is.na(treatment_freq$frequency)]<-0

# Format control dates into months
control$date_formatted <- format(as.Date(control$reported_date, "%Y-%m-%d"), "%Y/%m") # Alternative is
control_freq <- control %>%
  dplyr::group_by(control$date_formatted) %>%
  dplyr::summarise(frequency = n())
```

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

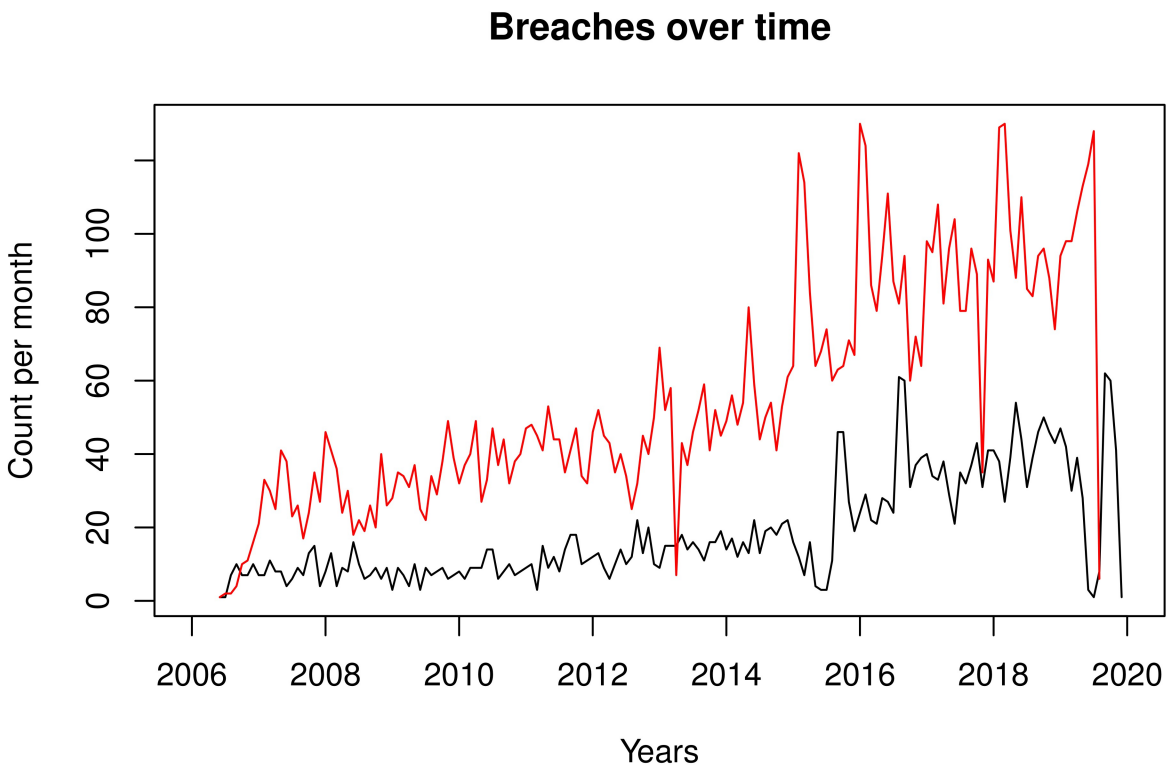
```r
names(control_freq)[1] <- "yearmonth"
control_freq$frequency[is.na(control_freq$frequency)]<-0

treatment_ts <- ts(treatment_freq$frequency, frequency = 12, start = c(2006,6))
control_ts <- ts(control_freq$frequency, frequency = 12, start = c(2006,6))

plot.ts(treatment_ts, main = "Breaches over time", xlim=c(2006,2020), xlab = "Years", ylab = "Count per
```




```
plot.ts(control_ts, main = "Breaches over time", xlim=c(2006,2020), xlab = "Years", ylab = "Count per m
```




```
ts.plot(control_ts, treatment_ts, main = "Breaches over time", xlim=c(2006,2020), gpars = list(col = c(
```




```
summary(treatment_ts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   34.00   46.00   55.19   79.00  130.00
```

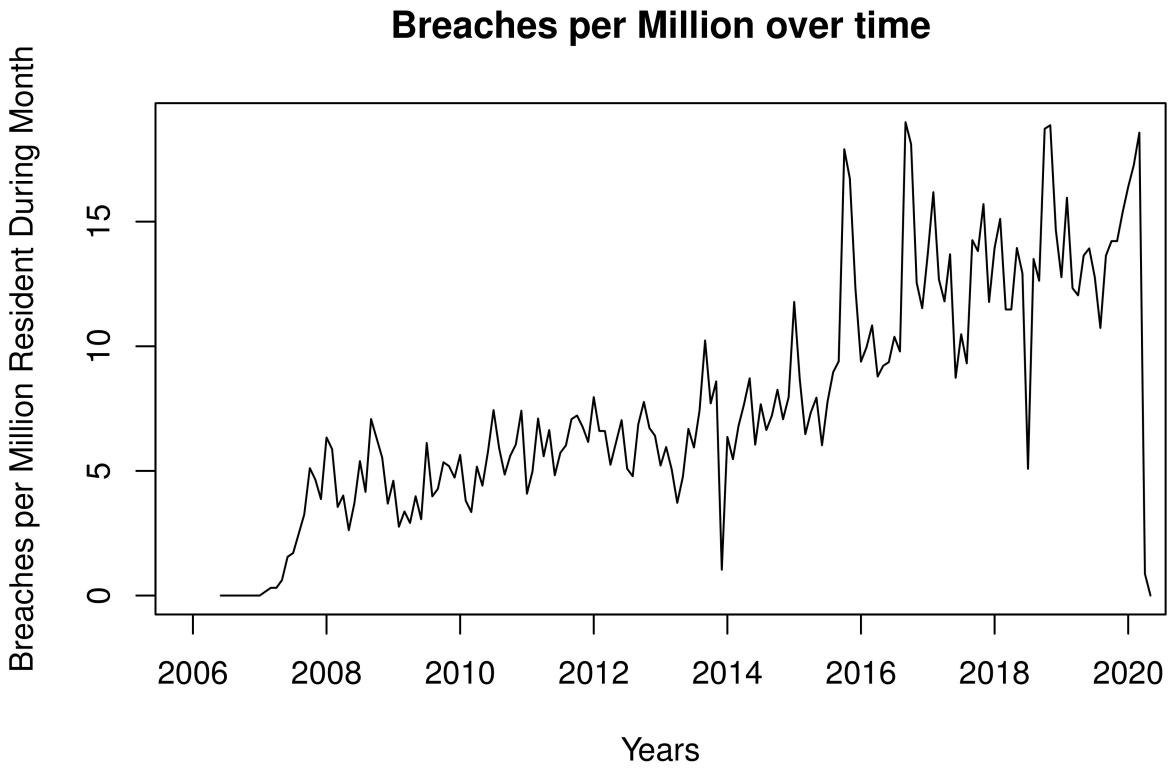
```
summary(control_ts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    8.00   13.00   18.39   27.00   62.00
```

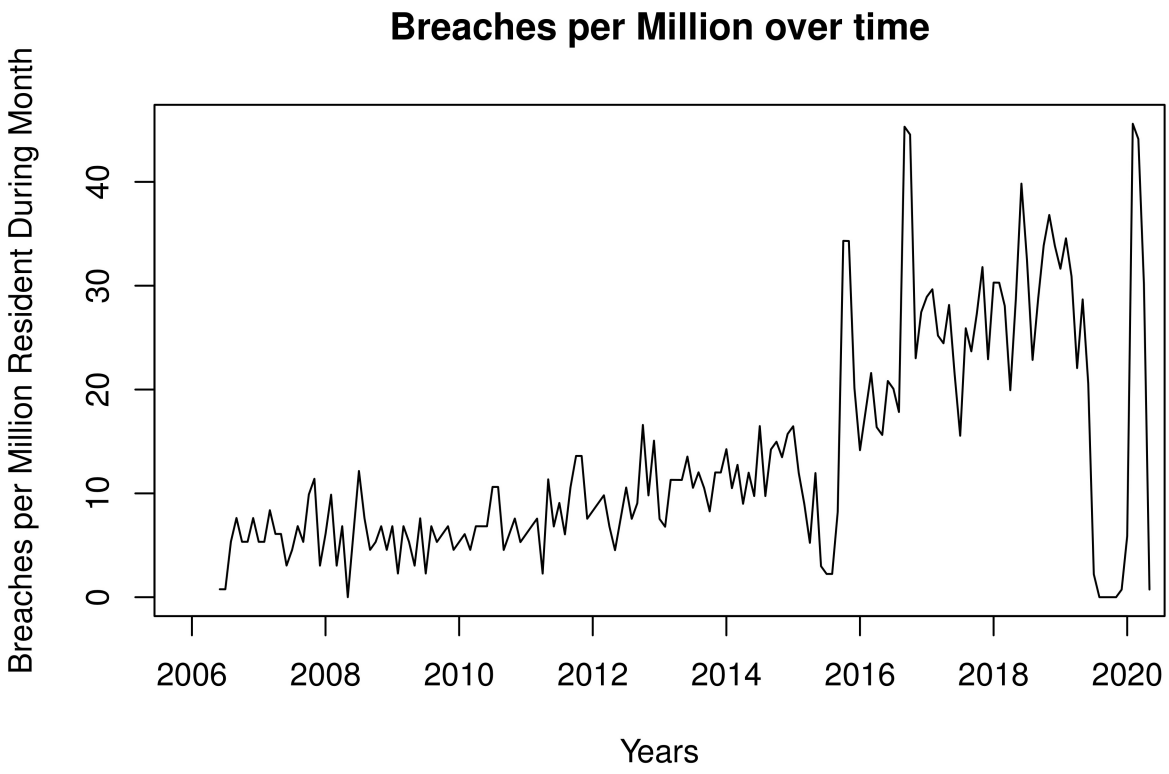
Create charts with breaches per million residents

```
# Merge Treatment and Control Together
comb_ts <- merge(treatment_freq, control_freq, by="yearmonth", all=TRUE)

# Merge Combined Treatment and Control Together with Population Statistics
comb_ts <- merge(comb_ts, pop, by='yearmonth', all.x = TRUE)
comb_ts$frequency.x[is.na(comb_ts$frequency.x)]<-0
comb_ts$frequency.y[is.na(comb_ts$frequency.y)]<-0

comb_ts$Massachusetts <- as.numeric(as.character(comb_ts$Massachusetts))
comb_ts$treatpermil <- comb_ts$frequency.x/(comb_ts$Massachusetts/1000000)
class(comb_ts$frequency.y)<-"numeric"
comb_ts$`New Hampshire` <- as.numeric(as.character(comb_ts$`New Hampshire`))
comb_ts$controlpermil <- comb_ts$frequency.y/(comb_ts$`New Hampshire`/1000000)
```

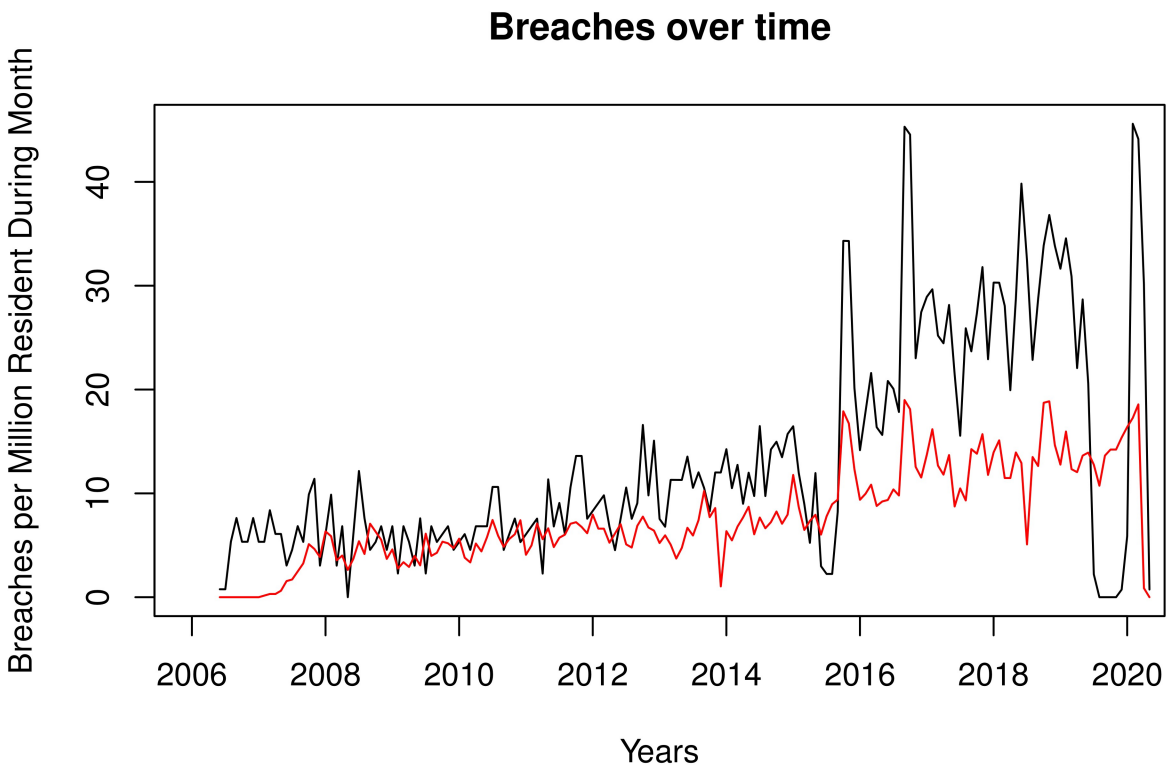
```r
treatment_tsM <- ts(comb_ts$treatpermil, frequency = 12, start = c(2006,6))
control_tsM <- ts(comb_ts$controlpermil, frequency = 12, start = c(2006,6))

plot.ts(treatment_tsM, main = "Breaches per Million over time", xlim=c(2006,2020), xlab = "Years", ylab
```




```r
plot.ts(control_tsM, main = "Breaches per Million over time", xlim=c(2006,2020), xlab = "Years", ylab =
```




```
ts.plot(control_tsM, treatment_tsM, main = "Breaches over time", xlim=c(2006,2020), gpars = list(col = c
```



Identifying and subsetting relevant dates

```
# Legislation H.B. 4144 signed into law August 3, 2007
# Legislation H.B. 4144 becomes effective on October 31, 2007
# OCABR finalized the regulation on September 22, 2008

treatment_start <- as.Date("09/22/2008", "%m/%d/%Y") # Legislation H.B. 4144 becomes effective
treatment_start<- format(as.Date(as.character(treatment_start), origin = "1970-01-01"), "%Y/%m")

treatment_end <- as.Date("03/01/2010", "%m/%d/%Y")+180 # post 6 months after enforcement
treatment_end<- format(as.Date(as.character(treatment_end), origin = "1970-01-01"), "%Y/%m")

pretreat <- comb_ts[(which(comb_ts$yearmonth==treatment_start)-5):which(comb_ts$yearmonth==treatment_sta
pretreat$type <- "pretest"

posttreat <- comb_ts[which(comb_ts$yearmonth==treatment_end):(which(comb_ts$yearmonth==treatment_end)+5
posttreat$type <- "posttest"

mean(posttreat$treatpermil) - mean(pretreat$treatpermil)
```

```
## [1] 0.6274941
```

```
mean(posttreat$controlpermil) - mean(pretreat$controlpermil)
```

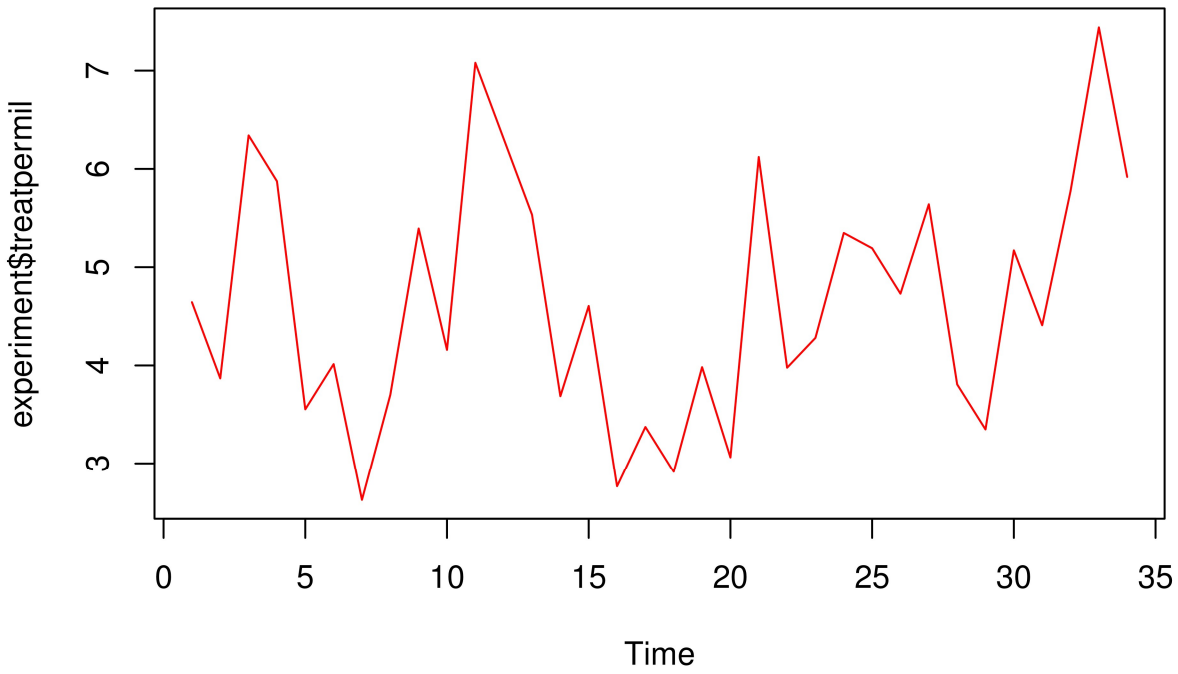
```
## [1] 0.9999738
```

```r
treatment_range <- comb_ts[(which(comb_ts$yearmonth==treatment_start)+1):(which(comb_ts$yearmonth==trea
treatment_range$type <- "test"

experiment <- rbind(pretreat,treatment_range,posttreat)
experiment$treatpermil[is.na(experiment$treatpermil)]<-0
experiment$controlpermil[is.na(experiment$controlpermil)]<-0
experiment
```

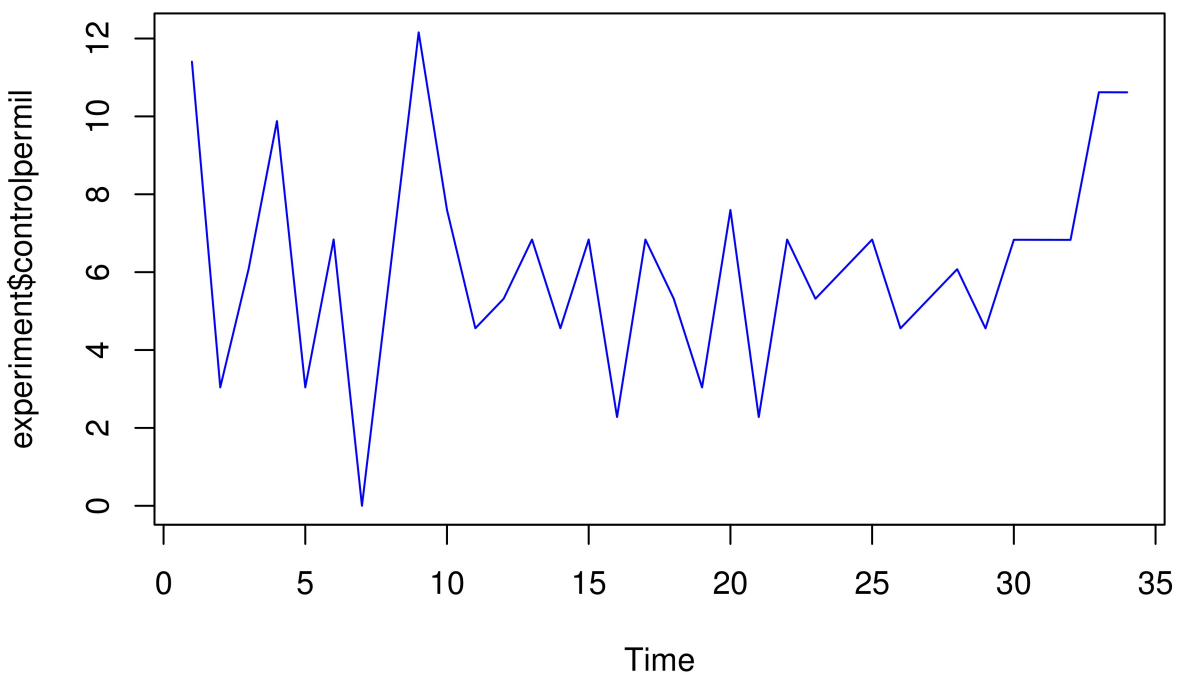
```
##    yearmonth frequency.x frequency.y Massachusetts New Hampshire North Carolina
## 18   2008/04          30          15       6459615       1315064        9261596
## 19   2008/05          25           4       6462732       1315345        9277547
## 20   2008/06          41           8       6465850       1315626        9293498
## 21   2008/07          38          13       6468967       1315906        9309449
## 22   2008/08          23           4       6473021       1315922        9321125
## 23   2008/09          26           9       6477075       1315939        9332802
## 24   2008/10          17           0       6481128       1315955        9344478
## 25   2008/11          24           8       6485182       1315971        9356155
## 26   2008/12          35          16       6489236       1315988        9367831
## 27   2009/01          27          10       6493290       1316004        9379508
## 28   2009/02          46           6       6497344       1316020        9391184
## 29   2009/03          41           7       6501398       1316037        9402860
## 30   2009/04          36           9       6505452       1316053        9414537
## 31   2009/05          24           6       6509505       1316069        9426213
## 32   2009/06          30           9       6513559       1316086        9437890
## 33   2009/07          18           3       6517613       1316102        9449566
## 34   2009/08          22           9       6520948       1316143        9459112
## 35   2009/09          19           7       6524283       1316184        9468659
## 36   2009/10          26           4       6527618       1316225        9478205
## 37   2009/11          20          10       6530953       1316266        9487751
## 38   2009/12          40           3       6534289       1316306        9497298
## 39   2010/01          26           9       6537624       1316347        9506844
## 40   2010/02          28           7       6540959       1316388        9516390
## 41   2010/03          35           8       6544294       1316429        9525937
## 42   2010/04          34           9       6547629       1316470        9535483
## 43   2010/05          31           6       6553855       1316567        9548430
## 44   2010/06          37           7       6560081       1316665        9561376
## 45   2010/07          25           8       6566307       1316762        9574323
## 46   2010/08          22           6       6570247       1317049        9581262
## 47   2010/09          34           9       6574186       1317335        9588201
## 48   2010/10          29           9       6578126       1317622        9595140
## 49   2010/11          38           9       6582066       1317909        9602079
## 50   2010/12          49          14       6586005       1318195        9609018
## 51   2011/01          39          14       6589945       1318482        9615958
##    treatpermil controlpermil     type
## 18    4.644240     11.406289  pretest
## 19    3.868333      3.041027  pretest
## 20    6.341007      6.080755  pretest
## 21    5.874199      9.879125  pretest
## 22    3.553210      3.039694  pretest
## 23    4.014158      6.839223  pretest
## 24    2.623000      0.000000     test
## 25    3.700744      6.079161     test
## 26    5.393547     12.158166     test
## 27    4.158139      7.598761     test
```

```
## 28    7.079816      4.559201     test
## 29    6.306336      5.318999     test
## 30    5.533820      6.838630     test
## 31    3.686916      4.559031     test
## 32    4.605777      6.838459     test
## 33    2.761747      2.279459     test
## 34    3.373743      6.838163     test
## 35    2.912197      5.318405     test
## 36    3.983076      3.038994     test
## 37    3.062340      7.597249     test
## 38    6.121554      2.279105     test
## 39    3.976980      6.837103     test
## 40    4.280718      5.317581     test
## 41    5.348170      6.077046     test
## 42    5.192719      6.836464     test
## 43    4.730041      4.557307     test
## 44    5.640174      5.316462     test
## 45    3.807315      6.075509     test
## 46    3.348428      4.555639 posttest
## 47    5.171743      6.831975 posttest
## 48    4.408550      6.830487 posttest
## 49    5.773263      6.829000 posttest
## 50    7.440019     10.620583 posttest
## 51    5.918107     10.618272 posttest
```

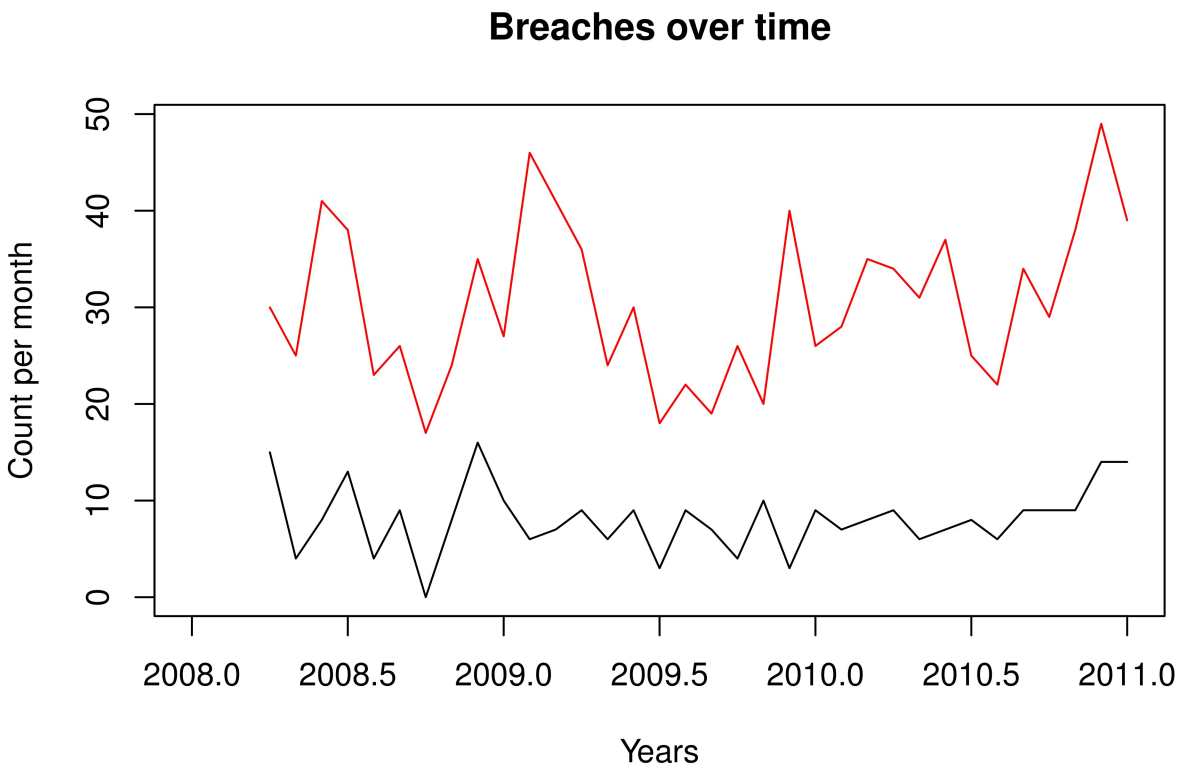
```
ts.plot(experiment$treatpermil, col = "red")
```



```
ts.plot(experiment$controlpermil, col = "blue")
```




```
# Look at Raw Frequency Counts

treatment_ts <- ts(experiment$frequency.x, frequency = 12, start = c(2008,4))
control_ts <- ts(experiment$frequency.y, frequency = 12, start = c(2008,4))
ts.plot(control_ts, treatment_ts, main = "Breaches over time", xlim=c(2008,2011), gpars = list(col = c(
```




```
# Look at Treatment and Control per Million

treatment_tsM <- ts(experiment$treatpermil, frequency = 12, start = c(2008,4))
mean(treatment_tsM)
```

```
## [1] 4.66571
```

```
sd(treatment_tsM)
```

```
## [1] 1.253152
```

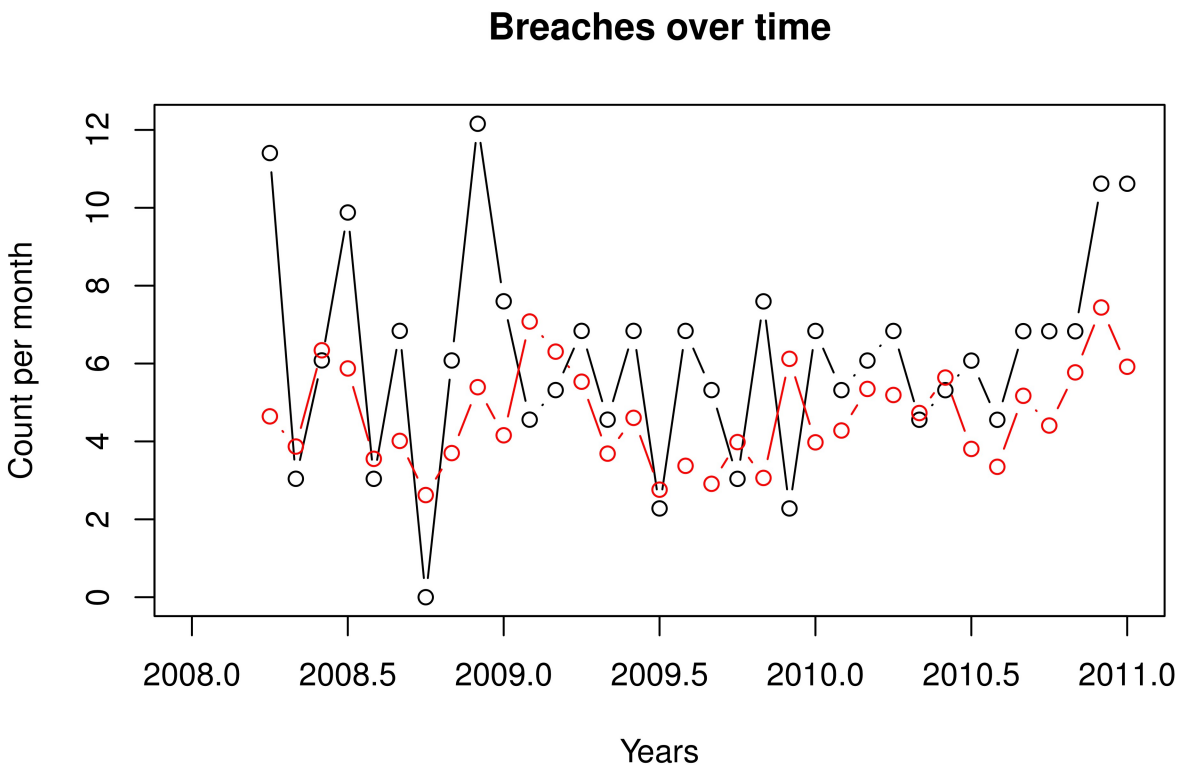
```
control_tsM <- ts(experiment$controlpermil, frequency = 12, start = c(2008,4))
mean(control_tsM)
```

```
## [1] 6.143863
```

```
sd(control_tsM)
```

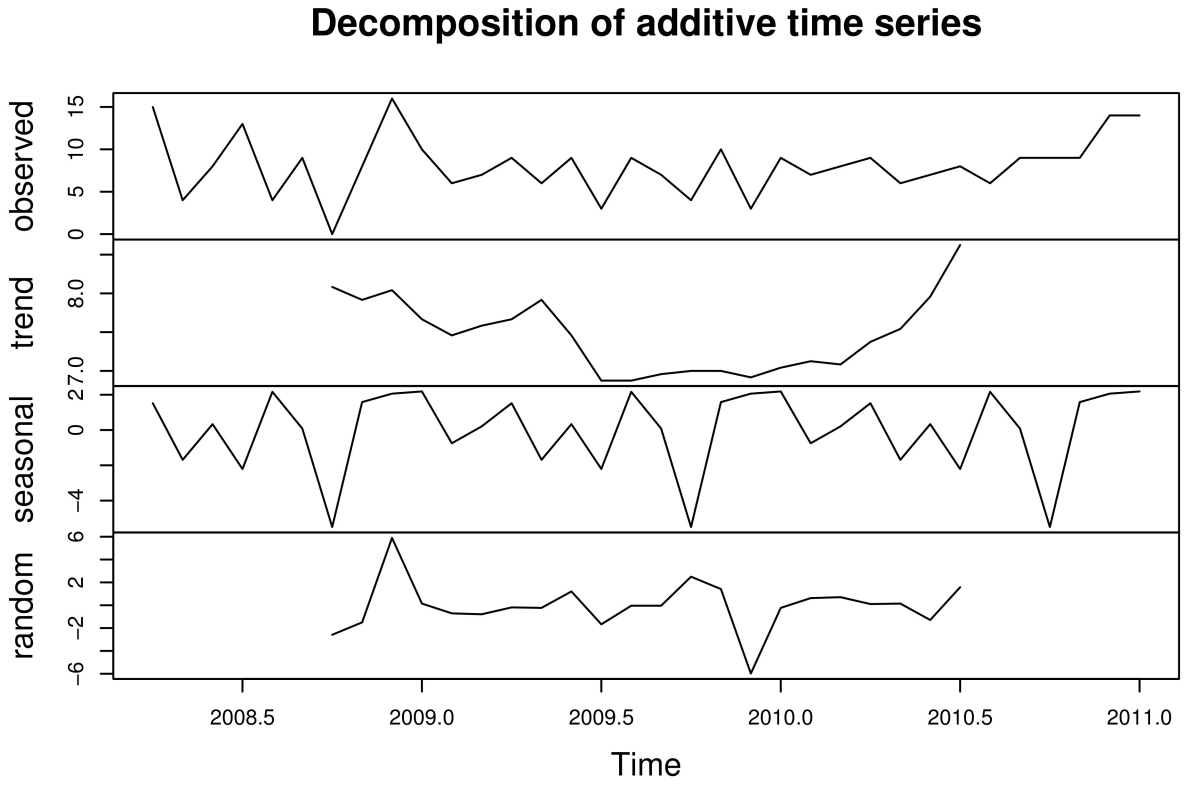
```
## [1] 2.672772
```

```
ts.plot(control_tsM, treatment_tsM, main = "Breaches over time", xlim=c(2008,2011),
                    gpars = list(col = c("black", "red")), type = "b", xlab = "Years", ylab = "Count pe
```

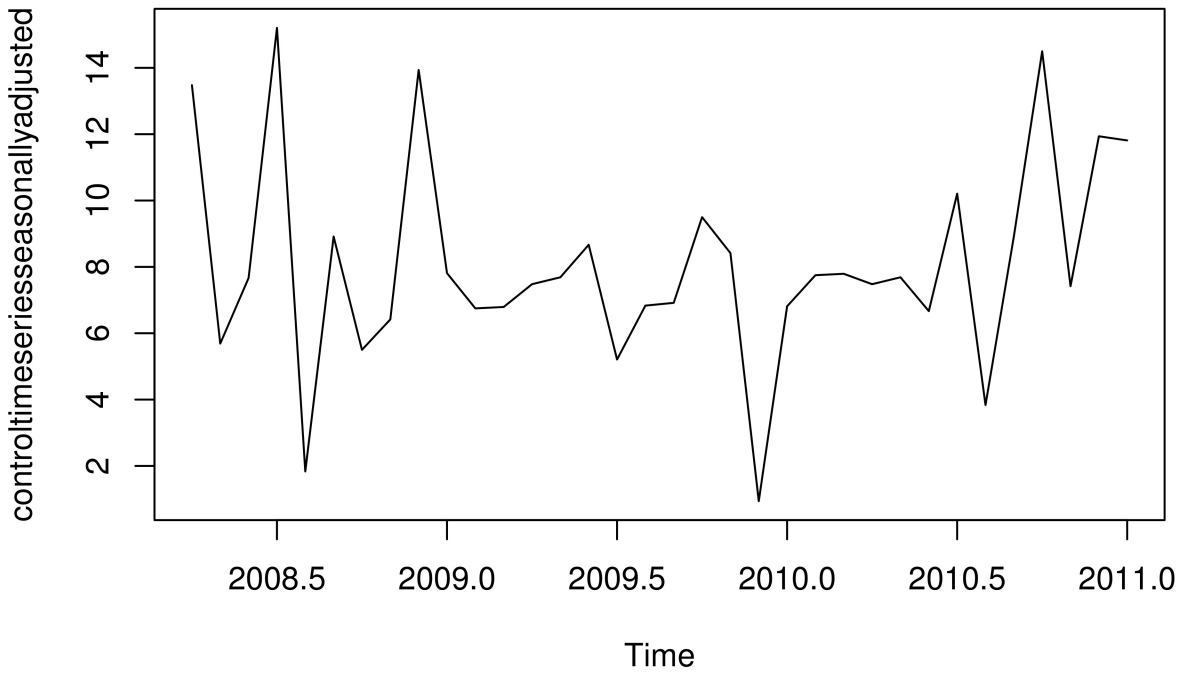


Decompose the Massachusetts Data to Find Seasonal Patterns

```
controltimeseriescomponents <- decompose(control_ts)
plot(controltimeseriescomponents)
```

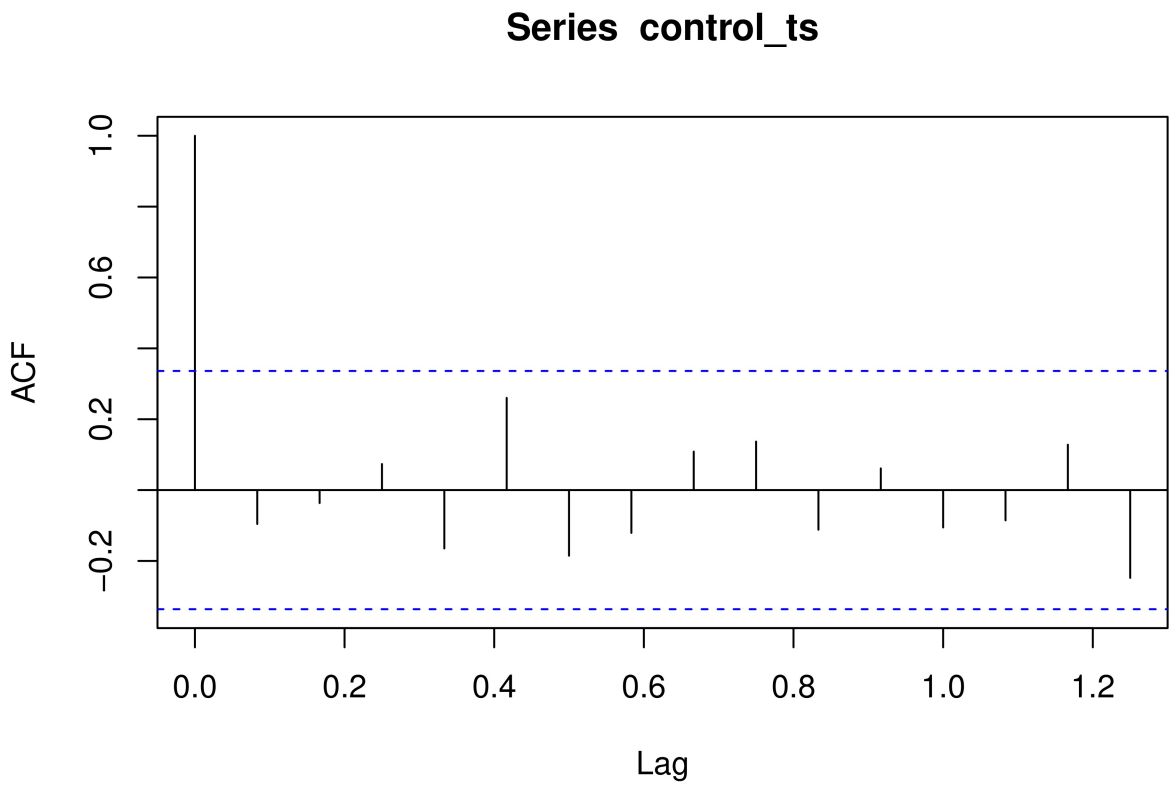



```
controltimeseriesseasonallyadjusted <- control_ts - controltimeseriescomponents$seasonal
plot(controltimeseriesseasonallyadjusted)
```

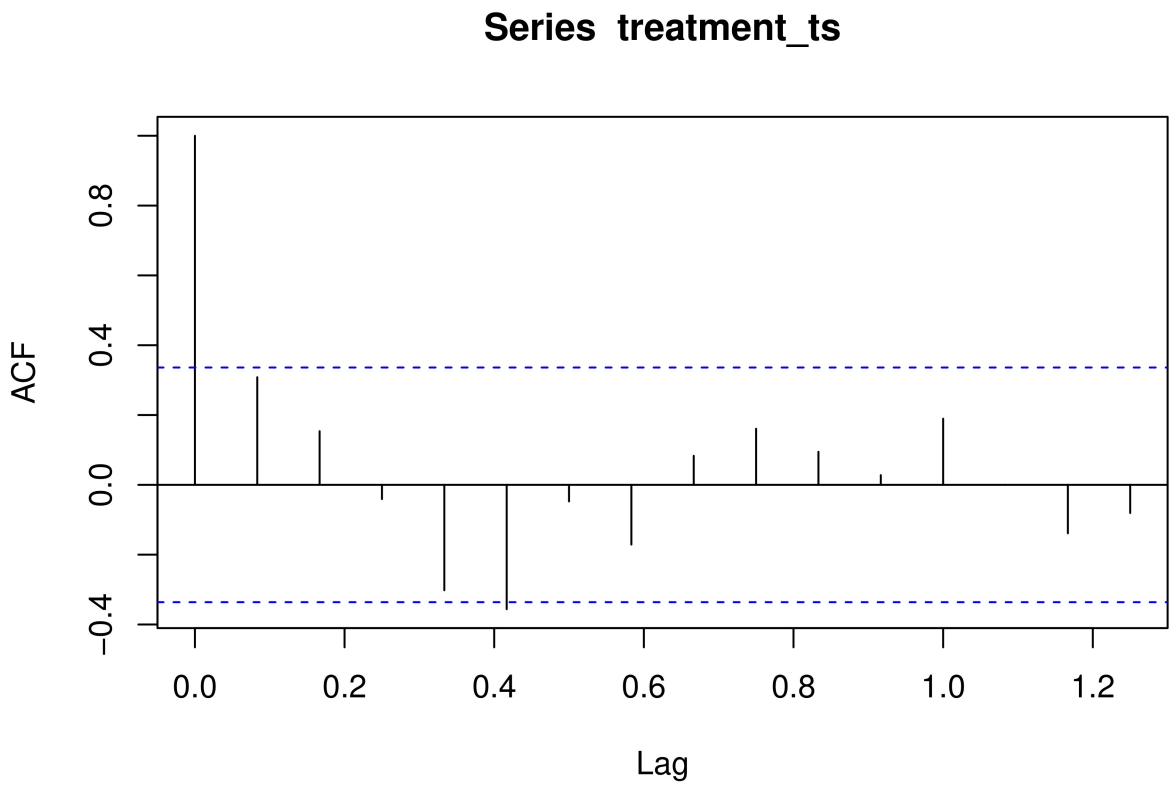


Run Statistical Tests on Time Series for Stationarity

```
# source of statistical tests http://r-statistics.co/Time-Series-Analysis-With-R.html

acfcontrolNH <- acf(control_ts) # autocorrelation (i.e. a Time Series with lags of itself)
```




```
acftreatmentMA <- acf(treatment_ts)
```




```
# shows that the control time series is a "stationary time series"

png(here::here("Output","acfcontrolNH.png"))
plot(acfcontrolNH)

png(here::here("Output","acttreatmentMA.png"))
plot(acftreatmentMA)

pacfcontrolNH <- pacf(control_ts)  # partial autocorrelation (i.e. correlation of the time series with
pacftreatmentMA <- pacf(treatment_ts)  # partial autocorrelation (i.e. correlation of the time series w

plot(pacfcontrolNH)

png(here::here("Output","pacftreatmentNH.png"))
plot(pacftreatmentMA)

ccfRes <- ccf(control_ts, treatment_ts, ylab = "cross-correlation")
ccfRes
```

```
##
## Autocorrelations of series 'X', by lag
##
## -1.0000 -0.9167 -0.8333 -0.7500 -0.6667 -0.5833 -0.5000 -0.4167 -0.3333 -0.2500
##   0.094   0.055   0.060  -0.059   0.009  -0.098  -0.229  -0.117  -0.145   0.200
## -0.1667 -0.0833  0.0000  0.0833  0.1667  0.2500  0.3333  0.4167  0.5000  0.5833
##   0.295   0.074   0.412   0.267  -0.074  -0.045  -0.142   0.006   0.148   0.026
##  0.6667  0.7500  0.8333  0.9167  1.0000
```

```
##  -0.068   0.177  -0.038   0.075   0.024
```

```
# adf test is an Augmented Dickey-Fuller Test
adf.test(control_ts) # p-value < 0.05 indicates the TS is stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  control_ts
## Dickey-Fuller = -2.7292, Lag order = 3, p-value = 0.2905
## alternative hypothesis: stationary
```

```
adf.test(treatment_ts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  treatment_ts
## Dickey-Fuller = -3.5501, Lag order = 3, p-value = 0.05258
## alternative hypothesis: stationary
```

```
kpss.test(control_ts) # Kwiatkowski-Phillips-Schmidt-Shin (KPSS) testz
```

```
## Warning in kpss.test(control_ts): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  control_ts
## KPSS Level = 0.16838, Truncation lag parameter = 3, p-value = 0.1
```

```
kpss.test(treatment_ts)
```

```
## Warning in kpss.test(treatment_ts): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  treatment_ts
## KPSS Level = 0.14572, Truncation lag parameter = 3, p-value = 0.1
```

```
# https://www.sas.com/content/dam/SAS/en_ca/User%20Group%20Presentations/Health-User-Groups/ITS_SAS.pdf
```

ITS analyses use regression-based techniques

```
quasiexp <- experiment[experiment$type != "test",]
quasiexp
```

```
##    yearmonth frequency.x frequency.y Massachusetts New Hampshire North Carolina
## 18   2008/04          30          15       6459615       1315064        9261596
## 19   2008/05          25           4       6462732       1315345        9277547
## 20   2008/06          41           8       6465850       1315626        9293498
## 21   2008/07          38          13       6468967       1315906        9309449
## 22   2008/08          23           4       6473021       1315922        9321125
## 23   2008/09          26           9       6477075       1315939        9332802
## 46   2010/08          22           6       6570247       1317049        9581262
## 47   2010/09          34           9       6574186       1317335        9588201
```

```
## 48   2010/10          29           9       6578126       1317622        9595140
## 49   2010/11          38           9       6582066       1317909        9602079
## 50   2010/12          49          14       6586005       1318195        9609018
## 51   2011/01          39          14       6589945       1318482        9615958
##    treatpermil controlpermil     type
## 18    4.644240     11.406289  pretest
## 19    3.868333      3.041027  pretest
## 20    6.341007      6.080755  pretest
## 21    5.874199      9.879125  pretest
## 22    3.553210      3.039694  pretest
## 23    4.014158      6.839223  pretest
## 46    3.348428      4.555639 posttest
## 47    5.171743      6.831975 posttest
## 48    4.408550      6.830487 posttest
## 49    5.773263      6.829000 posttest
## 50    7.440019     10.620583 posttest
## 51    5.918107     10.618272 posttest
```

```
# Added dummy variables for ITS
control <- as.data.frame(t(rbind(quasiexp$yearmonth,quasiexp$controlpermil)))
control$treat <- as.vector(rep(0,nrow(control)))                  # Create example vector
time <- 1:nrow(control)
control$time <- as.vector(time)
control$z <- c(rep(0,6),1:(nrow(control)-6))

treatment <- as.data.frame(t(rbind(quasiexp$yearmonth,quasiexp$treatpermil)))
treatment$treat <- as.vector(rep(1,nrow(control)))                  # Create example vector
time <- 1:nrow(control)
treatment$time <- as.vector(time)
treatment$z <- c(rep(0,6),1:(nrow(control)-6))
treatment
```

```
##         V1               V2 treat time z
## 1  2008/04 4.64423963347661     1    1 0
## 2  2008/05 3.86833308266535     1    2 0
## 3  2008/06 6.34100698284062     1    3 0
## 4  2008/07 5.87419908000767     1    4 0
## 5  2008/08 3.55320954466238     1    5 0
## 6  2008/09 4.01415762516259     1    6 0
## 7  2010/08 3.34842814889608     1    7 1
## 8  2010/09 5.17174293517099     1    8 2
## 9  2010/10 4.40855039870018     1    9 3
## 10 2010/11 5.77326328845685     1   10 4
## 11 2010/12 7.44001864559775     1   11 5
## 12 2011/01 5.91810705552171     1   12 6
```

```
AppendITS <- rbind(treatment,control)
names(AppendITS) <- c("yearmonth","incident_permil","treat","time","z")
AppendITS$incident_permil <- as.numeric(as.character(AppendITS$incident_permil))
AppendITS$time <- as.numeric(as.character(AppendITS$time))
AppendITS$z <- as.numeric(as.character(AppendITS$z))
AppendITS
```

```
##    yearmonth incident_permil treat time z
## 1    2008/04        4.644240     1    1 0
## 2    2008/05        3.868333     1    2 0
```

```
## 3    2008/06        6.341007     1    3 0
## 4    2008/07        5.874199     1    4 0
## 5    2008/08        3.553210     1    5 0
## 6    2008/09        4.014158     1    6 0
## 7    2010/08        3.348428     1    7 1
## 8    2010/09        5.171743     1    8 2
## 9    2010/10        4.408550     1    9 3
## 10   2010/11        5.773263     1   10 4
## 11   2010/12        7.440019     1   11 5
## 12   2011/01        5.918107     1   12 6
## 13   2008/04       11.406289     0    1 0
## 14   2008/05        3.041027     0    2 0
## 15   2008/06        6.080755     0    3 0
## 16   2008/07        9.879125     0    4 0
## 17   2008/08        3.039694     0    5 0
## 18   2008/09        6.839223     0    6 0
## 19   2010/08        4.555639     0    7 1
## 20   2010/09        6.831975     0    8 2
## 21   2010/10        6.830487     0    9 3
## 22   2010/11        6.829000     0   10 4
## 23   2010/12       10.620583     0   11 5
## 24   2011/01       10.618272     0   12 6
```

```
factor_cols <- c("treat","time","z")

sapply(AppendITS, class)
```

```
##       yearmonth incident_permil           treat            time               z
##     "character"       "numeric"       "numeric"       "numeric"       "numeric"
```

```
regTest <- lm(incident_permil ~ time + treat + z, AppendITS)
summary(regTest)
```

```
##
## Call:
## lm(formula = incident_permil ~ time + treat + z, data = AppendITS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3229 -1.0957 -0.1703  1.1921  3.5534
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.3420     1.2881   6.476 2.58e-06 ***
## time         -0.4890     0.2846  -1.719   0.1011
## treat        -2.1847     0.7973  -2.740   0.0126 *
## z             1.1721     0.4620   2.537   0.0196 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.953 on 20 degrees of freedom
## Multiple R-squared:  0.4465, Adjusted R-squared:  0.3634
## F-statistic: 5.377 on 3 and 20 DF,  p-value: 0.007037
```

```
regTest2 <- lm(incident_permil ~ time + treat + time*treat + z + z*time + z*treat + z*treat*time, Appen
summary(regTest2)
```

```
##
## Call:
## lm(formula = incident_permil ~ time + treat + time * treat +
##     z + z * time + z * treat + z * treat * time, data = AppendITS)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5044 -0.8769 -0.2531  1.2697  3.5121
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.7239     1.9058   4.578  0.00031 ***
## time          -0.5892     0.4735  -1.244  0.23131
## treat         -3.3444     2.6952  -1.241  0.23254
## z             -0.1885     2.7195  -0.069  0.94559
## time:treat     0.3700     0.6697   0.553  0.58824
## time:z         0.1466     0.2084   0.703  0.49190
## treat:z        0.4889     3.8460   0.127  0.90043
## time:treat:z  -0.1162     0.2947  -0.394  0.69858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.083 on 16 degrees of freedom
## Multiple R-squared:  0.4961, Adjusted R-squared:  0.2756
## F-statistic:  2.25 on 7 and 16 DF,  p-value: 0.08494
```

```
AppendITS
```

```
##    yearmonth incident_permil treat time z
## 1    2008/04        4.644240     1    1 0
## 2    2008/05        3.868333     1    2 0
## 3    2008/06        6.341007     1    3 0
## 4    2008/07        5.874199     1    4 0
## 5    2008/08        3.553210     1    5 0
## 6    2008/09        4.014158     1    6 0
## 7    2010/08        3.348428     1    7 1
## 8    2010/09        5.171743     1    8 2
## 9    2010/10        4.408550     1    9 3
## 10   2010/11        5.773263     1   10 4
## 11   2010/12        7.440019     1   11 5
## 12   2011/01        5.918107     1   12 6
## 13   2008/04       11.406289     0    1 0
## 14   2008/05        3.041027     0    2 0
## 15   2008/06        6.080755     0    3 0
## 16   2008/07        9.879125     0    4 0
## 17   2008/08        3.039694     0    5 0
## 18   2008/09        6.839223     0    6 0
## 19   2010/08        4.555639     0    7 1
## 20   2010/09        6.831975     0    8 2
## 21   2010/10        6.830487     0    9 3
## 22   2010/11        6.829000     0   10 4
## 23   2010/12       10.620583     0   11 5
## 24   2011/01       10.618272     0   12 6
```

```
# View(AppendITS)
```