Supplementary Document

Lamar Hunt III

10/26/2016

The following R code was used to compute everything needed for the analysis, including exploratory plots not presented in the paper that we used to make decisions about the analysis.

Downloading the FARS data

We last downlaoded the data on 10/26/16:

```
dir.create("./data")
```

```
## Warning in dir.create("./data"): './data' already exists
```

```
path <- "./data"
for(i in 2000:2015){
  dir.create(paste(path,"/fars",i,sep=""))
  assign(paste(path,i,"/zip.zip", sep=""),file())

  if((i >= 1975 & i <= 1993) | (i >= 2001 & i <= 2011)){
    file <- paste("ftp://ftp.nhtsa.dot.gov/fars/",i,"/DBF/FARS",i,".zip", sep="")
  }
  if(i >= 1994 & i <= 2000){
    file <- paste("ftp://ftp.nhtsa.dot.gov/fars/",i,"/DBF/FARSDBF",substr(i,3,4),".zip", sep="")
  }
  if(i == 2012){
    file <- paste("ftp://ftp.nhtsa.dot.gov/fars/",i,"/National/DBF/FARS",i,".zip", sep="")
  }
  if(i >= 2013){
    file <- paste("ftp://ftp.nhtsa.dot.gov/fars/",i,"/National/FARS",i,"NationalDBF.zip", sep="")
  }
  download.file(file, paste(path,"/fars",i,"/zip.zip", sep=""))
  unzip(paste(path,"/fars",i,"/zip.zip", sep=""),exdir=paste(path,"/fars",i,sep=""))
}
```

```
## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2000'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2001'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2002'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2003'
## already exists
```

```
## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2004'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2005'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2006'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2007'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2008'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2009'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2010'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2011'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2012'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2013'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2014'
## already exists

## Warning in dir.create(paste(path, "/fars", i, sep = "")): './data/fars2015'
## already exists
```

```
DownloadTime <- Sys.time()
```

Reading in the data

```
library(foreign)
library(plyr)
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 object is masked from 'package:plyr':
##
##     here

## The following object is masked from 'package:base':
##
##     date
```

```
path <- "./data"
for(i in 2000:2015){
   #person file
    if(i <= 1982 | i >= 1994){
        assign(paste0("person",i),read.dbf(paste0(path, "/fars",i,"/person.dbf")))
    }
    else{
        assign(paste0("person",i),read.dbf(paste0(path, "/fars",i,"/per",i,".dbf")))
    }
}
```

We then concatenate all the person data frames into one giant data frame:

```
person <- data.frame()
for(i in 2000:2015){
    text0 <- paste0("person",i,"$YEAR <- rep(",i,", dim(person",i,")[1])")
    eval(parse(text=text0))

    text <- paste0("person <- rbind.fill(person, person",i,")")
    eval(parse(text=text))
}
```

Cleaning the Data

Next, we subset the data to only get data about the drivers and then clean it. The variable "ALC_RES" was coded differently in years prior to 2015, so we corrected this. We also identified missing values and coded them properly in R. Finally, we defined the new variables "drunk" and "DATE". The variable "drunk" indicated whether a driver was deemed drunk by a police officer (indicated by the variable "DRINKING") OR had a blood alcohol content measured above 0.08 (indicated by the variable "ALC_RES"), and the variable "DATE" simply specifies the date of the accident:

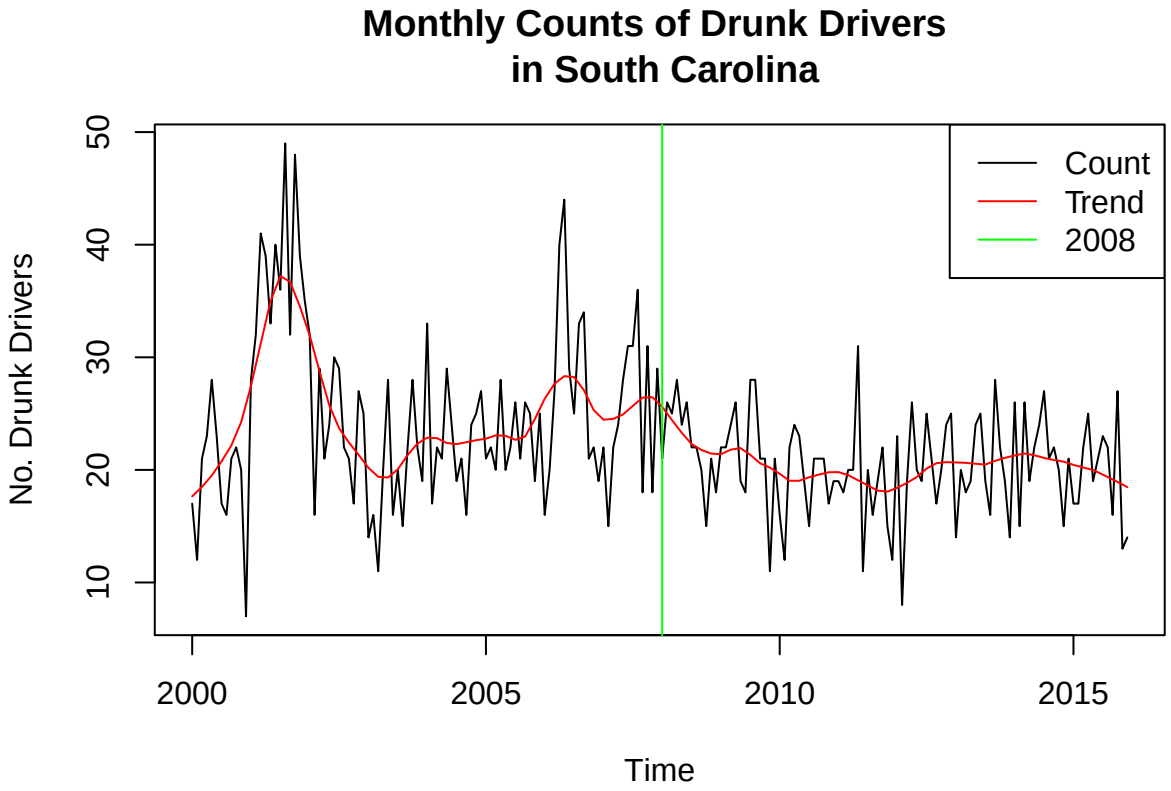
```
# get only info about drivers
drivers <- person %>% filter(PER_TYP==1)

# get ALC_RES for years prior to 2015 to agree with 2015 values
drivers$ALC_RES[drivers$YEAR < 2015] <- 10*drivers$ALC_RES[drivers$YEAR < 2015]

# add a variable indicating whether the driver was drunk
# drunk driving in south carolina is defined as BAC > .08
drivers <- dplyr::mutate(drivers, drunk=(DRINKING==1 | (ALC_RES >= 800 & ALC_RES <= 940)))

# missing data
#drunk
drivers$drunk <- ifelse((drivers$ALC_RES>940 & drivers$DRINKING %in% c(8,9)),
                        NA,drivers$drunk)
#day
drivers$DAY <- ifelse(drivers$DAY==99,
                      NA,drivers$DAY)

# compute dates from year and month variables
drivers$DATE <- ymd(paste(drivers$YEAR, drivers$MONTH, drivers$DAY, sep="-"))
```

```
## Warning: 5 failed to parse.
```

Exploratory plots

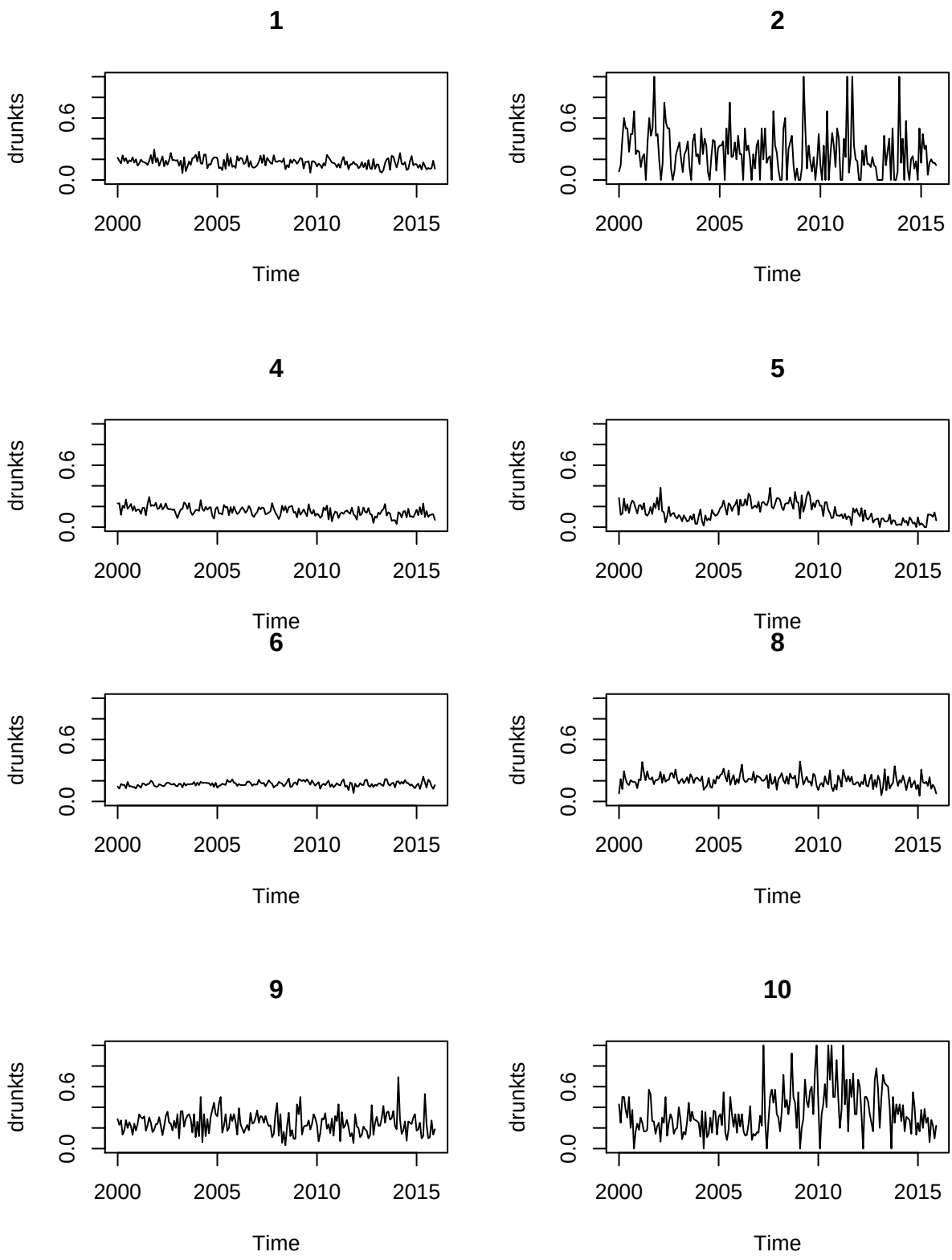
Now, we plot counts of drunk driving for South Carolina from 2000 to 2015. We use this plot in the document to determine the time frame to use:

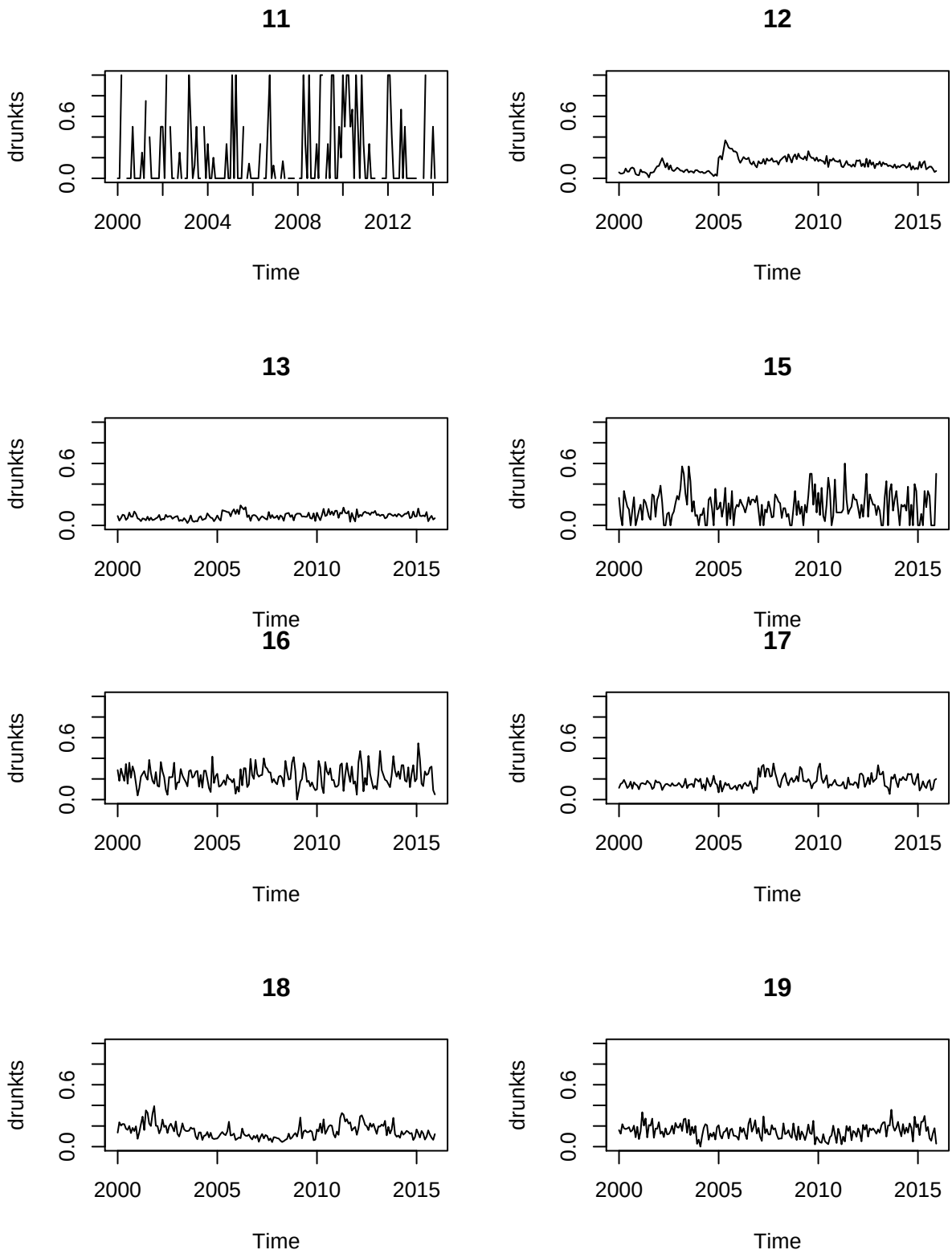
```
# plot monthly dd counts for all USA per month
MonthlyDrunk <- drivers %>%
    group_by(YEAR, MONTH) %>% filter(STATE==45) %>%
    summarise(drunks=sum(drunk, na.rm=T))
southcarolina.drunk <- ts(MonthlyDrunk$drunks, frequency=12, start=c(2000,1))
plot(southcarolina.drunk, main="Monthly Counts of Drunk Drivers \n in South Carolina",
     ylab="No. Drunk Drivers")
trend <- stl(southcarolina.drunk, s.window="periodic")
lines(trend$time.series[,2], col="red")
abline(v=2008, col="green")
legend("topright", col=c("black","red", "green"), lty=1, legend=c("Count", "Trend","2008"))
```

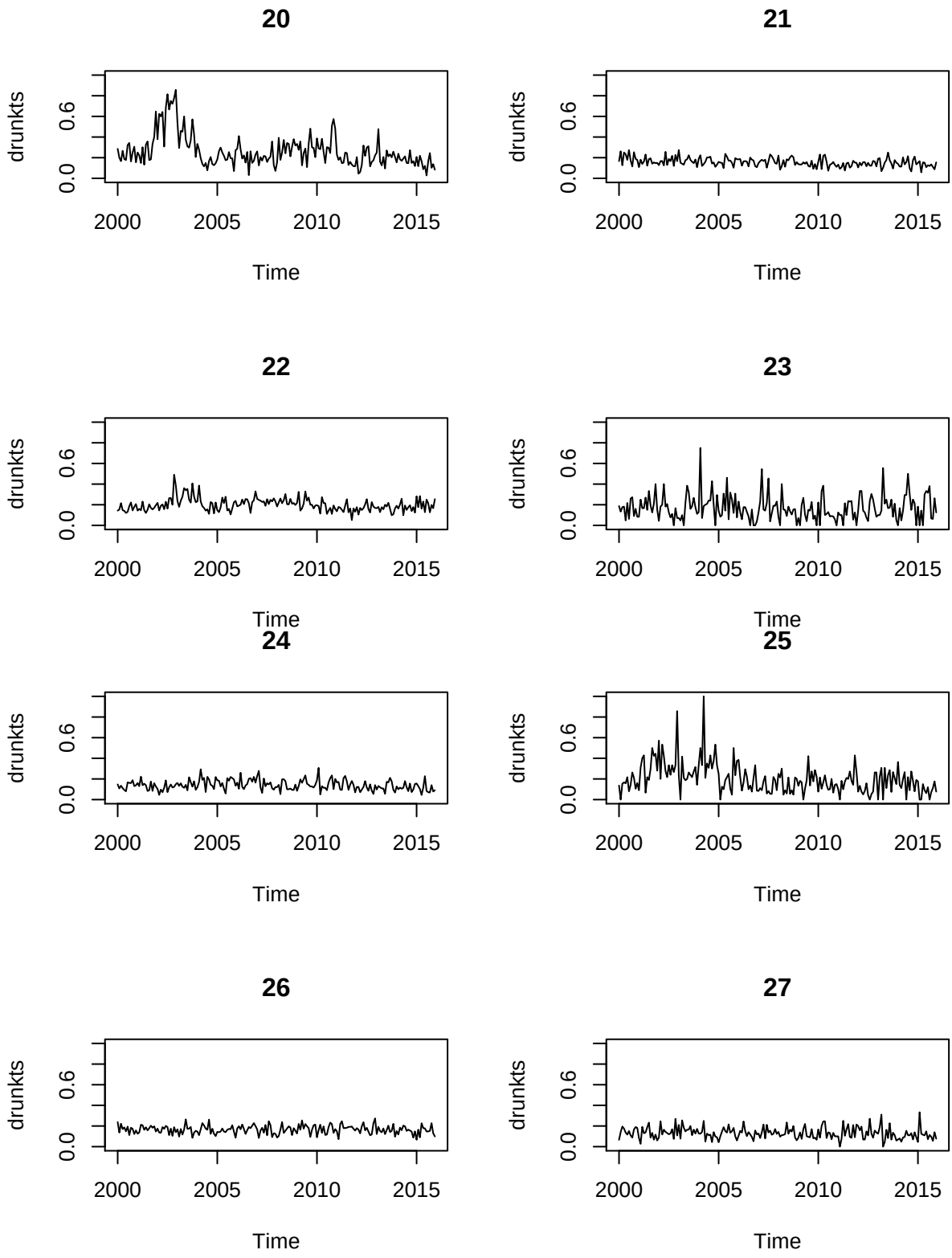


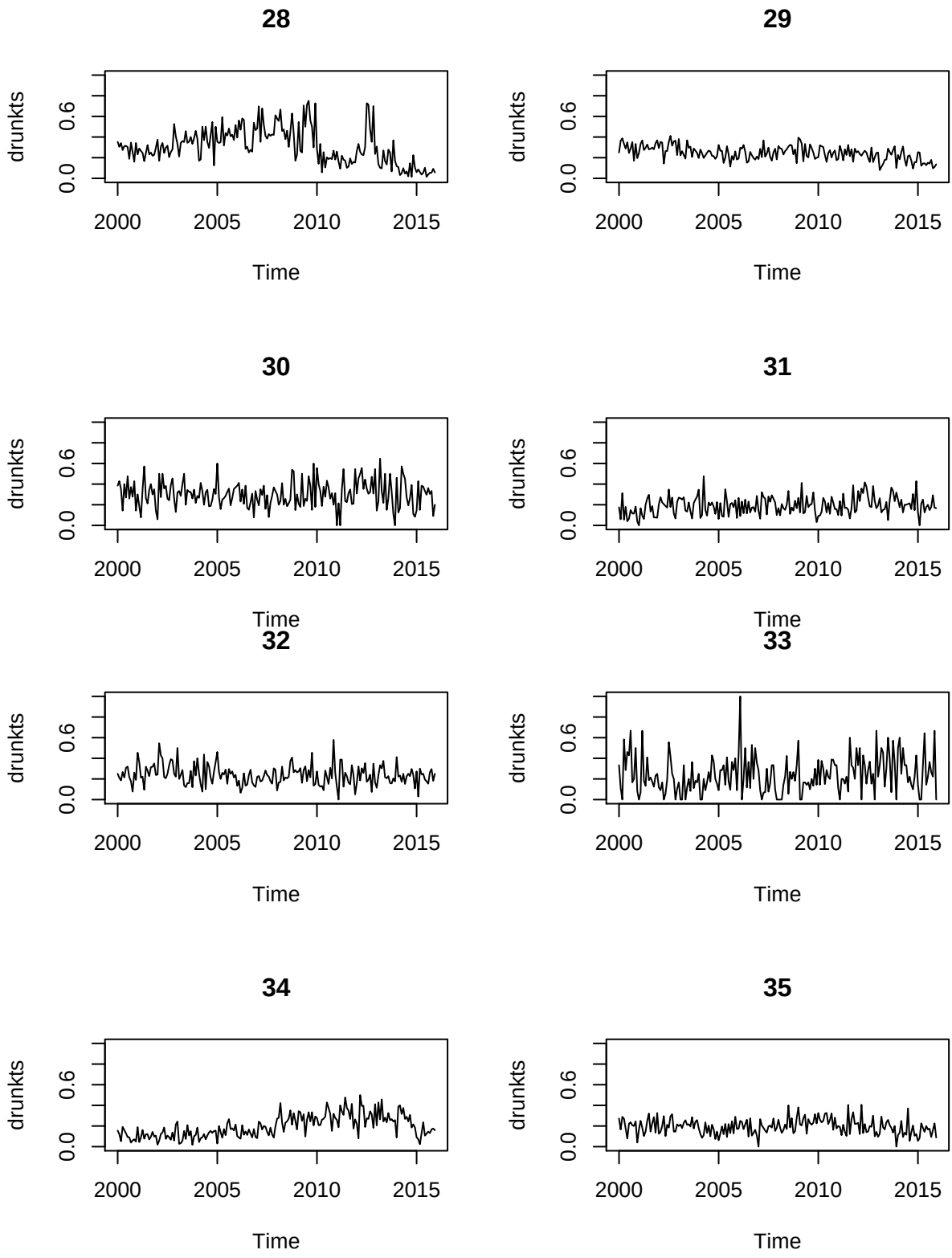
We need to find a state similar to South Carolina in terms of the proportions of drivers involved in fatalities who were drunk. So here we plot every single state's proportion of drivers who were drunk for each month from 2008 to 2015. We find that Texas (state 48) is similar looking to South Caroline (state 45):

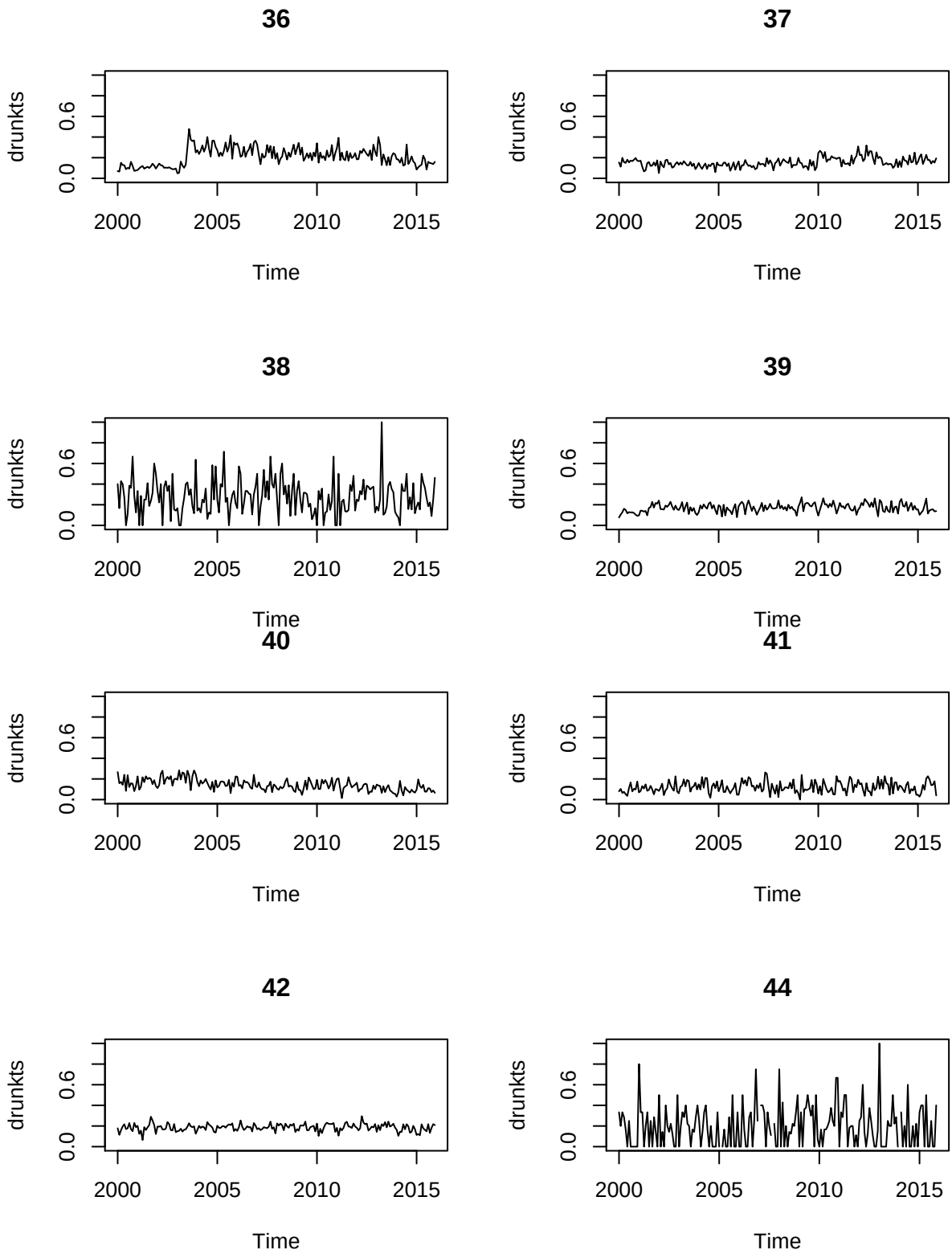
```
par(mfrow=c(2,2))
for(i in unique(drivers$STATE)){
    if(i==45){title <- "South Carolina"}
    else if(i==48){title <- "Texas"}
    else{title <- paste0(i)}

    drunk.props <- drivers %>%
                   group_by(STATE, YEAR, MONTH) %>%
                   filter(STATE==i) %>%
                   summarize(prop=mean(drunk, na.rm=T))
    drunkts <- ts(drunk.props$prop, frequency=12, start=c(2000,1))
    plot(drunkts, ylim=c(0,1), main=title)
}
```

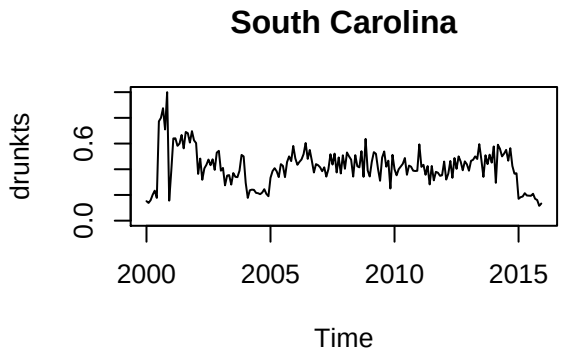

1
2
4
5
6
8
9
10
drunkts
0.0
0.6
2000
2005
2010
2015
Time

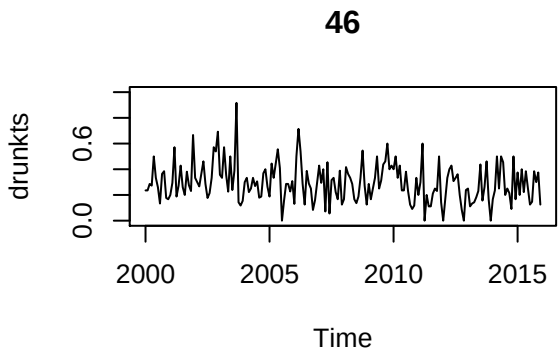

11
12
13
15
16
17
18
19
drunkts
0.0
0.6
2000
2004
2008
2012
2005
2010
2015
Time

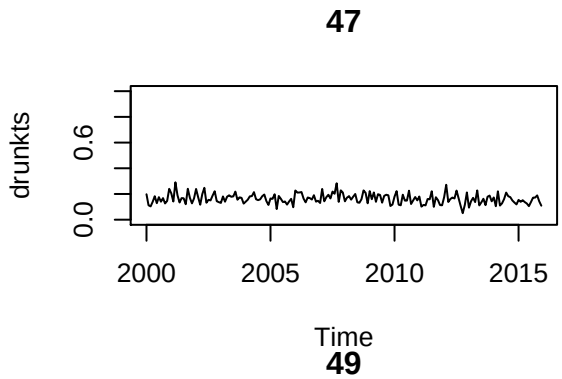

20
21
22
23
24
25
26
27
drunkts
0.0
0.6
2000
2005
2010
2015
Time

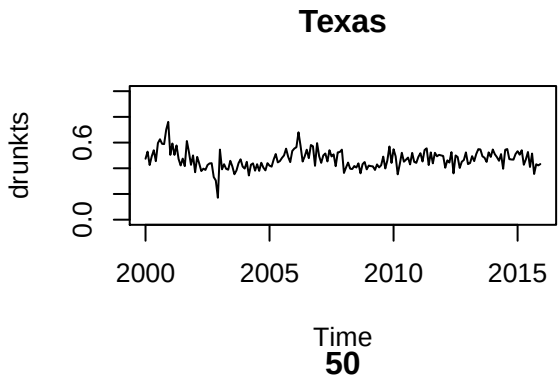

28
29
30
31
32
33
34
35
drunkts
0.0
0.6
2000
2005
2010
2015
Time

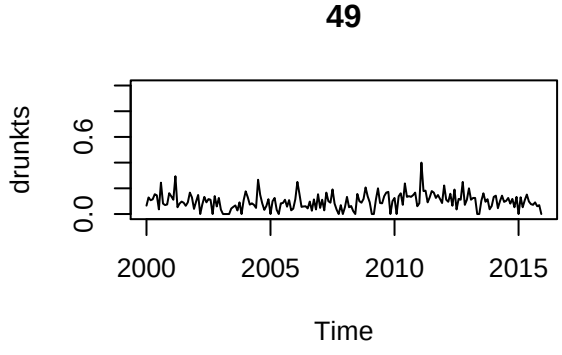

36
37
38
39
40
41
42
44
drunkts
0.0
0.6
2000
2005
2010
2015
Time

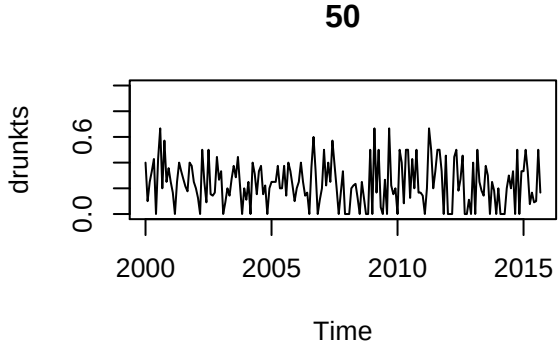

South Carolina
drunkts
0.0
0.6
2000
2005
2010
2015
Time

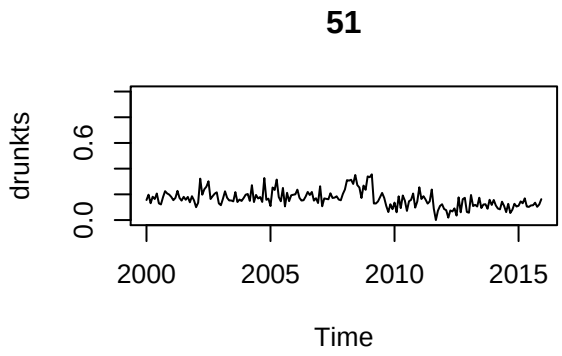

46
drunkts
0.0
0.6
2000
2005
2010
2015
Time

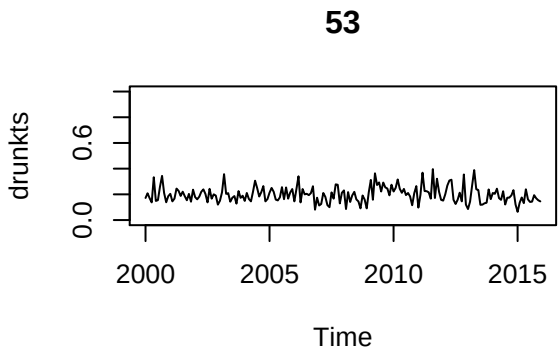

47
drunkts
0.0
0.6
2000
2005
2010
2015
Time

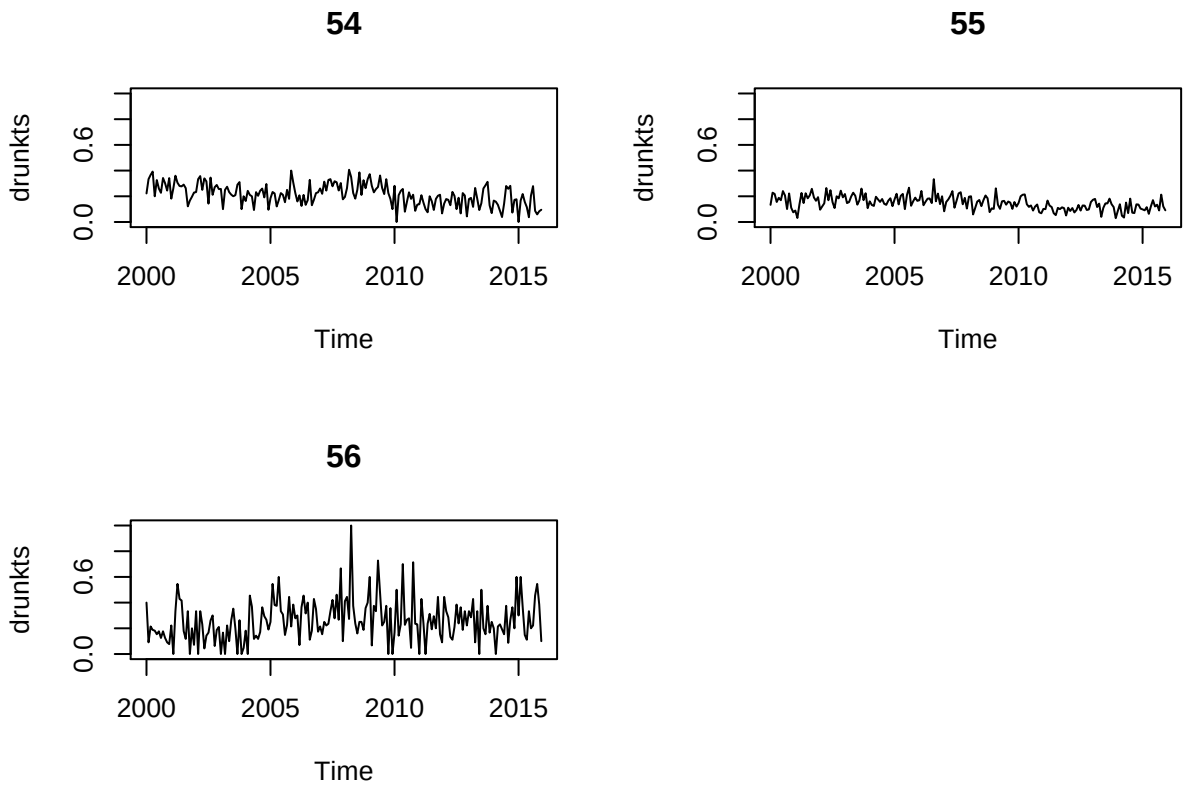

Texas
drunkts
0.0
0.6
2000
2005
2010
2015
Time


49
drunkts
0.0
0.6
2000
2005
2010
2015
Time


50
drunkts
0.0
0.6
2000
2005
2010
2015
Time

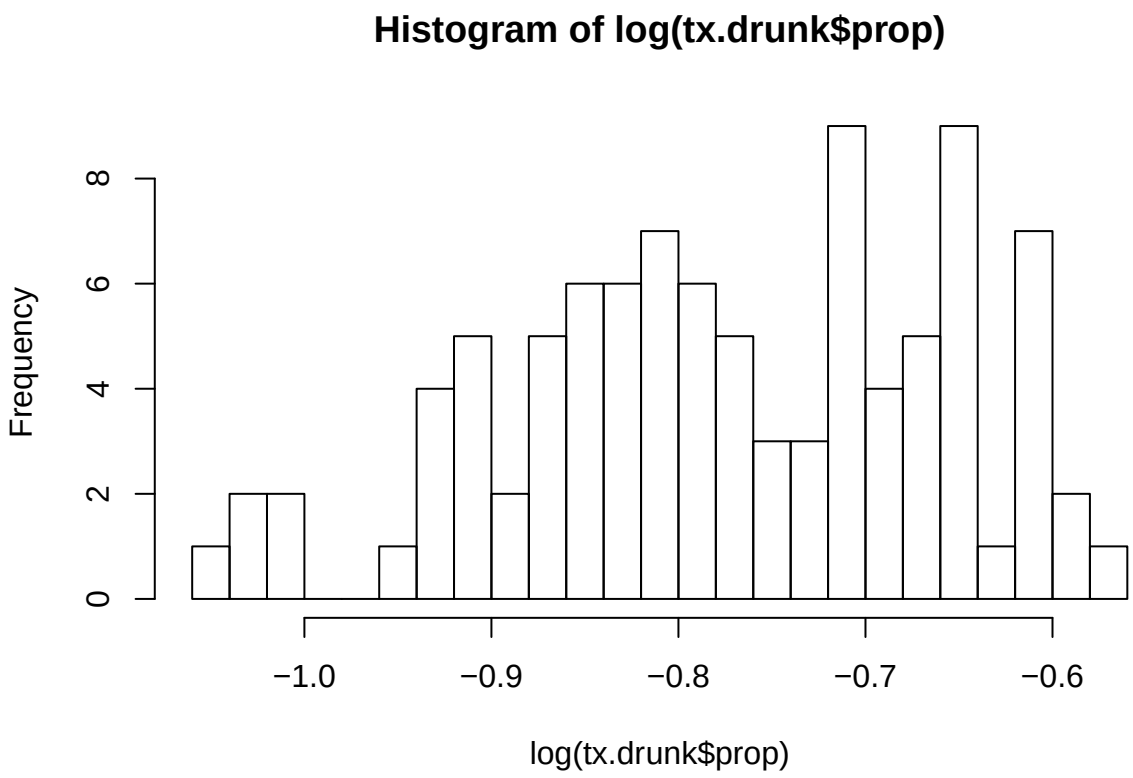

51
drunkts
0.0
0.6
2000
2005
2010
2015
Time


53
drunkts
0.0
0.6
2000
2005
2010
2015
Time

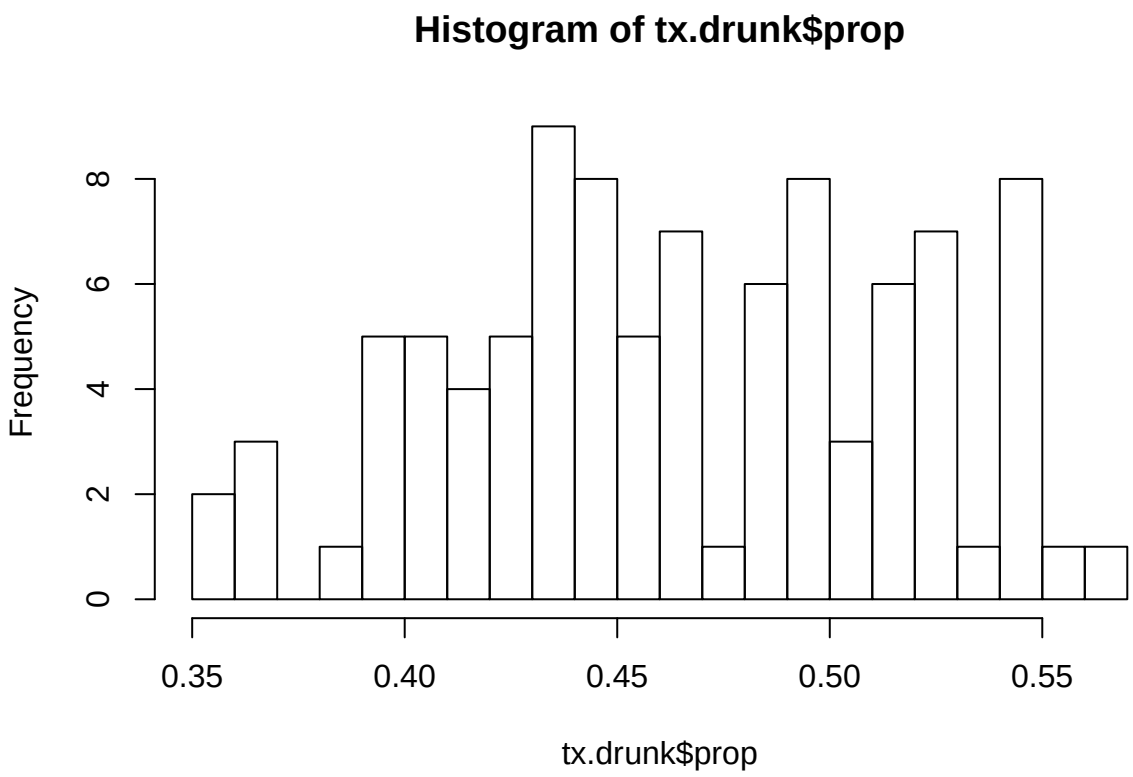


Log transforming the data

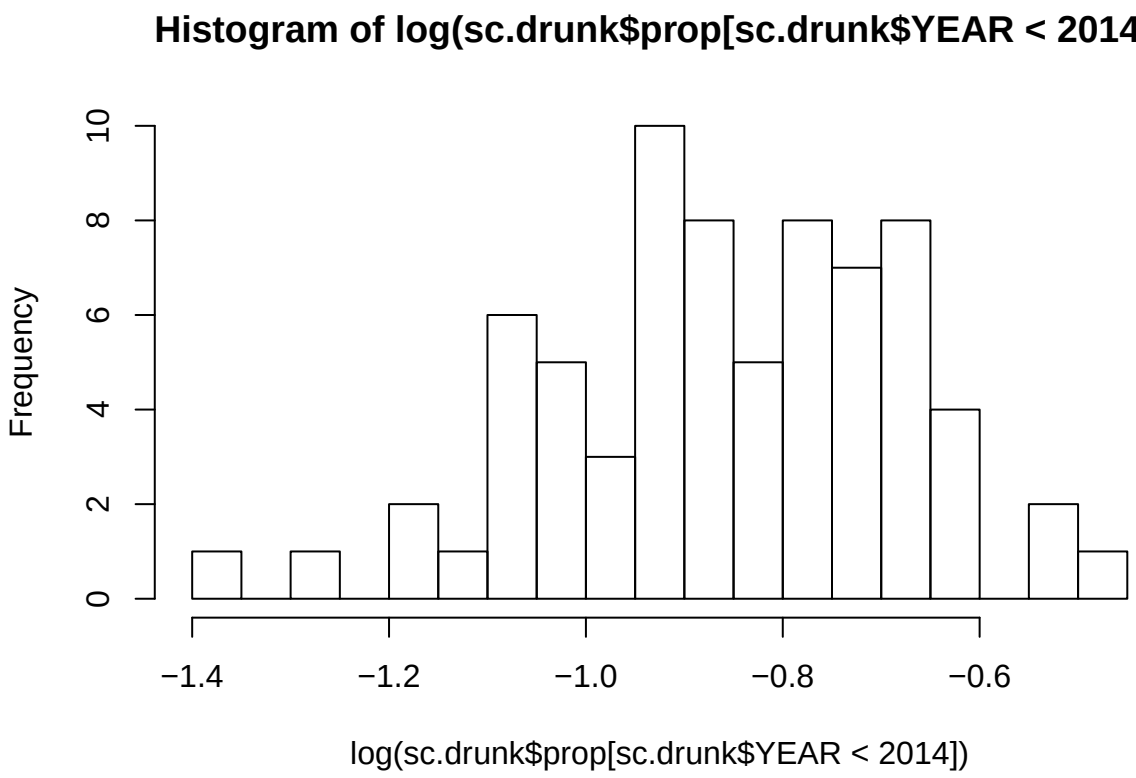
```
# first, we get proportion of drunk drivers per month, per year, per state for 96 months
# let the day be the avg day for that month
sc.drunk <- drivers %>% filter(YEAR >= 2008) %>%
  group_by(STATE, YEAR, MONTH) %>%
  filter(STATE==45) %>%
  summarize(prop=mean(drunk, na.rm=T), DAY=round(mean(DAY)))
tx.drunk <- drivers %>% filter(YEAR >= 2008) %>%
  group_by(STATE, YEAR, MONTH) %>%
  filter(STATE==48) %>%
  summarize(prop=mean(drunk, na.rm=T), DAY=round(mean(DAY)))

# add a date variable
sc.drunk$DATE <- ymd(paste(sc.drunk$YEAR, sc.drunk$MONTH, sc.drunk$DAY, sep="-"))
sc.drunk$DATE_reg <- sc.drunk$DATE - ymd("2008-1-1")

tx.drunk$DATE <- ymd(paste(tx.drunk$YEAR, tx.drunk$MONTH, tx.drunk$DAY, sep="-"))
tx.drunk$DATE_reg <- tx.drunk$DATE - ymd("2008-1-1")

# Now, we plot histograms of the data
hist(log(tx.drunk$prop), breaks=20)
```



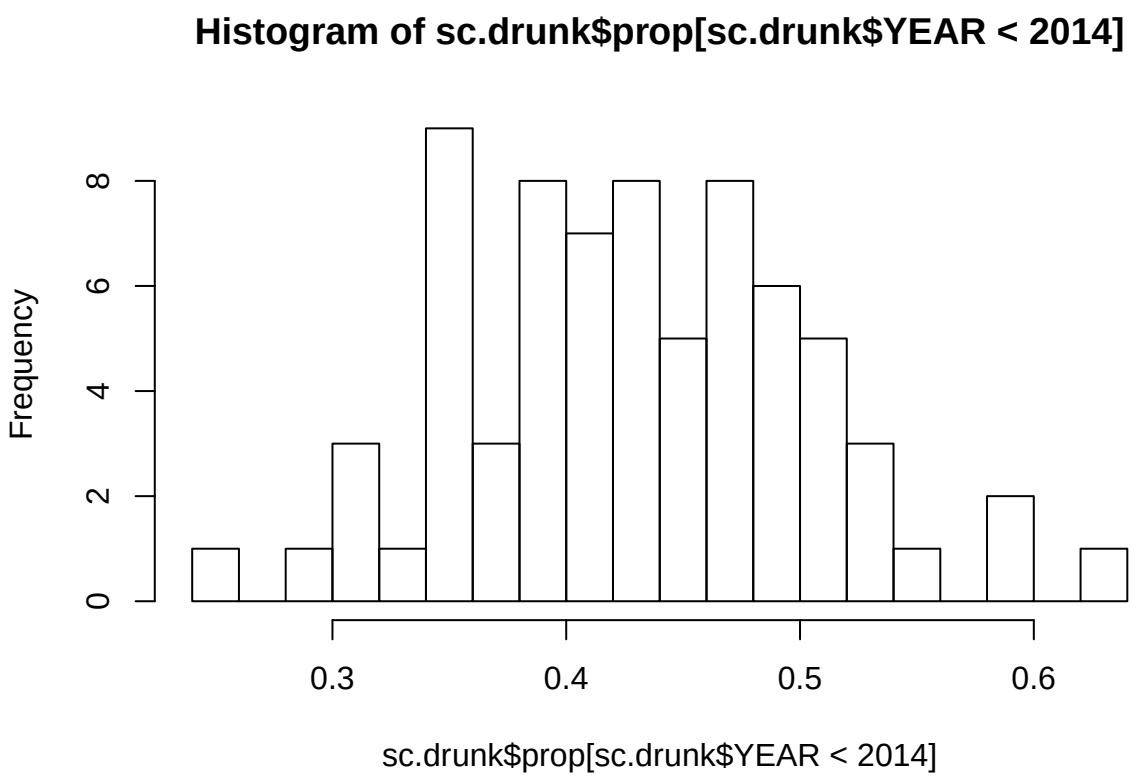

```
hist(tx.drunk$prop, breaks=20)
```



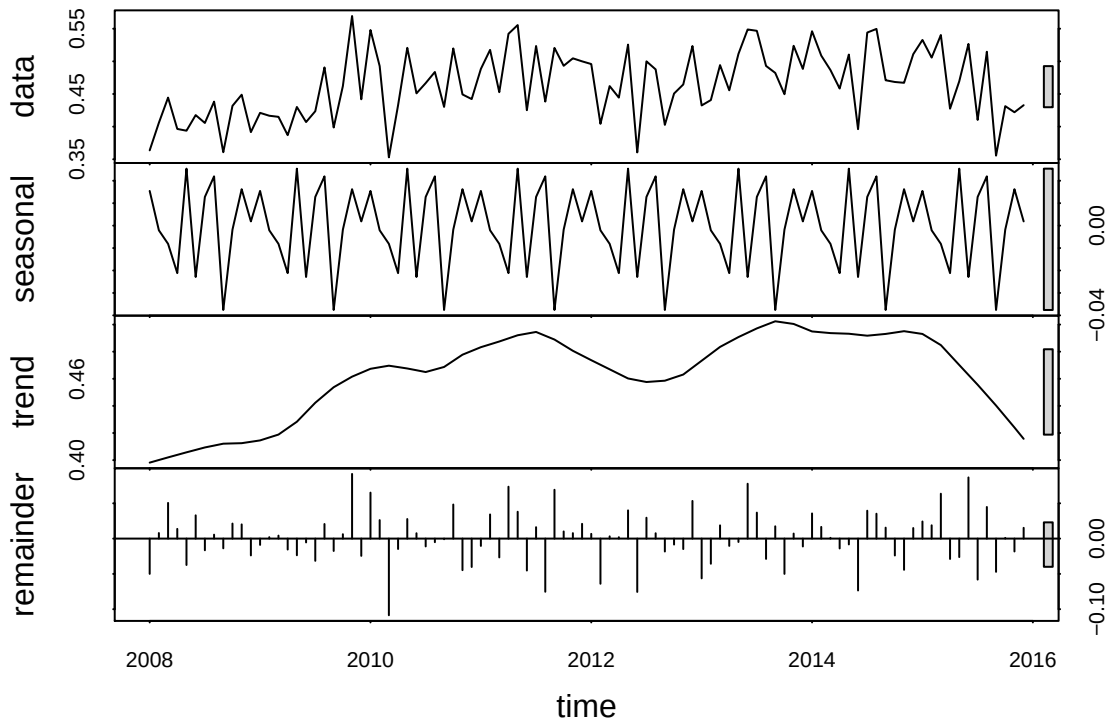
```
hist(log(sc.drunk$prop[sc.drunk$YEAR < 2014]), breaks=20)
```



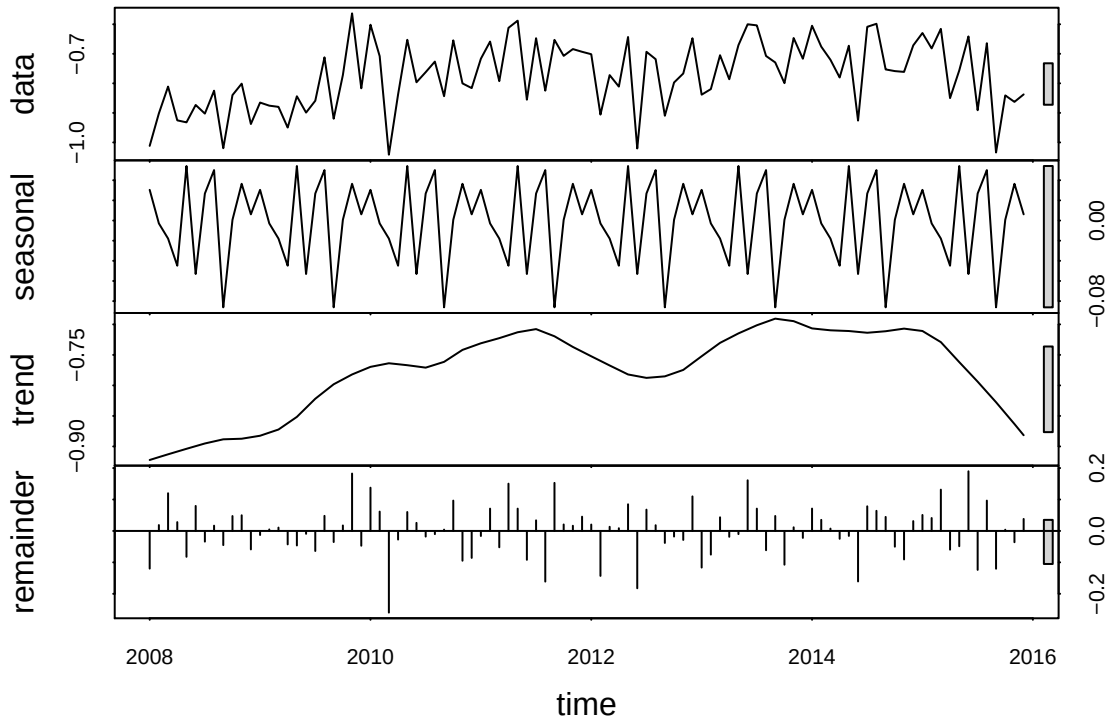

```
hist(sc.drunk$prop[sc.drunk$YEAR < 2014], breaks=20)
```



```r
# examine seasonal trends of the logged and non logged data
sc.ts <- ts(sc.drunk$prop, frequency = 12, start=c(2008,1))
sc.ts.log <- ts(log(sc.drunk$prop), frequency = 12, start=c(2008,1))
tx.ts <- ts(tx.drunk$prop, frequency = 12, start=c(2008,1))
tx.ts.log <- ts(log(tx.drunk$prop), frequency = 12, start=c(2008,1))
plot(stl(tx.ts, s.window="periodic"))
```

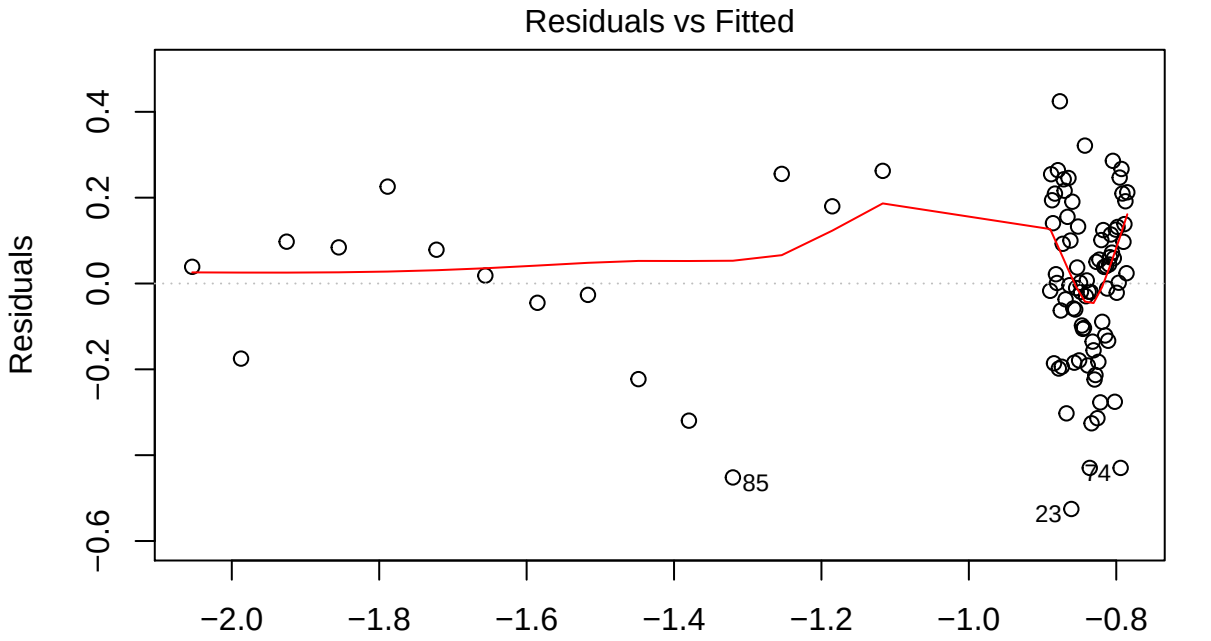


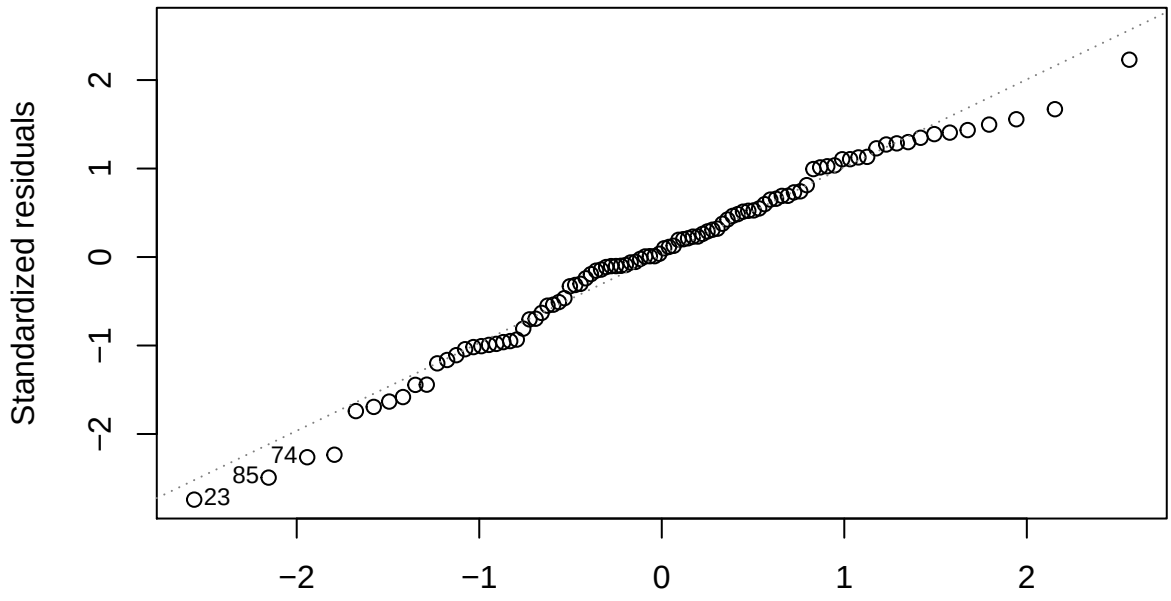

```r
plot(stl(tx.ts.log, s.window="periodic"))
```

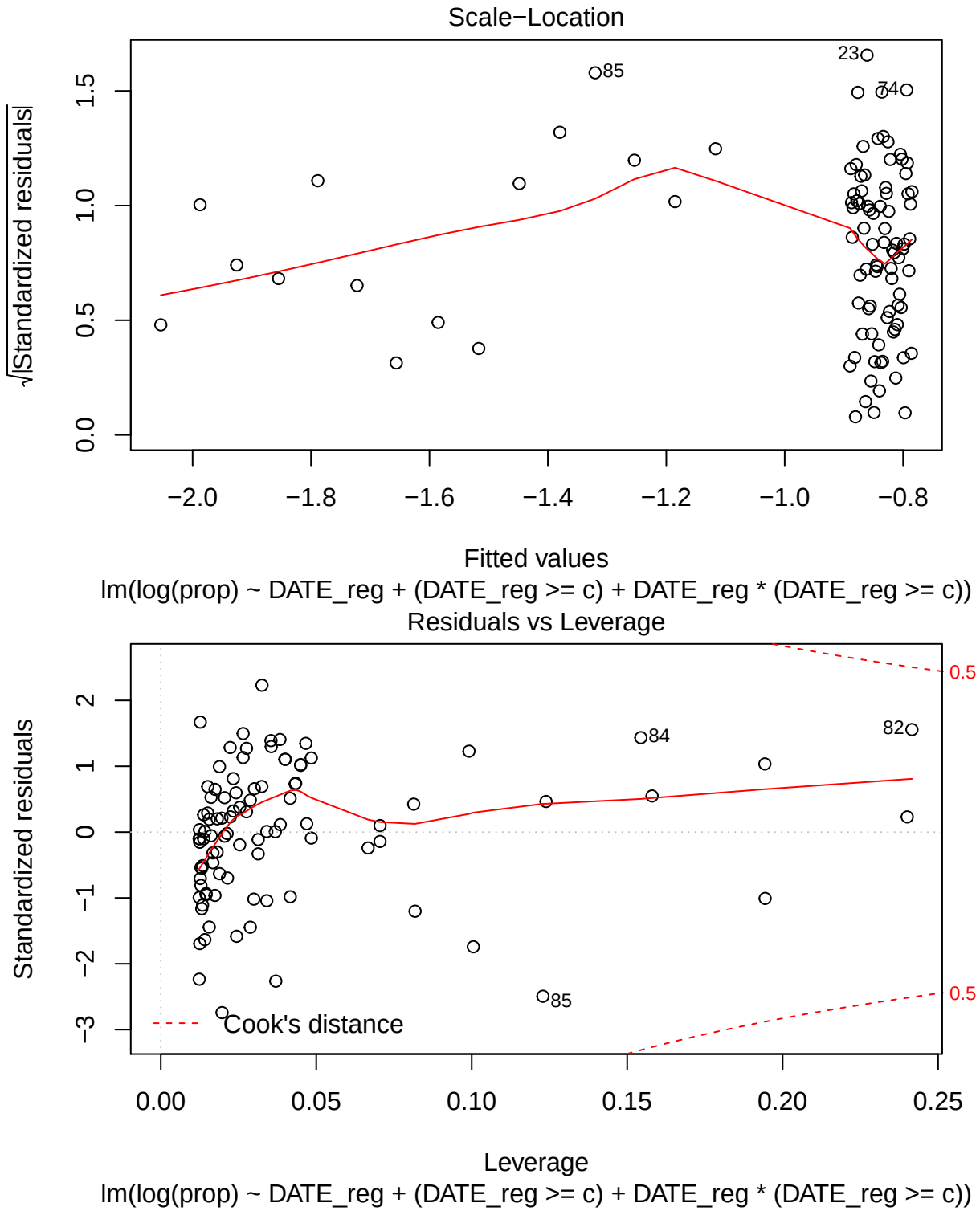


Modeling the data

```
# fit a linear interrupted time series using, but set intercept at Jan 1st, 2008
c <- ymd("2014-10-1") - ymd("2008-1-1")
model.sc <- lm(log(prop) ~ DATE_reg + (DATE_reg >= c) + DATE_reg*(DATE_reg >= c),
               data=sc.drunk)
model.tx <- lm(log(prop) ~ DATE_reg + (DATE_reg >= c) + DATE_reg*(DATE_reg >= c),
               data=tx.drunk)

plot(model.sc)
```


Residuals vs Fitted
Residuals
0.4
0.2
0.0
−0.2
−0.6
−2.0
−1.8
−1.6
−1.4
−1.2
−1.0
−0.8
85
74
23
Fitted values
lm(log(prop) ~ DATE_reg + (DATE_reg >= c) + DATE_reg * (DATE_reg >= c))


Normal Q−Q
Standardized residuals
2
1
0
−1
−2
−2
−1
0
1
2
74
85
23
Theoretical Quantiles
lm(log(prop) ~ DATE_reg + (DATE_reg >= c) + DATE_reg * (DATE_reg >= c))




```
# plot the data with the fitted line
# southcaroline
par(mfrow=c(1,1))
plot(sc.drunk$DATE, log(sc.drunk$prop), type="l",
     ylab="Log(Proportion)",
     xlab="Time",
```

```
     main="South Carolina vs. Texas",
     ylim=c(-2.3,-0.5),
     lwd=2)
lines(sc.drunk$DATE, model.sc$fitted.values, col="red", lwd=2)

# texas
points(tx.drunk$DATE, log(tx.drunk$prop), type="l",
       col="darkorange",lwd=2)
lines(tx.drunk$DATE, model.tx$fitted.values, col="blue",lwd=2)
legend("bottomleft", col=c("black","red","darkorange","blue"), lty=1, lwd=3,
       legend=c("SC","SC fitted","TX","TX fitted"))
```