Lyme Mobility ARIMA

Sulyok & Walker

4/7/2021

library(readr)  
#lyme\_file <- read\_delim("lyme\_file.csv", ";", escape\_double = FALSE, col\_types = cols(date = #col\_date(format = "%d/%m/%y")),   
#trim\_ws = TRUE)  
  
lyme\_file <- read.csv2("/cloud/project/lyme\_file.csv")  
lyme\_file<-na.omit(lyme\_file)  
  
lyme\_file$date<-as.Date(lyme\_file$date, format = "%d.%m.%y")  
summary(lyme\_file)

## date drivingweekmean transitweekmean walkingweekmean  
## Min. :2020-02-16 Min. : 46.14 Min. : 31.18 Min. : 45.8   
## 1st Qu.:2020-05-17 1st Qu.: 75.03 1st Qu.: 74.65 1st Qu.: 74.1   
## Median :2020-08-16 Median :102.61 Median : 90.65 Median :104.9   
## Mean :2020-08-16 Mean :103.43 Mean : 94.93 Mean :102.2   
## 3rd Qu.:2020-11-15 3rd Qu.:133.43 3rd Qu.:125.81 3rd Qu.:133.0   
## Max. :2021-02-14 Max. :157.24 Max. :152.76 Max. :155.9   
## stringency retail grocery parks   
## Min. :11.11 Min. :-65.143 Min. :-30.000 Min. :-16.714   
## 1st Qu.:55.09 1st Qu.:-57.000 1st Qu.:-13.143 1st Qu.: 3.143   
## Median :61.57 Median :-22.571 Median : -1.571 Median : 20.714   
## Mean :61.63 Mean :-28.473 Mean : -4.652 Mean : 42.625   
## 3rd Qu.:76.85 3rd Qu.: -5.571 3rd Qu.: 1.714 3rd Qu.: 84.571   
## Max. :83.33 Max. : 5.500 Max. : 27.000 Max. :149.429   
## transit workplace residential week   
## Min. :-58.00 Min. :-58.57 Min. : 0.000 Min. : 1   
## 1st Qu.:-48.29 1st Qu.:-29.14 1st Qu.: 3.286 1st Qu.:14   
## Median :-28.57 Median :-21.00 Median : 7.714 Median :27   
## Mean :-30.11 Mean :-23.89 Mean : 7.973 Mean :27   
## 3rd Qu.:-16.86 3rd Qu.:-15.00 3rd Qu.:12.286 3rd Qu.:40   
## Max. : 2.00 Max. : 0.00 Max. :18.857 Max. :53   
## lymecases   
## Min. : 31.0   
## 1st Qu.: 57.0   
## Median :147.0   
## Mean :265.8   
## 3rd Qu.:371.0   
## Max. :870.0

lapply(lyme\_file[c(2:11, 13)], shapiro.test)

## $drivingweekmean  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.93912, p-value = 0.009398  
##   
##   
## $transitweekmean  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.95599, p-value = 0.04896  
##   
##   
## $walkingweekmean  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.937, p-value = 0.007699  
##   
##   
## $stringency  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.88673, p-value = 0.0001174  
##   
##   
## $retail  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.86454, p-value = 2.443e-05  
##   
##   
## $grocery  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.91578, p-value = 0.001167  
##   
##   
## $parks  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.8857, p-value = 0.0001089  
##   
##   
## $transit  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.93215, p-value = 0.004918  
##   
##   
## $workplace  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.92797, p-value = 0.003371  
##   
##   
## $residential  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.92294, p-value = 0.00216  
##   
##   
## $lymecases  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.81598, p-value = 1.199e-06

spearmanf<-function(x) {  
 cor.test(lyme\_file$lymecases, x)$p.value  
}  
p<-as.data.frame(lapply(lyme\_file[2:11], spearmanf))  
p<-as.vector(p)  
padj<-p.adjust(p,"holm",10)  
padj

## drivingweekmean transitweekmean walkingweekmean stringency retail   
## 6.856049e-10 2.616879e-03 1.911682e-07 3.001512e-01 1.531436e-06   
## grocery parks transit workplace residential   
## 2.616879e-03 1.325816e-19 1.948479e-04 1.131213e-01 1.303302e-05

spearmanf<-function(x) {  
 cor.test(lyme\_file$lymecases, x)$estimate  
}  
est<-as.data.frame(lapply(lyme\_file[2:11], spearmanf))  
est

## drivingweekmean transitweekmean walkingweekmean stringency retail  
## cor 0.7532957 0.4457556 0.6783459 -0.1450246 0.6421414  
## grocery parks transit workplace residential  
## cor 0.453236 0.9053086 0.5335097 0.2635376 -0.5988318

contemp<-rbind(est, padj)  
contemp

## drivingweekmean transitweekmean walkingweekmean stringency retail  
## cor 7.532957e-01 0.445755582 6.783459e-01 -0.1450246 6.421414e-01  
## 2 6.856049e-10 0.002616879 1.911682e-07 0.3001512 1.531436e-06  
## grocery parks transit workplace residential  
## cor 0.453235964 9.053086e-01 0.5335097031 0.2635376 -5.988318e-01  
## 2 0.002616879 1.325816e-19 0.0001948479 0.1131213 1.303302e-05

library(Hmisc)

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

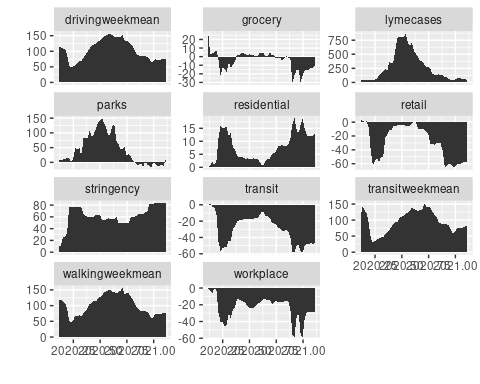
ccfspear<-function(y){ ccfspearmanx <- sapply( -3:3, function(l) cor.test(y, Hmisc::Lag(lyme\_file$lymecases,l),method = "spearman", use = "complete.obs", exact=FALSE)$estimate)  
}  
  
est<-as.data.frame(lapply(lyme\_file[2:11], ccfspear))  
  
ccfspear<-function(y){ ccfspearmanx <- sapply( -3:3, function(l) cor.test(y, Hmisc::Lag(lyme\_file$lymecases,l),method = "spearman", use = "complete.obs", exact=FALSE)$p.value)  
}  
  
p<-as.data.frame(lapply(lyme\_file[2:11], ccfspear))  
p<-as.matrix(p)  
padjust<-matrix(p.adjust(p, "holm", 70), nrow=7, ncol=10)  
padjust

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1.030773e-02 1.000000e+00 7.188969e-02 1.000000e+00 5.796880e-02  
## [2,] 4.586225e-04 8.195944e-01 5.806039e-03 8.971776e-01 2.321095e-02  
## [3,] 7.348136e-06 1.962974e-01 2.034070e-04 4.157828e-01 1.162783e-02  
## [4,] 4.006174e-08 3.090666e-02 1.673522e-06 1.842451e-01 4.497554e-03  
## [5,] 1.197153e-11 9.906363e-04 1.441556e-09 1.932415e-02 2.898585e-04  
## [6,] 1.478655e-15 1.344637e-06 3.549070e-13 4.586225e-04 8.985510e-06  
## [7,] 3.327436e-21 9.568091e-11 6.187088e-18 1.612858e-06 5.649610e-07  
## [,6] [,7] [,8] [,9] [,10]  
## [1,] 1.842451e-01 1.624768e-14 2.251047e-01 1.000000000 1.962974e-01  
## [2,] 1.173394e-01 1.177133e-14 1.532331e-01 1.000000000 1.173394e-01  
## [3,] 7.401433e-02 2.750420e-13 7.401433e-02 1.000000000 6.188786e-02  
## [4,] 3.172052e-02 1.824615e-11 2.402644e-02 0.685941027 1.803414e-02  
## [5,] 5.101221e-03 2.022157e-09 9.357701e-04 0.187700410 6.379077e-04  
## [6,] 8.815148e-04 9.513125e-08 2.783123e-05 0.039344225 1.410062e-05  
## [7,] 7.565156e-05 6.249852e-06 4.933157e-07 0.003659362 4.805127e-07

est

## drivingweekmean transitweekmean walkingweekmean stringency retail  
## 1 0.4882219 0.1088246 0.4061952 -0.1423657 0.4180399  
## 2 0.5729155 0.2221971 0.5017762 -0.2047386 0.4535989  
## 3 0.6535457 0.3298040 0.5877556 -0.2777455 0.4748665  
## 4 0.7273241 0.4333784 0.6734542 -0.3416661 0.5019462  
## 5 0.8151390 0.5473680 0.7704820 -0.4566812 0.5795217  
## 6 0.8791239 0.6873331 0.8459519 -0.5734035 0.6547002  
## 7 0.9337015 0.8064354 0.9081042 -0.6893385 0.7056690  
## grocery parks transit workplace residential  
## 1 0.3498451 0.8697918 0.3221654 0.1114127 -0.3380174  
## 2 0.3759196 0.8676201 0.3594722 0.1380465 -0.3761657  
## 3 0.3955753 0.8434738 0.3945812 0.1753453 -0.4061412  
## 4 0.4311357 0.8070045 0.4431562 0.2382950 -0.4559553  
## 5 0.5019219 0.7667478 0.5494161 0.3374111 -0.5598428  
## 6 0.5561767 0.7258882 0.6335182 0.4299455 -0.6463785  
## 7 0.6186679 0.6667227 0.7078970 0.5218550 -0.7085345

###################modelling#########################################  
library(ggfortify)  
autoplot(ts(lyme\_file[c(2:11, 13)], start=c(2020,7), frequency=52.18), ncol=3, geom="ribbon")



library(tsibble)

##   
## Attaching package: 'tsibble'

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

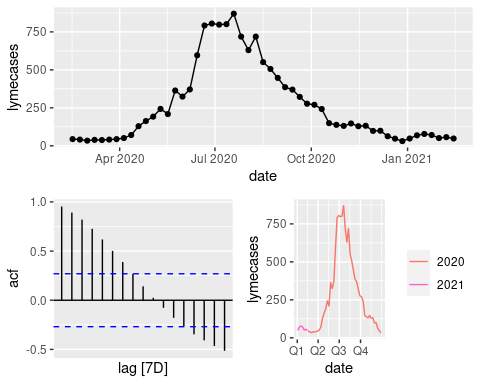
library(fable)

## Loading required package: fabletools

library(feasts)  
  
  
gon<-as\_tsibble(lyme\_file)

## Using `date` as index variable.

gg\_tsdisplay(gon, y=lymecases) # does not look stationary



library(forecast)

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

## Registered S3 methods overwritten by 'forecast':  
## method from   
## autoplot.Arima ggfortify  
## autoplot.acf ggfortify  
## autoplot.ar ggfortify  
## autoplot.bats ggfortify  
## autoplot.decomposed.ts ggfortify  
## autoplot.ets ggfortify  
## autoplot.forecast ggfortify  
## autoplot.stl ggfortify  
## autoplot.ts ggfortify  
## fitted.ar ggfortify  
## fortify.ts ggfortify  
## residuals.ar ggfortify

ndiffs(gon$lymecases, test="kpss")

## [1] 1

lymediff<-diff(gon$lymecases)  
ndiffs(lymediff, test="kpss") #ok now

## [1] 0

ndiffs(gon$parks, test="kpss")

## [1] 1

parkdiff<-diff(gon$parks)  
ndiffs(parkdiff, test="kpss")

## [1] 0

fit <- gon[-c(1:4),] %>% ###later the cb will drop the 4 four obs, so to make the models compareble we delete the first 4 obs  
 model(  
 arima = ARIMA(lymecases~ 1+ pdq(p = 0:5, d = 1:2, q = 0:5), trace=TRUE)  
 )

## Model specification Selection metric  
## ARIMA(2,1,2)(0,0,0)[1]+c 550.499135  
## ARIMA(0,1,0)(0,0,0)[1]+c 546.575170  
## ARIMA(1,1,0)(0,0,0)[1]+c 546.463784  
## ARIMA(0,1,1)(0,0,0)[1]+c 546.748064  
## ARIMA(1,1,1)(0,0,0)[1]+c 546.164269  
## ARIMA(1,1,2)(0,0,0)[1]+c 548.141371  
## ARIMA(2,1,1)(0,0,0)[1]+c 548.466069  
## ARIMA(0,1,2)(0,0,0)[1]+c 549.094259  
## ARIMA(2,1,0)(0,0,0)[1]+c 548.303964

fit

## # A mable: 1 x 1  
## arima  
## <model>  
## 1 <ARIMA(1,1,1) w/ drift>

glance(fit)

## # A tibble: 1 x 8  
## .model sigma2 log\_lik AIC AICc BIC ar\_roots ma\_roots   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <list> <list>   
## 1 arima 4517. -269. 545. 546. 553. <cpl [1]> <cpl [1]>

report(fit)

## Series: lymecases   
## Model: ARIMA(1,1,1) w/ drift   
##   
## Coefficients:  
## ar1 ma1 constant  
## 0.8227 -0.6406 0.1066  
## s.e. 0.1446 0.1836 3.1951  
##   
## sigma^2 estimated as 4517: log likelihood=-268.62  
## AIC=545.23 AICc=546.16 BIC=552.72

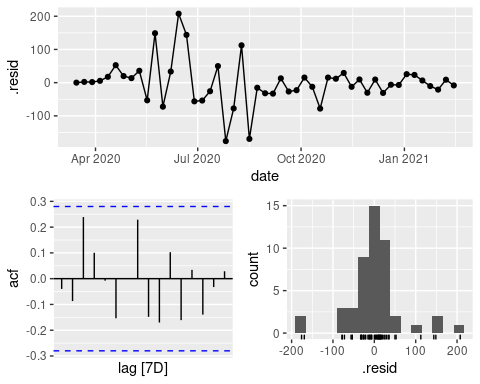
coefficients(fit)

## # A tibble: 3 x 6  
## .model term estimate std.error statistic p.value  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 arima ar1 0.823 0.145 5.69 0.000000748  
## 2 arima ma1 -0.641 0.184 -3.49 0.00105   
## 3 arima constant 0.107 3.20 0.0334 0.974

accuracy(fit)

## # A tibble: 1 x 10  
## .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 arima Training -0.365 64.4 41.7 -0.0441 17.2 0.936 0.937 -0.0399

gg\_tsresiduals(fit) #looks fine



fitselected <-gon[-c(1:4),] %>%  
 model(  
 arima = ARIMA(lymecases~ 1+ pdq(p = 1, d = 1, q = 1) + walkingweekmean + stringency + parks, trace=TRUE)  
 )

## Model specification Selection metric  
## ARIMA(1,1,1)(0,0,0)[1]+c 553.312799

fitselected

## # A mable: 1 x 1  
## arima  
## <model>  
## 1 <LM w/ ARIMA(1,1,1) errors>

glance(fitselected )

## # A tibble: 1 x 8  
## .model sigma2 log\_lik AIC AICc BIC ar\_roots ma\_roots   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <list> <list>   
## 1 arima 4772. -268. 551. 553. 564. <cpl [1]> <cpl [1]>

report(fitselected )

## Series: lymecases   
## Model: LM w/ ARIMA(1,1,1) errors   
##   
## Coefficients:  
## ar1 ma1 walkingweekmean stringency parks intercept  
## 0.8030 -0.6410 1.0756 0.5885 0.0483 0.3335  
## s.e. 0.1744 0.2113 1.3613 1.6882 0.4875 16.4346  
##   
## sigma^2 estimated as 4772: log likelihood=-268.26  
## AIC=550.51 AICc=553.31 BIC=563.61

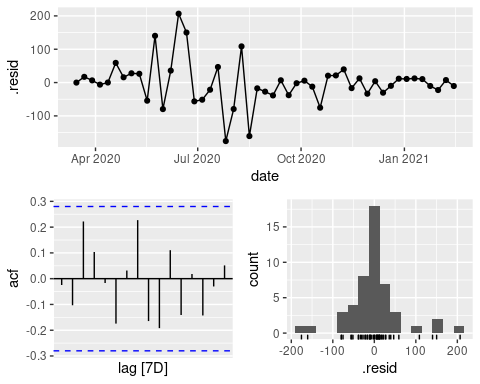
coefficients(fitselected )

## # A tibble: 6 x 6  
## .model term estimate std.error statistic p.value  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 arima ar1 0.803 0.174 4.60 0.0000305  
## 2 arima ma1 -0.641 0.211 -3.03 0.00390   
## 3 arima walkingweekmean 1.08 1.36 0.790 0.433   
## 4 arima stringency 0.589 1.69 0.349 0.729   
## 5 arima parks 0.0483 0.488 0.0991 0.921   
## 6 arima intercept 0.334 16.4 0.0203 0.984

accuracy(fitselected )

## # A tibble: 1 x 10  
## .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 arima Training -0.483 64.0 41.5 0.248 17.7 0.931 0.931 -0.0245

gg\_tsresiduals(fitselected )



fitselected <- gon[-c(1:4),] %>%  
 model(  
 arima = ARIMA(lymecases~ 1+ pdq(p = 1, d = 1, q = 1) + parks, trace=TRUE ))

## Model specification Selection metric  
## ARIMA(1,1,1)(0,0,0)[1]+c 548.608760

fitselected

## # A mable: 1 x 1  
## arima  
## <model>  
## 1 <LM w/ ARIMA(1,1,1) errors>

report(fitselected )

## Series: lymecases   
## Model: LM w/ ARIMA(1,1,1) errors   
##   
## Coefficients:  
## ar1 ma1 parks intercept  
## 0.8210 -0.6453 0.1096 0.5765  
## s.e. 0.1484 0.1877 0.4740 17.6376  
##   
## sigma^2 estimated as 4616: log likelihood=-268.59  
## AIC=547.18 AICc=548.61 BIC=556.54

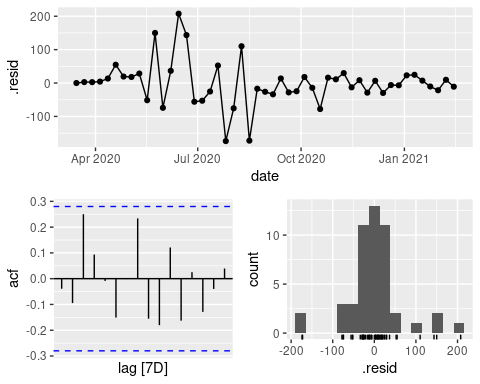
coefficients(fitselected ) #not significant

## # A tibble: 4 x 6  
## .model term estimate std.error statistic p.value  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 arima ar1 0.821 0.148 5.53 0.00000128  
## 2 arima ma1 -0.645 0.188 -3.44 0.00122   
## 3 arima parks 0.110 0.474 0.231 0.818   
## 4 arima intercept 0.577 17.6 0.0327 0.974

accuracy(fitselected )

## # A tibble: 1 x 10  
## .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 arima Training -0.369 64.4 41.7 -0.232 17.2 0.935 0.937 -0.0391

gg\_tsresiduals(fitselected )



augment(fitselected) %>%  
 features(.innov, ljung\_box)

## # A tibble: 1 x 3  
## .model lb\_stat lb\_pvalue  
## <chr> <dbl> <dbl>  
## 1 arima 0.0796 0.778

####################GAM with distibuted lag#####################  
  
lyme\_file$NumDate <- as.numeric(lyme\_file$date)-min(as.numeric(lyme\_file$date))  
library(dlnm)

## This is dlnm 2.4.5. For details: help(dlnm) and vignette('dlnmOverview').

lyme\_file$cb1 <- crossbasis(lyme\_file$drivingweekmean , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
lyme\_file$cb2 <- crossbasis(lyme\_file$transitweekmean , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
lyme\_file$cb3 <- crossbasis(lyme\_file$walkingweekmean , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
lyme\_file$cb4 <- crossbasis(lyme\_file$stringency , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
lyme\_file$cb5 <- crossbasis(lyme\_file$retail , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
lyme\_file$cb6 <- crossbasis(lyme\_file$grocery , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
lyme\_file$cb7 <- crossbasis(lyme\_file$parks , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
lyme\_file$cb8 <- crossbasis(lyme\_file$transit , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
lyme\_file$cb9 <- crossbasis(lyme\_file$workplace , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
lyme\_file$cb10 <- crossbasis(lyme\_file$residential , lag=4, argvar=list(fun="bs"),arglag=list(df=3))  
  
  
lyme\_file1<-lyme\_file[-c(1:4),] ### to make the datasets using the same length (cb produces 4 NA beacues of the lags)  
  
library(mgcv)

## Loading required package: nlme

##   
## Attaching package: 'nlme'

## The following object is masked from 'package:forecast':  
##   
## getResponse

## The following object is masked from 'package:feasts':  
##   
## ACF

## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.

fitplain<-gam(lymecases ~s(NumDate), data=lyme\_file1, family="tw")  
summary(fitplain)

##   
## Family: Tweedie(p=1.174)   
## Link function: log   
##   
## Formula:  
## lymecases ~ s(NumDate)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.20594 0.02847 182.9 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(NumDate) 7.337 8.302 182.5 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.974 Deviance explained = 97.8%  
## -REML = 253.19 Scale est. = 2.1811 n = 49

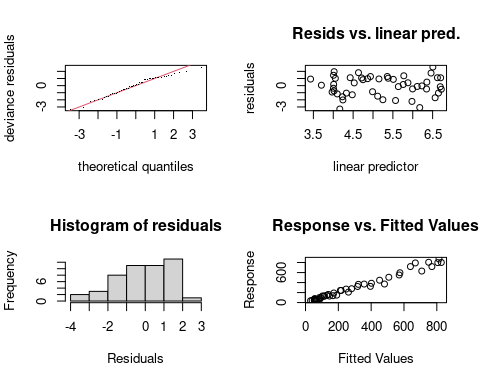
fitplain$gcv.ubre

## GCV.Cp   
## 253.1886

AIC(fitplain)

## [1] 488.5591

gam.check(fitplain)



##   
## Method: REML Optimizer: outer newton  
## full convergence after 7 iterations.  
## Gradient range [1.050796e-05,2.715565e-05]  
## (score 253.1886 & scale 2.181054).  
## Hessian positive definite, eigenvalue range [0.171687,33.48781].  
## Model rank = 10 / 10   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value   
## s(NumDate) 9.00 7.34 0.75 0.015 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

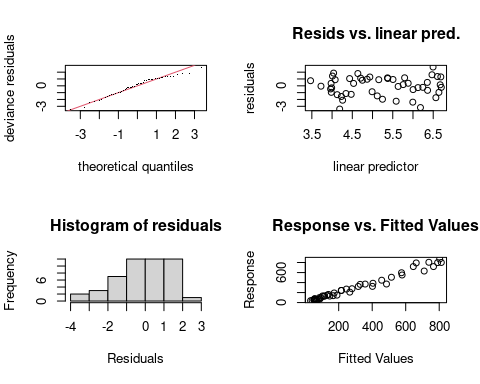
fitpark<-gam(lymecases~s(NumDate) + s(parks), data = lyme\_file1, select=TRUE, family=Tweedie(p=1.174))  
summary(fitpark)

##   
## Family: Tweedie(1.174)   
## Link function: log   
##   
## Formula:  
## lymecases ~ s(NumDate) + s(parks)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.20863 0.02811 185.3 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(NumDate) 6.401e+00 9 170.3 <2e-16 \*\*\*  
## s(parks) 8.692e-07 9 0.0 0.415   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.973 Deviance explained = 97.7%  
## GCV = 2.6216 Scale est. = 2.139 n = 49

AIC(fitpark)

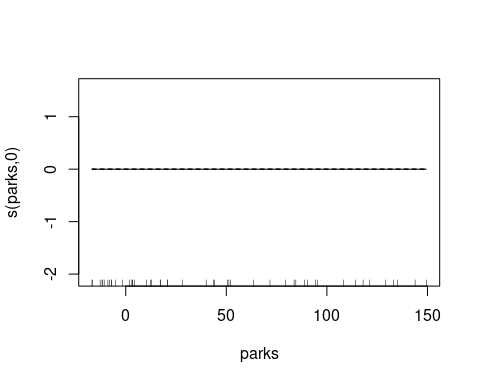
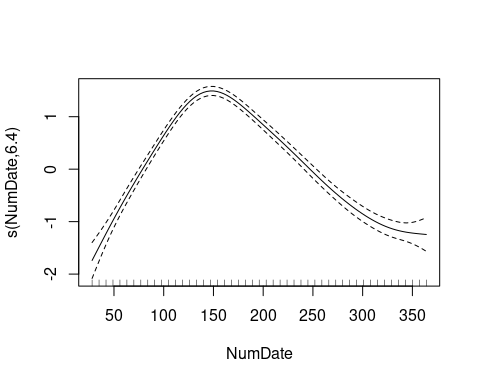
## [1] 485.7583

gam.check(fitpark)



##   
## Method: GCV Optimizer: outer newton  
## full convergence after 15 iterations.  
## Gradient range [-9.612677e-07,2.630451e-06]  
## (score 2.621624 & scale 2.139049).  
## Hessian positive definite, eigenvalue range [3.540021e-08,0.08346545].  
## Model rank = 19 / 19   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value   
## s(NumDate) 9.00e+00 6.40e+00 0.73 0.01 \*\*  
## s(parks) 9.00e+00 8.69e-07 1.27 0.95   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

plot(fitpark)



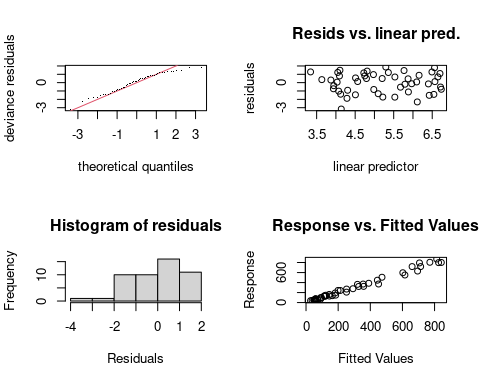
fitcbpark<-gam(lymecases~s(NumDate) + cb7, data = lyme\_file1, select=TRUE, family=Tweedie(p=1.174))  
summary(fitcbpark)

##   
## Family: Tweedie(1.174)   
## Link function: log   
##   
## Formula:  
## lymecases ~ s(NumDate) + cb7  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.9678 0.9536 7.307 2.9e-08 \*\*\*  
## cb7v1.l1 -0.5033 0.6849 -0.735 0.4678   
## cb7v1.l2 -0.9308 0.8065 -1.154 0.2571   
## cb7v1.l3 -1.1622 0.6835 -1.700 0.0989 .   
## cb7v2.l1 -0.5972 0.5183 -1.152 0.2579   
## cb7v2.l2 -1.1450 0.6826 -1.678 0.1033   
## cb7v2.l3 0.3449 0.3894 0.886 0.3825   
## cb7v3.l1 -0.4488 0.4423 -1.015 0.3180   
## cb7v3.l2 -1.6344 0.7605 -2.149 0.0394 \*   
## cb7v3.l3 -0.4177 0.3724 -1.122 0.2705   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(NumDate) 7.502 9 8.334 1.14e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.977 Deviance explained = 98.3%  
## GCV = 3.3413 Scale est. = 2.0657 n = 49

AIC(fitcbpark)

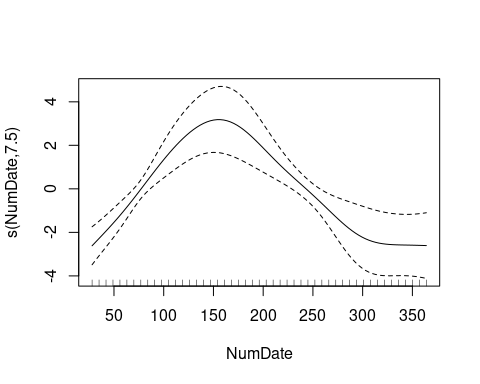
## [1] 490.5013

gam.check(fitcbpark)



##   
## Method: GCV Optimizer: outer newton  
## full convergence after 11 iterations.  
## Gradient range [1.219552e-07,1.311313e-06]  
## (score 3.341314 & scale 2.065665).  
## Hessian positive definite, eigenvalue range [0.01454991,0.08596097].  
## Model rank = 19 / 19   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value  
## s(NumDate) 9.0 7.5 0.88 0.18

plot(fitcbpark)



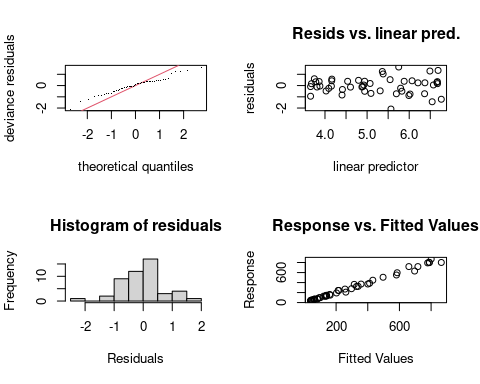
fitcb<-gam(lymecases~s(NumDate) + cb3 + cb7 + cb4, data = lyme\_file1, select=TRUE, family=Tweedie(p=1.174))  
summary(fitcb)

##   
## Family: Tweedie(1.174)   
## Link function: log   
##   
## Formula:  
## lymecases ~ s(NumDate) + cb3 + cb7 + cb4  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.49814 5.84206 1.112 0.2803   
## cb3v1.l1 -0.59839 1.04891 -0.570 0.5753   
## cb3v1.l2 2.38564 1.12892 2.113 0.0485 \*  
## cb3v1.l3 -0.43908 0.72010 -0.610 0.5495   
## cb3v2.l1 2.33826 1.21508 1.924 0.0699 .  
## cb3v2.l2 0.51590 1.12596 0.458 0.6522   
## cb3v2.l3 0.47198 0.74284 0.635 0.5330   
## cb3v3.l1 1.95667 0.99876 1.959 0.0654 .  
## cb3v3.l2 1.06685 1.15406 0.924 0.3672   
## cb3v3.l3 0.14658 0.65538 0.224 0.8255   
## cb7v1.l1 -2.03802 1.02435 -1.990 0.0617 .  
## cb7v1.l2 -2.25967 0.98168 -2.302 0.0332 \*  
## cb7v1.l3 -1.37077 0.70652 -1.940 0.0679 .  
## cb7v2.l1 0.24395 0.85857 0.284 0.7795   
## cb7v2.l2 -0.56728 0.83867 -0.676 0.5072   
## cb7v2.l3 -0.03293 0.57109 -0.058 0.9546   
## cb7v3.l1 -0.64653 0.76568 -0.844 0.4093   
## cb7v3.l2 -1.12353 0.97164 -1.156 0.2624   
## cb7v3.l3 -0.82303 0.52282 -1.574 0.1325   
## cb4v1.l1 11.51851 7.62731 1.510 0.1480   
## cb4v1.l2 -0.14850 4.22619 -0.035 0.9723   
## cb4v1.l3 7.03502 4.85480 1.449 0.1642   
## cb4v2.l1 4.97906 4.22978 1.177 0.2542   
## cb4v2.l2 -4.73858 3.34610 -1.416 0.1735   
## cb4v2.l3 3.96030 2.62978 1.506 0.1491   
## cb4v3.l1 7.79090 4.95328 1.573 0.1328   
## cb4v3.l2 -1.80689 3.58198 -0.504 0.6199   
## cb4v3.l3 5.58117 3.35666 1.663 0.1133   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(NumDate) 2.619 9 2.394 0.000883 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.977 Deviance explained = 99.4%  
## GCV = 3.648 Scale est. = 1.358 n = 49

AIC(fitcb)

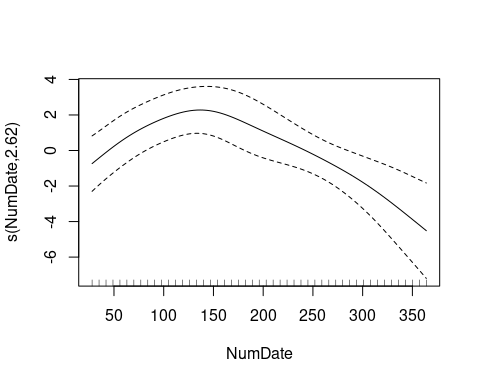
## [1] 468.1082

gam.check(fitcb)



##   
## Method: GCV Optimizer: outer newton  
## full convergence after 7 iterations.  
## Gradient range [-3.789734e-07,2.557653e-08]  
## (score 3.647965 & scale 1.357977).  
## Hessian positive definite, eigenvalue range [3.78974e-07,0.3519935].  
## Model rank = 37 / 37   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value  
## s(NumDate) 9.00 2.62 1.63 1

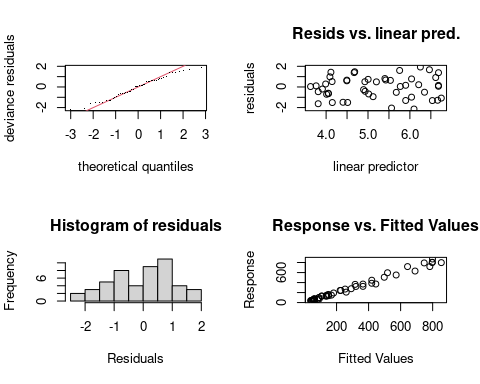
plot(fitcb)



fitsmall<-gam(lymecases~s(NumDate) + s(walkingweekmean) + s(parks) + s(stringency), data = lyme\_file1, select=TRUE, family=Tweedie(p=1.174))  
summary(fitsmall)

##   
## Family: Tweedie(1.174)   
## Link function: log   
##   
## Formula:  
## lymecases ~ s(NumDate) + s(walkingweekmean) + s(parks) + s(stringency)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.20145 0.02456 211.8 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(NumDate) 7.338e+00 9 42.331 <2e-16 \*\*\*  
## s(walkingweekmean) 3.707e+00 9 1.079 0.0253 \*   
## s(parks) 9.256e-06 9 0.000 0.8295   
## s(stringency) 4.627e+00 9 1.420 0.0141 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.977 Deviance explained = 98.7%  
## GCV = 2.4554 Scale est. = 1.6088 n = 49

gam.check(fitsmall)

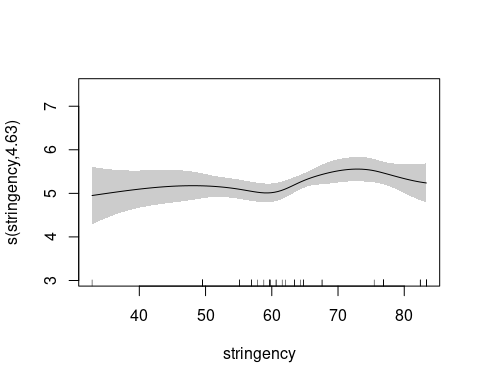
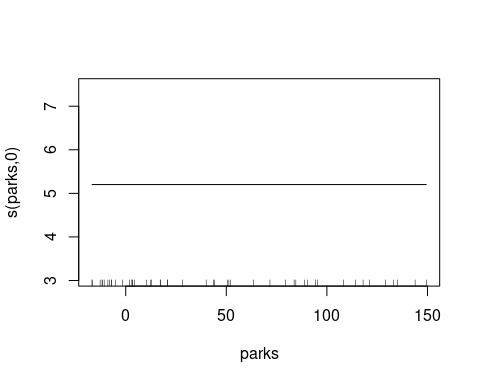
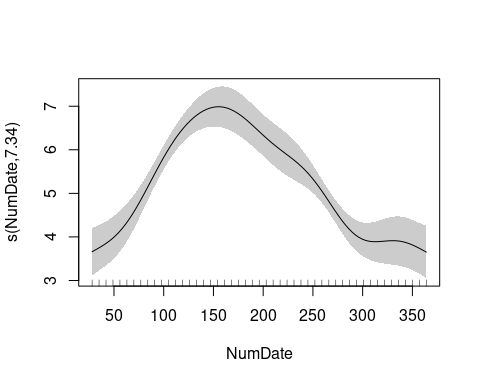


##   
## Method: GCV Optimizer: outer newton  
## full convergence after 14 iterations.  
## Gradient range [-8.377212e-07,3.948998e-07]  
## (score 2.455441 & scale 1.608823).  
## Hessian positive definite, eigenvalue range [4.518848e-09,0.1287026].  
## Model rank = 37 / 37   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value  
## s(NumDate) 9.00e+00 7.34e+00 1.08 0.66  
## s(walkingweekmean) 9.00e+00 3.71e+00 1.12 0.72  
## s(parks) 9.00e+00 9.26e-06 1.24 0.94  
## s(stringency) 9.00e+00 4.63e+00 1.14 0.76

AIC(fitsmall)

## [1] 476.2428

plot(fitsmall, shade=TRUE, shift = coef(fitsmall)[1])



AIC(fitplain)

## [1] 488.5591

AIC(fitpark)

## [1] 485.7583

AIC(fitcbpark)

## [1] 490.5013

AIC(fitsmall)

## [1] 476.2428

AIC(fitcb)

## [1] 468.1082

sessionInfo()

## R version 4.0.3 (2020-10-10)  
## Platform: x86\_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 20.04.2 LTS  
##   
## Matrix products: default  
## BLAS: /usr/lib/x86\_64-linux-gnu/atlas/libblas.so.3.10.3  
## LAPACK: /usr/lib/x86\_64-linux-gnu/atlas/liblapack.so.3.10.3  
##   
## 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   
##   
## other attached packages:  
## [1] mgcv\_1.8-33 nlme\_3.1-149 dlnm\_2.4.5 forecast\_8.14   
## [5] feasts\_0.2.1 fable\_0.3.0 fabletools\_0.3.1 tsibble\_1.0.0   
## [9] ggfortify\_0.4.11 Hmisc\_4.5-0 ggplot2\_3.3.3 Formula\_1.2-4   
## [13] survival\_3.2-7 lattice\_0.20-41 readr\_1.4.0   
##   
## loaded via a namespace (and not attached):  
## [1] tseries\_0.10-48 tidyr\_1.1.3 splines\_4.0.3   
## [4] distributional\_0.2.2 TTR\_0.24.2 latticeExtra\_0.6-29   
## [7] yaml\_2.2.1 pillar\_1.5.1 backports\_1.2.1   
## [10] glue\_1.4.2 quadprog\_1.5-8 digest\_0.6.27   
## [13] RColorBrewer\_1.1-2 checkmate\_2.0.0 colorspace\_2.0-0   
## [16] htmltools\_0.5.1.1 Matrix\_1.2-18 timeDate\_3043.102   
## [19] pkgconfig\_2.0.3 purrr\_0.3.4 scales\_1.1.1   
## [22] jpeg\_0.1-8.1 htmlTable\_2.1.0 tibble\_3.1.0   
## [25] generics\_0.1.0 farver\_2.1.0 ellipsis\_0.3.1   
## [28] withr\_2.4.1 urca\_1.3-0 nnet\_7.3-14   
## [31] cli\_2.4.0 quantmod\_0.4.18 magrittr\_2.0.1   
## [34] crayon\_1.4.1 evaluate\_0.14 fansi\_0.4.2   
## [37] anytime\_0.3.9 xts\_0.12.1 foreign\_0.8-80   
## [40] progressr\_0.7.0 tools\_4.0.3 data.table\_1.14.0   
## [43] hms\_1.0.0 lifecycle\_1.0.0 stringr\_1.4.0   
## [46] munsell\_0.5.0 cluster\_2.1.0 tsModel\_0.6   
## [49] compiler\_4.0.3 rlang\_0.4.10 grid\_4.0.3   
## [52] rstudioapi\_0.13 htmlwidgets\_1.5.3 base64enc\_0.1-3   
## [55] labeling\_0.4.2 rmarkdown\_2.7 gtable\_0.3.0   
## [58] fracdiff\_1.5-1 curl\_4.3 R6\_2.5.0   
## [61] gridExtra\_2.3 zoo\_1.8-9 lubridate\_1.7.10   
## [64] knitr\_1.31 dplyr\_1.0.5 utf8\_1.2.1   
## [67] stringi\_1.5.3 parallel\_4.0.3 Rcpp\_1.0.6   
## [70] vctrs\_0.3.7 rpart\_4.1-15 png\_0.1-7   
## [73] tidyselect\_1.1.0 xfun\_0.22 lmtest\_0.9-38