GHB final 2020

Sulyok

July 19, 2020

#setwd("~/Rprojects/GHB")  
Sys.setenv(LANG = "en")  
sessionInfo()

## R version 4.0.2 (2020-06-22)  
## Platform: x86\_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 16.04.6 LTS  
##   
## Matrix products: default  
## BLAS: /usr/lib/atlas-base/atlas/libblas.so.3.0  
## LAPACK: /usr/lib/atlas-base/atlas/liblapack.so.3.0  
##   
## locale:  
## [1] LC\_CTYPE=C.UTF-8 LC\_NUMERIC=C LC\_TIME=C.UTF-8   
## [4] LC\_COLLATE=C.UTF-8 LC\_MONETARY=C.UTF-8 LC\_MESSAGES=C.UTF-8   
## [7] LC\_PAPER=C.UTF-8 LC\_NAME=C LC\_ADDRESS=C   
## [10] LC\_TELEPHONE=C LC\_MEASUREMENT=C.UTF-8 LC\_IDENTIFICATION=C   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## loaded via a namespace (and not attached):  
## [1] compiler\_4.0.2 magrittr\_1.5 tools\_4.0.2 htmltools\_0.5.0  
## [5] yaml\_2.2.1 stringi\_1.4.6 rmarkdown\_2.3 knitr\_1.29   
## [9] stringr\_1.4.0 xfun\_0.16 digest\_0.6.25 rlang\_0.4.7   
## [13] evaluate\_0.14

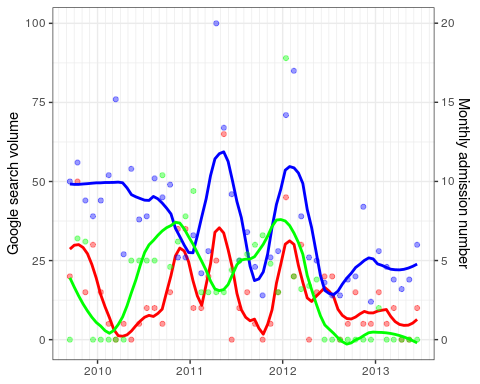
library(haven)  
data <- read\_sav("GHB\_tox\_adat\_140711.sav")  
GTD <- read\_sav("GHB\_tox\_google.sav")  
#GTD<-GTD[,-c(6:9)]  
#GTD<-na.omit(GTD)  
GTD<-GTD[-47,]  
counts<-table(data$FELVETEL)  
counts<-as.data.frame(counts)  
  
  
GTD$date <- seq(as.Date("2009-09-14"), as.Date("2013-07-13"), "month" )  
  
library(ggplot2)  
  
  
p<-ggplot(GTD, aes(x = date)) +  
geom\_point(aes(y=tox\_eset\*5, colour="Admission numbers"), alpha= 0.4) + geom\_smooth(aes(y=tox\_eset\*5, colour="Admission numbers"), span = 0.2, se=FALSE) +  
geom\_point(aes(y=google1, col="GHB-related seach volumes"), alpha= 0.4) + geom\_smooth(aes(y=google1, col="GHB-related seach volumes"), span = 0.3, se=FALSE) +  
 geom\_point(aes(y=google2, col="Gina-related seach volumes"), alpha= 0.4) + geom\_smooth(aes(y=google2, col="Gina-related seach volumes"), span = 0.3, se=FALSE) +   
 scale\_colour\_manual("", breaks = c("Admission numbers", "GHB-related seach volumes","Gina-related seach volumes") , values = c("red", "blue", "green")) +  
scale\_x\_date(date\_minor\_breaks = "1 month") +  
xlab(NULL) +  
ylab("Google search volume") + ylim(c(0, 100)) +  
 scale\_y\_continuous(sec.axis = sec\_axis(~./5, name = "Monthly admission number")) +  
theme\_bw() + theme(legend.position = "none")

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

p

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'  
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



cor.test(GTD$tox\_eset, GTD$google1, method="kendall")

## Warning in cor.test.default(GTD$tox\_eset, GTD$google1, method = "kendall"):  
## Cannot compute exact p-value with ties

##   
## Kendall's rank correlation tau  
##   
## data: GTD$tox\_eset and GTD$google1  
## z = 2.058, p-value = 0.03959  
## alternative hypothesis: true tau is not equal to 0  
## sample estimates:  
## tau   
## 0.2238888

NSM3::kendall.ci(GTD$tox\_eset, GTD$google1)

## fANCOVA 0.5-1 loaded

##   
## 1 - alpha = 0.95 two-sided CI for tau:  
## -0.004, 0.452

cor.test(GTD$tox\_eset, GTD$google2, method="kendall")

## Warning in cor.test.default(GTD$tox\_eset, GTD$google2, method = "kendall"):  
## Cannot compute exact p-value with ties

##   
## Kendall's rank correlation tau  
##   
## data: GTD$tox\_eset and GTD$google2  
## z = 1.2315, p-value = 0.2181  
## alternative hypothesis: true tau is not equal to 0  
## sample estimates:  
## tau   
## 0.1407854

NSM3::kendall.ci(GTD$tox\_eset, GTD$google2)

##   
## 1 - alpha = 0.95 two-sided CI for tau:  
## -0.063, 0.345

library(lubridate)

##   
## Attaching package: 'lubridate'

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

counts$Var1 <- ymd(counts$Var1)  
  
# extract week year  
  
week <- function(x)format(x, '%Y.%W')  
  
counts$weekyear<-week(counts$Var1)  
  
month<- function(x)format(x, '%Y.%B')  
  
counts$month<-month(counts$Var1)  
#create a df with 0-counts for all days  
  
n<-rep(0, 1369)  
n<-as.data.frame(n)  
n$Var1<-seq(as.Date("2009-09-14"), as.Date("2013-06-13"), "day" )  
  
#join them  
library(dplyr)

##   
## Attaching package: 'dplyr'

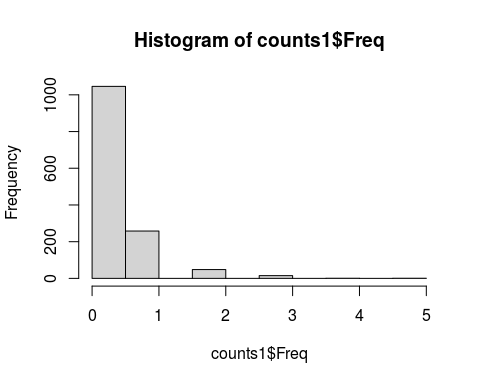
## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

counts<-full\_join(n, counts, by="Var1")  
counts$Freq[is.na(counts$Freq)] <- 0   
  
# extract week year  
  
week <- function(x)format(x, '%Y.%W')  
  
counts$weekyear<-week(counts$Var1)  
summary(counts)

## n Var1 Freq weekyear   
## Min. :0 Min. :2009-09-14 Min. :0.000 Length:1369   
## 1st Qu.:0 1st Qu.:2010-08-22 1st Qu.:0.000 Class :character   
## Median :0 Median :2011-07-30 Median :0.000 Mode :character   
## Mean :0 Mean :2011-07-30 Mean :0.298   
## 3rd Qu.:0 3rd Qu.:2012-07-06 3rd Qu.:0.000   
## Max. :0 Max. :2013-06-13 Max. :5.000   
## month   
## Length:1369   
## Class :character   
## Mode :character   
##   
##   
##

counts$month<-month(counts$Var1)  
  
  
  
# extract week year  
  
GTD$weekyear<-week(GTD$date)  
GTD$month<-month(GTD$date)  
  
#join  
  
library(dplyr)  
counts1<-full\_join(counts, GTD, by="month")  
hist(counts1$Freq)



mean(counts1$Freq)

## [1] 0.2980278

var(counts1$Freq)

## [1] 0.3687169

counts$date<-NULL  
counts1$n<-NULL  
summary(counts1)

## Var1 Freq weekyear.x month   
## Min. :2009-09-14 Min. :0.000 Length:1369 Length:1369   
## 1st Qu.:2010-08-22 1st Qu.:0.000 Class :character Class :character   
## Median :2011-07-30 Median :0.000 Mode :character Mode :character   
## Mean :2011-07-30 Mean :0.298   
## 3rd Qu.:2012-07-06 3rd Qu.:0.000   
## Max. :2013-06-13 Max. :5.000   
##   
## datum tox\_eset tox\_sz\_eset google1   
## Length:1369 Min. : 0.000 Min. :0.000 Min. : 12.00   
## Class :character 1st Qu.: 1.000 1st Qu.:0.000 1st Qu.: 23.00   
## Mode :character Median : 2.000 Median :1.000 Median : 28.00   
## Mean : 2.866 Mean :1.897 Mean : 36.46   
## 3rd Qu.: 4.000 3rd Qu.:3.000 3rd Qu.: 46.00   
## Max. :13.000 Max. :8.000 Max. :100.00   
## NA's :30   
## google2 google1\_csusz google2\_csusz google1\_2csusz   
## Min. : 0.00 Min. : 12.00 Min. : 0.00 Min. : 12.00   
## 1st Qu.: 0.00 1st Qu.: 23.00 1st Qu.: 0.00 1st Qu.: 23.00   
## Median :15.00 Median : 28.00 Median :15.00 Median : 33.00   
## Mean :17.01 Mean : 36.86 Mean :17.04 Mean : 37.45   
## 3rd Qu.:25.00 3rd Qu.: 46.00 3rd Qu.:25.00 3rd Qu.: 49.00   
## Max. :89.00 Max. :100.00 Max. :89.00 Max. :100.00   
## NA's :17 NA's :17 NA's :48   
## google2\_2csusz date weekyear.y   
## Min. : 0.00 Min. :2009-09-14 Length:1369   
## 1st Qu.: 0.00 1st Qu.:2010-08-14 Class :character   
## Median :16.00 Median :2011-07-14 Mode :character   
## Mean :17.62 Mean :2011-07-28   
## 3rd Qu.:25.00 3rd Qu.:2012-07-14   
## Max. :89.00 Max. :2013-06-14   
## NA's :48

ts <- ts(counts1$Freq, start=c(2009, 9, 14), end=c(2013, 6, 14), frequency=365)   
counts1$weekday<-weekdays(counts1$Var1)  
  
  
kruskal.test(counts1$Freq, as.factor(counts1$weekday))

##   
## Kruskal-Wallis rank sum test  
##   
## data: counts1$Freq and as.factor(counts1$weekday)  
## Kruskal-Wallis chi-squared = 20.621, df = 6, p-value = 0.002145

library(dunn.test)  
dunn.test(counts1$Freq, as.factor(counts1$weekday), method="holm")

## Kruskal-Wallis rank sum test  
##   
## data: x and group  
## Kruskal-Wallis chi-squared = 20.621, df = 6, p-value = 0  
##   
##   
## Comparison of x by group   
## (Holm)   
## Col Mean-|  
## Row Mean | Friday Monday Saturday Sunday Thursday Tuesday  
## ---------+------------------------------------------------------------------  
## Monday | 0.615825  
## | 1.0000  
## |  
## Saturday | -1.200698 -1.818058  
## | 0.9195 0.4834  
## |  
## Sunday | -1.797887 -2.416010 -0.597189  
## | 0.4693 0.1334 1.0000  
## |  
## Thursday | 0.285429 -0.330818 1.487662 2.085615  
## | 0.3877 0.7408 0.7526 0.2961  
## |  
## Tuesday | 1.891698 1.277508 3.093931 3.691884 1.608327  
## | 0.4390 0.9064 0.0188\* 0.0023\* 0.6466  
## |  
## Wednesda | 1.329397 0.714485 2.531629 3.129582 1.045304 -0.563022  
## | 0.9186 1.0000 0.1022 0.0175\* 1.0000 0.8601  
##   
## alpha = 0.05  
## Reject Ho if p <= alpha/2

#crosscorr months  
  
ccfKendall1<- sapply( -12:12, function(l) cor.test( GTD$tox\_eset, Hmisc::Lag(GTD$google1,l),method =  
"kendall", use ="complete.obs")$estimate )

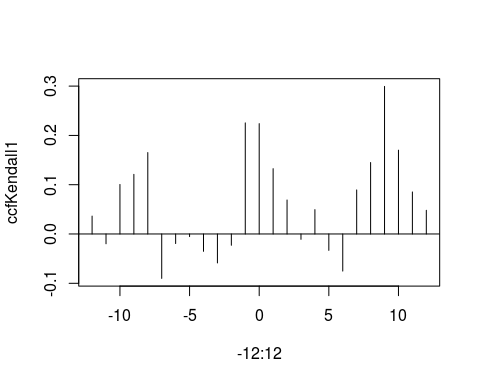
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties

## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google1, l), method =  
## "kendall", : Cannot compute exact p-value with ties

ccfKendall1

## tau tau tau tau tau tau   
## 0.036322034 -0.019818244 0.100512969 0.120989147 0.165190921 -0.090095249   
## tau tau tau tau tau tau   
## -0.019363707 -0.005267546 -0.035190406 -0.058742526 -0.022894552 0.225398852   
## tau tau tau tau tau tau   
## 0.223888816 0.132548329 0.068986848 -0.010838497 0.049411564 -0.033250326   
## tau tau tau tau tau tau   
## -0.075258492 0.089419556 0.144981647 0.299442426 0.170067700 0.085270952   
## tau   
## 0.048054092

plot(-12:12,ccfKendall1,type="h")  
abline(h=0)



ccfKendall2<- sapply( -12:12, function(l) cor.test( GTD$tox\_eset, Hmisc::Lag(GTD$google2,l),method =  
"kendall", use ="complete.obs")$estimate )

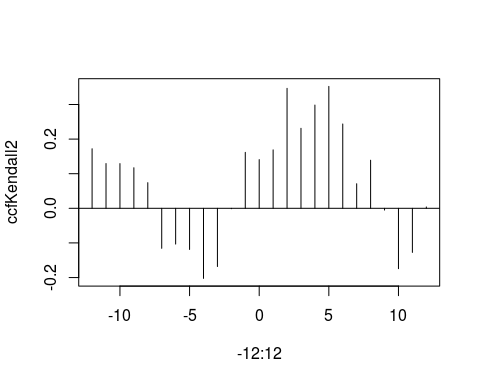
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties

## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties  
  
## Warning in cor.test.default(GTD$tox\_eset, Hmisc::Lag(GTD$google2, l), method =  
## "kendall", : Cannot compute exact p-value with ties

ccfKendall2

## tau tau tau tau tau tau   
## 0.172093340 0.129413752 0.129271414 0.117250667 0.074134942 -0.115567343   
## tau tau tau tau tau tau   
## -0.103329166 -0.118828999 -0.202487709 -0.168138164 -0.001239466 0.161675582   
## tau tau tau tau tau tau   
## 0.140785431 0.168952379 0.346707551 0.231214302 0.298393835 0.352305773   
## tau tau tau tau tau tau   
## 0.243726558 0.071046051 0.138945369 -0.005085476 -0.174410636 -0.127372989   
## tau   
## 0.003953874

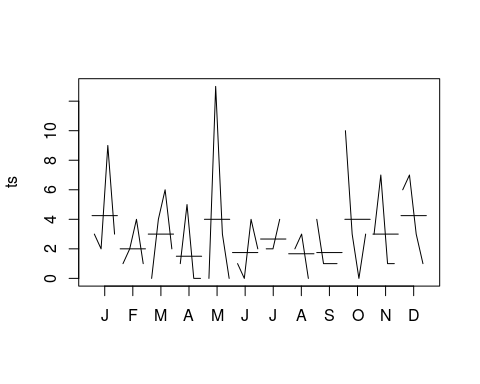
plot(-12:12,ccfKendall2,type="h")  
abline(h=0)



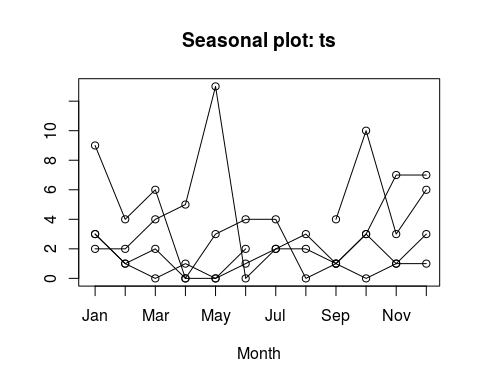
##monthly data  
#setwd("~/Rprojects/GHB")  
library(haven)  
data <- read\_sav("GHB\_tox\_adat\_140711.sav")  
GTD <- read\_sav("GHB\_tox\_google.sav")  
GTD<-GTD[-47,]  
library(forecast)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

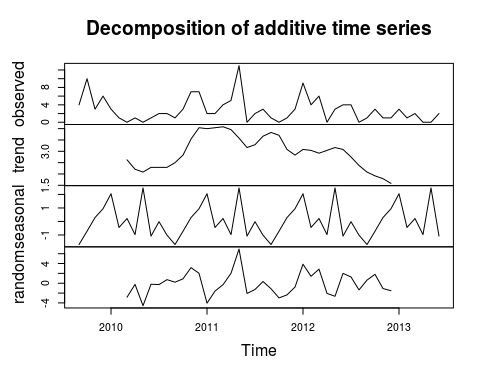
ts <- ts(GTD$tox\_eset, start=c(2009, 9, 14), end=c(2013, 6, 14), frequency=12)   
monthplot(ts)



seasonplot(ts)

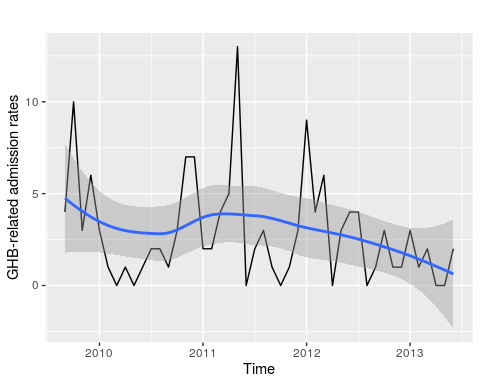


decomp<-decompose(ts)  
plot(decomp)

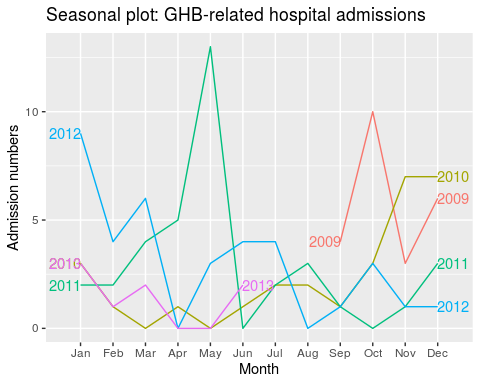


#monthplot(decomp, choice="seasonal")  
autoplot(ts)+ geom\_smooth(method="loess") + ylab("GHB-related admission rates")

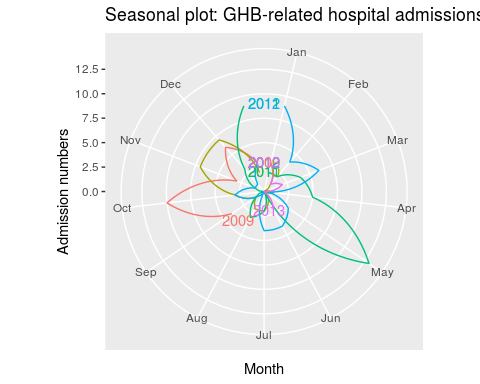
## `geom\_smooth()` using formula 'y ~ x'



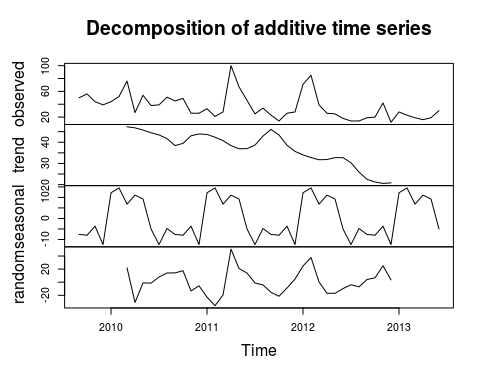
ggseasonplot(ts, year.labels=TRUE, year.labels.left=TRUE) +  
 ylab("Admission numbers") +  
 ggtitle("Seasonal plot: GHB-related hospital admissions")



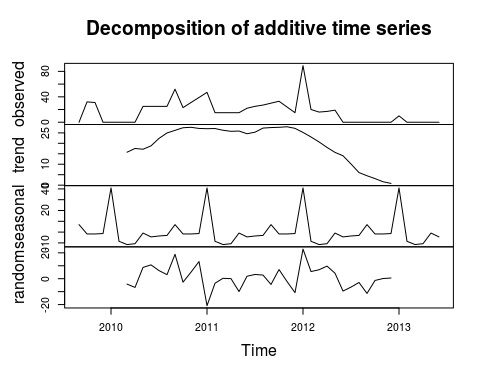
ggseasonplot(ts, year.labels=TRUE, year.labels.left=TRUE, polar=TRUE) +  
 ylab("Admission numbers") +  
 ggtitle("Seasonal plot: GHB-related hospital admissions")



tsg1 <- ts(GTD$google1, start=c(2009, 9, 14), end=c(2013, 6, 14), frequency=12)   
decomp<-decompose(tsg1)  
plot(decomp)



tsg2 <- ts(GTD$google2, start=c(2009, 9, 14), end=c(2013, 6, 14), frequency=12)   
decomp<-decompose(tsg2)  
plot(decomp)



library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

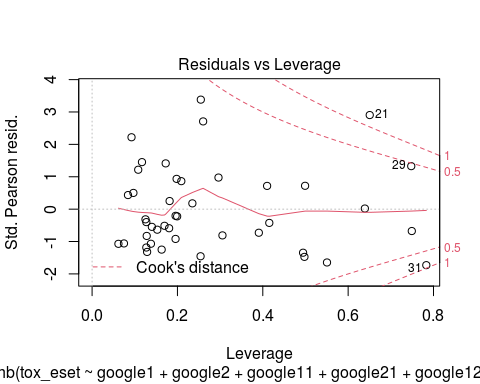
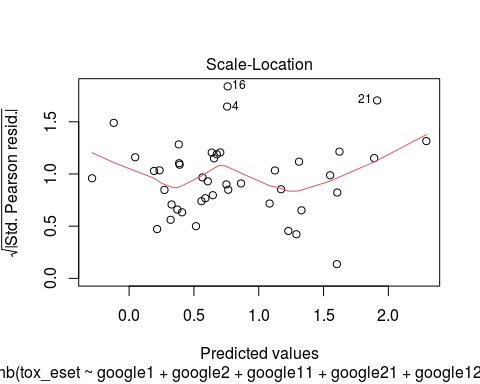
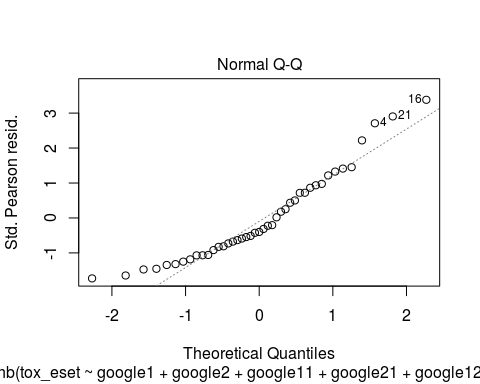
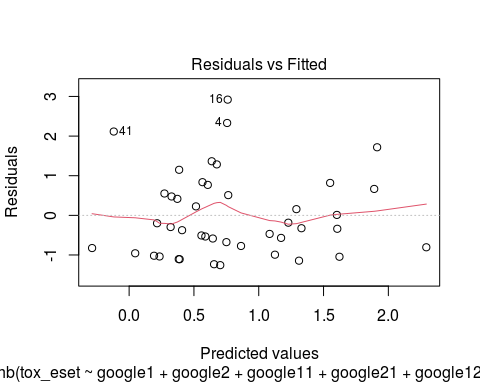
#lag-distributed  
  
GTD$google11 <- lag(GTD$google1,1)  
GTD$google12 <- lag(GTD$google1,2)  
GTD$google13 <- lag(GTD$google1,3)  
  
  
GTD$google21 <- lag(GTD$google2, 1)  
GTD$google22 <- lag(GTD$google2, 2)  
GTD$google23 <- lag(GTD$google2,3)  
  
  
GTD$tox\_eset1 <- lag(GTD$tox\_eset,1)  
GTD$tox\_eset2 <- lag(GTD$tox\_eset,2)  
GTD$tox\_eset3 <- lag(GTD$tox\_eset,3)  
  
  
fit<- glm.nb(tox\_eset ~ google1 + google2 + google11 + google21 + google12 + google22 + google13 + google23 +  
 tox\_eset1 + tox\_eset2 + tox\_eset3, data=GTD)

## Warning in glm.nb(tox\_eset ~ google1 + google2 + google11 + google21 + google12  
## + : alternation limit reached

summary(fit)

##   
## Call:  
## glm.nb(formula = tox\_eset ~ google1 + google2 + google11 + google21 +   
## google12 + google22 + google13 + google23 + tox\_eset1 + tox\_eset2 +   
## tox\_eset3, data = GTD, init.theta = 7.222048562, link = log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8874 -1.0526 -0.3414 0.5604 2.1681   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.173541 0.394321 0.440 0.65987   
## google1 0.009288 0.006938 1.339 0.18063   
## google2 0.007897 0.007034 1.123 0.26160   
## google11 0.009713 0.007661 1.268 0.20486   
## google21 -0.005819 0.007294 -0.798 0.42503   
## google12 -0.021545 0.009065 -2.377 0.01747 \*   
## google22 0.023124 0.007958 2.906 0.00366 \*\*  
## google13 0.002906 0.007975 0.364 0.71559   
## google23 -0.001030 0.009815 -0.105 0.91644   
## tox\_eset1 0.053445 0.058570 0.913 0.36150   
## tox\_eset2 0.008479 0.049912 0.170 0.86510   
## tox\_eset3 0.012386 0.049308 0.251 0.80166   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Negative Binomial(7.222) family taken to be 1)  
##   
## Null deviance: 79.469 on 42 degrees of freedom  
## Residual deviance: 48.889 on 31 degrees of freedom  
## (3 observations deleted due to missingness)  
## AIC: 186.43  
##   
## Number of Fisher Scoring iterations: 1  
##   
##   
## Theta: 7.22   
## Std. Err.: 6.09   
## Warning while fitting theta: alternation limit reached   
##   
## 2 x log-likelihood: -160.427

plot(fit)



estim<-cbind(fit$coefficients, confint(fit))

## Waiting for profiling to be done...

estim

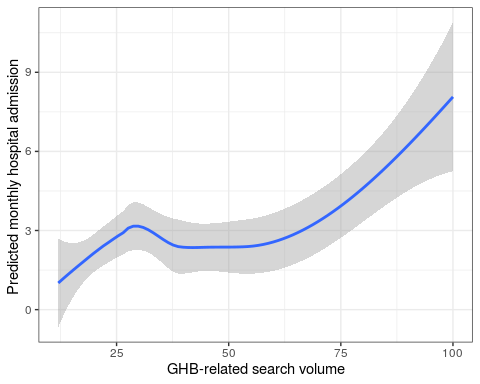
## 2.5 % 97.5 %  
## (Intercept) 0.173540645 -0.631376627 0.932355380  
## google1 0.009288089 -0.004575750 0.022989010  
## google2 0.007896824 -0.005730211 0.021471353  
## google11 0.009713090 -0.004231130 0.023593073  
## google21 -0.005818769 -0.020597435 0.008456166  
## google12 -0.021545221 -0.040449585 -0.004113863  
## google22 0.023124158 0.007276166 0.039553655  
## google13 0.002905748 -0.013164171 0.018639884  
## google23 -0.001029816 -0.021387635 0.017800659  
## tox\_eset1 0.053445025 -0.067978220 0.172621173  
## tox\_eset2 0.008479441 -0.089745580 0.103558645  
## tox\_eset3 0.012386082 -0.085562722 0.105800623

exp(estim)

## 2.5 % 97.5 %  
## (Intercept) 1.1895090 0.5318591 2.5404859  
## google1 1.0093314 0.9954347 1.0232553  
## google2 1.0079281 0.9942862 1.0217035  
## google11 1.0097604 0.9957778 1.0238736  
## google21 0.9941981 0.9796132 1.0084920  
## google12 0.9786852 0.9603576 0.9958946  
## google22 1.0233936 1.0073027 1.0403463  
## google13 1.0029100 0.9869221 1.0188147  
## google23 0.9989707 0.9788395 1.0179600  
## tox\_eset1 1.0548990 0.9342808 1.1884158  
## tox\_eset2 1.0085155 0.9141637 1.1091108  
## tox\_eset3 1.0124631 0.9179956 1.1116002

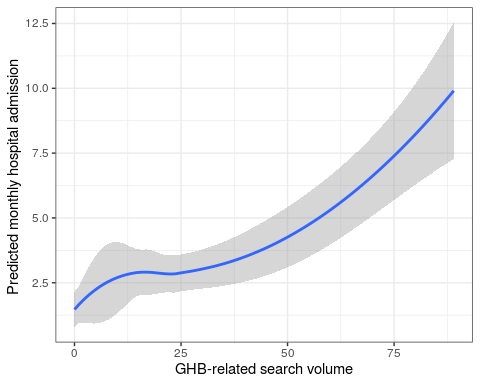
GTD<-GTD[-c(1:3),]  
  
GTD$predicted<-predict(fit)  
GTD$predictedcounts<-exp(GTD$predicted)  
library(ggplot2)  
p<-ggplot(GTD, aes(google11, predictedcounts)) + geom\_smooth(method="loess") + xlab("GHB-related search volume") + ylab("Predicted monthly hospital admission ") + theme\_bw()  
p

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



p<-ggplot(GTD, aes(google22, predictedcounts)) + geom\_smooth(method="loess") + xlab("GHB-related search volume") + ylab("Predicted monthly hospital admission ") + theme\_bw()  
p

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



m2 <- glm(tox\_eset ~ google1 + google2 + google11 + google21 + google12 + google22 + google13 + google23 +  
 tox\_eset1 + tox\_eset2 + tox\_eset3, family = "poisson", data=GTD)  
pchisq(2 \* (logLik(fit) - logLik(m2)), df = 1, lower.tail = FALSE)

## 'log Lik.' 0.170335 (df=13)

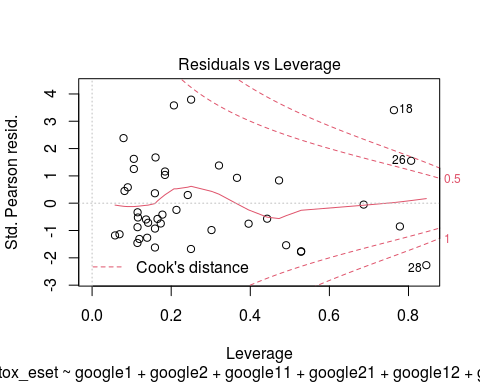
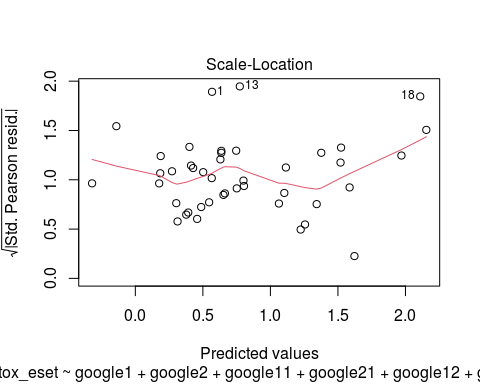
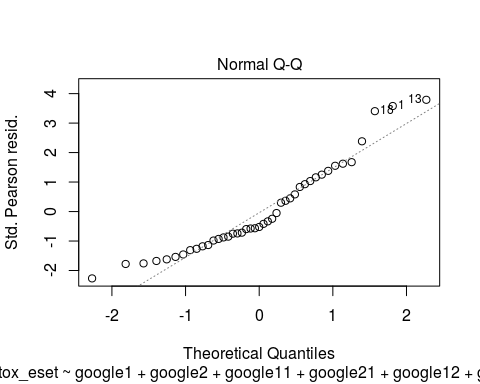
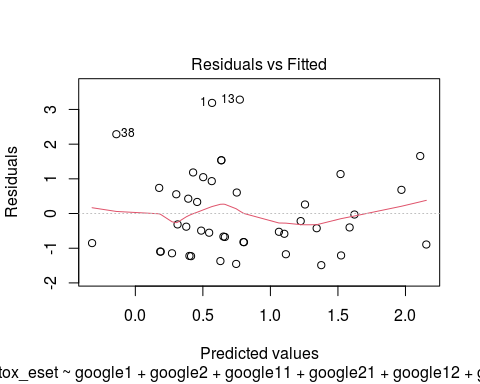
summary(m2)

##   
## Call:  
## glm(formula = tox\_eset ~ google1 + google2 + google11 + google21 +   
## google12 + google22 + google13 + google23 + tox\_eset1 + tox\_eset2 +   
## tox\_eset3, family = "poisson", data = GTD)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0542 -1.0751 -0.4140 0.6645 2.5976   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.1745213 0.3501166 0.498 0.61816   
## google1 0.0084080 0.0056818 1.480 0.13892   
## google2 0.0090804 0.0053557 1.695 0.08999 .   
## google11 0.0131457 0.0057904 2.270 0.02319 \*   
## google21 -0.0074297 0.0059300 -1.253 0.21024   
## google12 -0.0225048 0.0078068 -2.883 0.00394 \*\*  
## google22 0.0201390 0.0063907 3.151 0.00163 \*\*  
## google13 0.0021402 0.0068071 0.314 0.75321   
## google23 0.0015816 0.0084363 0.187 0.85129   
## tox\_eset1 0.0510520 0.0505450 1.010 0.31248   
## tox\_eset2 0.0003761 0.0436740 0.009 0.99313   
## tox\_eset3 0.0173420 0.0420597 0.412 0.68011   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 107.432 on 42 degrees of freedom  
## Residual deviance: 63.125 on 31 degrees of freedom  
## AIC: 186.31  
##   
## Number of Fisher Scoring iterations: 6

aic<-AIC(fit, m2)   
aic

## df AIC  
## fit 13 186.4267  
## m2 12 186.3066

plot(m2)



estim<-cbind(fit$coefficients, confint(m2))

## Waiting for profiling to be done...

estim

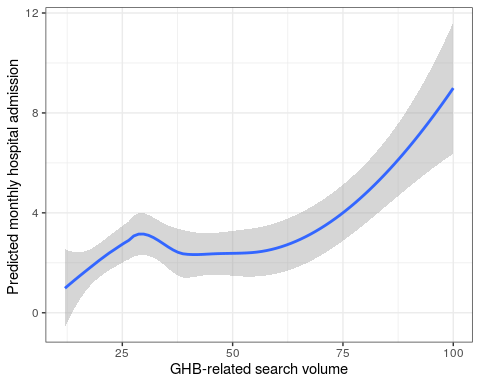
## 2.5 % 97.5 %  
## (Intercept) 0.173540645 -0.541979091 0.834597537  
## google1 0.009288089 -0.003111303 0.019283282  
## google2 0.007896824 -0.001804679 0.019345621  
## google11 0.009713090 0.001390049 0.024198066  
## google21 -0.005818769 -0.019658047 0.003732625  
## google12 -0.021545221 -0.038577838 -0.007805796  
## google22 0.023124158 0.007635728 0.032847352  
## google13 0.002905748 -0.011511479 0.015244494  
## google23 -0.001029816 -0.015829019 0.017427940  
## tox\_eset1 0.053445025 -0.050420322 0.148543810  
## tox\_eset2 0.008479441 -0.087666162 0.084329257  
## tox\_eset3 0.012386082 -0.068763427 0.096724563

exp(estim)

## 2.5 % 97.5 %  
## (Intercept) 1.1895090 0.5815961 2.3038866  
## google1 1.0093314 0.9968935 1.0194704  
## google2 1.0079281 0.9981969 1.0195340  
## google11 1.0097604 1.0013910 1.0244932  
## google21 0.9941981 0.9805339 1.0037396  
## google12 0.9786852 0.9621568 0.9922246  
## google22 1.0233936 1.0076650 1.0333928  
## google13 1.0029100 0.9885545 1.0153613  
## google23 0.9989707 0.9842956 1.0175807  
## tox\_eset1 1.0548990 0.9508297 1.1601436  
## tox\_eset2 1.0085155 0.9160666 1.0879871  
## tox\_eset3 1.0124631 0.9335475 1.1015569

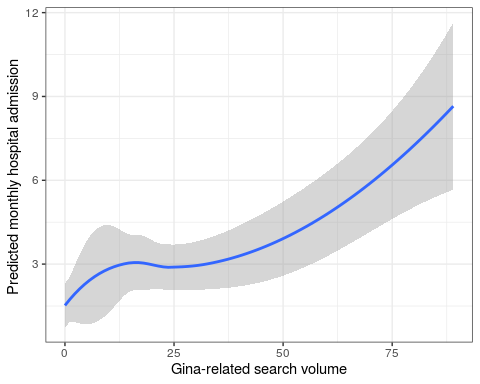
GTD$predicted<-predict(m2)  
GTD$predictedcounts<-exp(GTD$predicted)  
library(ggplot2)  
p<-ggplot(GTD, aes(google11, predictedcounts)) + geom\_smooth(method="loess") + xlab("GHB-related search volume") + ylab("Predicted monthly hospital admission ") + theme\_bw()  
p

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



p<-ggplot(GTD, aes(google22, predictedcounts)) + geom\_smooth(method="loess") + xlab("Gina-related search volume") + ylab("Predicted monthly hospital admission ") + theme\_bw()  
p

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



mean(GTD$tox\_eset)

## [1] 2.651163

var(GTD$tox\_eset)

## [1] 7.327796