Accidents Prediction(Generalized Additive Models) With R Analytic Methods in Road Accident Research

Ali_Mirzaei

2022-11-07

##Libraries

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

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

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

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
## This is mgcv 1.8-40. For overview type 'help("mgcv-package")'.
```

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

```
## Loading required package: plotly
```

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

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

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

```
## Loading required package: GGally
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

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

```
## Loading required package: gratia
```

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

```
## Loading required package: ggeffects
```

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

```
## Loading required package: scico
```

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

```
## Loading required package: beepr
```

```
if(!require('visibly')) devtools::install_github('m-clark/visibly', upgrade = "never")
```

```
## Loading required package: visibly
```

```
if(!require('tidyext')) devtools::install_github('m-clark/tidyext', upgrade = "never")
```

```
## Loading required package: tidyext
```

```
##
## Attaching package: 'tidyext'
```

```
## The following object is masked from 'package:visibly':
##
##     create_prediction_data
```

Dataset

```
MF_read_CsV<-function(path,use_UTF8=TRUE,choose_file=FALSE,set_max_overlaps=TRUE){
  if(set_max_overlaps){
    options(ggrepel.max.overlaps = Inf)
  }
  if(choose_file){
    path=file.choose()
  }
  if(use_UTF8){
    data<- read.csv(path,encoding="UTF-8")
  }else{
    data<- read.csv(path)
  }

  return(data)
}
```

```
df<- MF_read_CsV(path="C:\\Users\\Traffic\\Desktop\\traffic_data\\df.csv")
```

##Show dataset

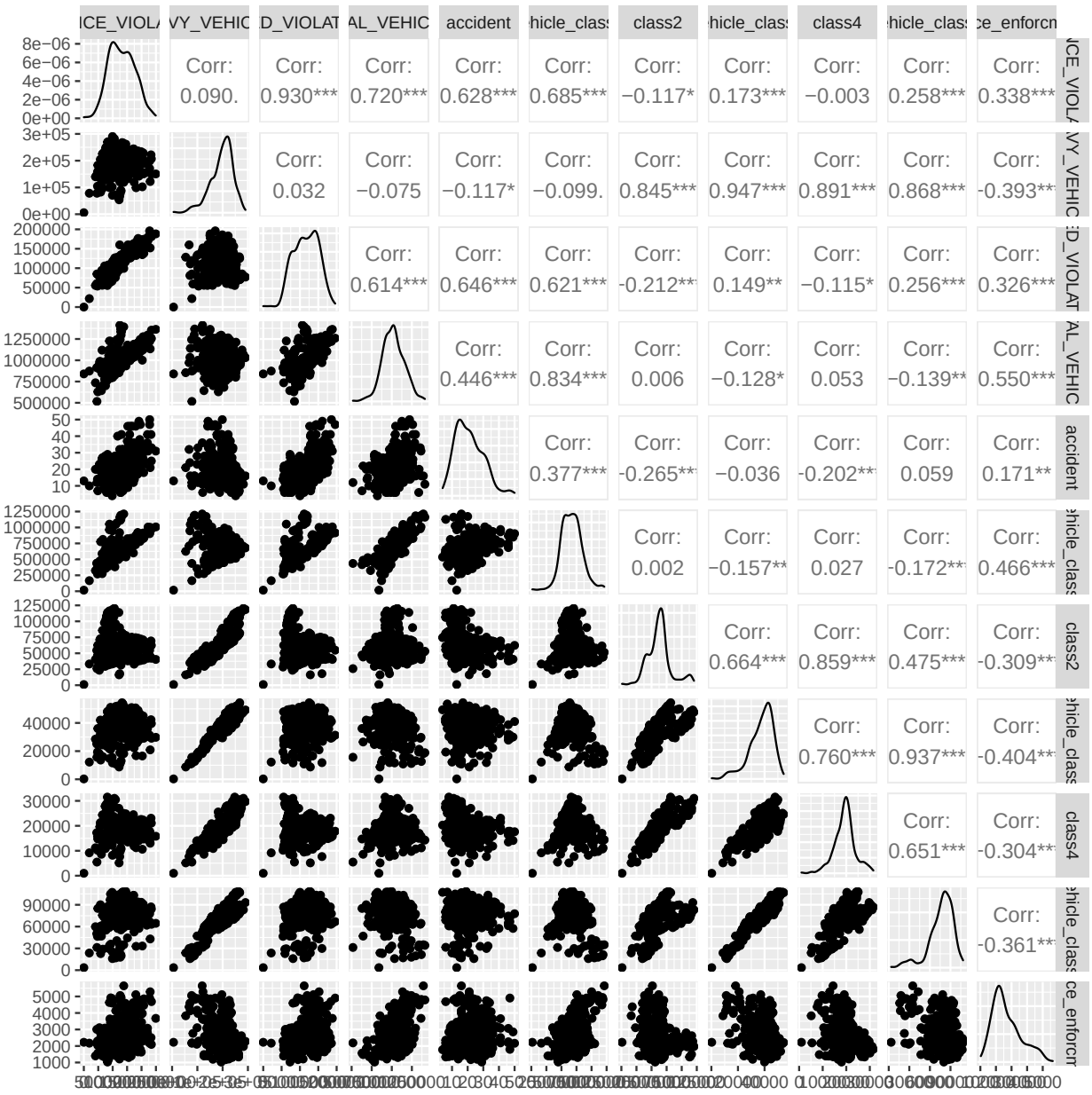
```
glimpse(df)
```

```
## Rows: 365
## Columns: 11
## $ DISTANCE_VIOLATIONS <int> 110796, 125998, 129162, 128019, 88914, 66045, 1284~
## $ HEAVY_VEHICLES      <int> 72388, 77201, 87328, 89462, 99464, 97415, 121246, ~
## $ SPEED_VIOLATIONS    <int> 110999, 119682, 118878, 117921, 62935, 58900, 1110~
## $ TOTAL_VEHICLES      <int> 1140261, 1243123, 1297016, 1367602, 1226635, 96912~
## $ accident            <int> 16, 15, 16, 16, 14, 14, 6, 11, 16, 14, 13, 12, 24,~
## $ vehicle_class1      <int> 1047169, 1109906, 1182792, 1191201, 1052151, 81258~
## $ class2              <int> 35124, 38483, 43725, 44285, 41236, 38708, 55471, 5~
## $ vehicle_class3      <int> 10838, 11201, 13171, 12661, 14786, 16070, 19318, 1~
## $ class4              <int> 9470, 9726, 10708, 11154, 14043, 13760, 14989, 142~
## $ vehicle_class5      <int> 16956, 17791, 19724, 21362, 29399, 28877, 31468, 3~
## $ police_enforcment   <int> 4134, 5058, 4939, 4821, 2136, 2108, 4685, 4790, 56~
```

Exploratory Data Analysis

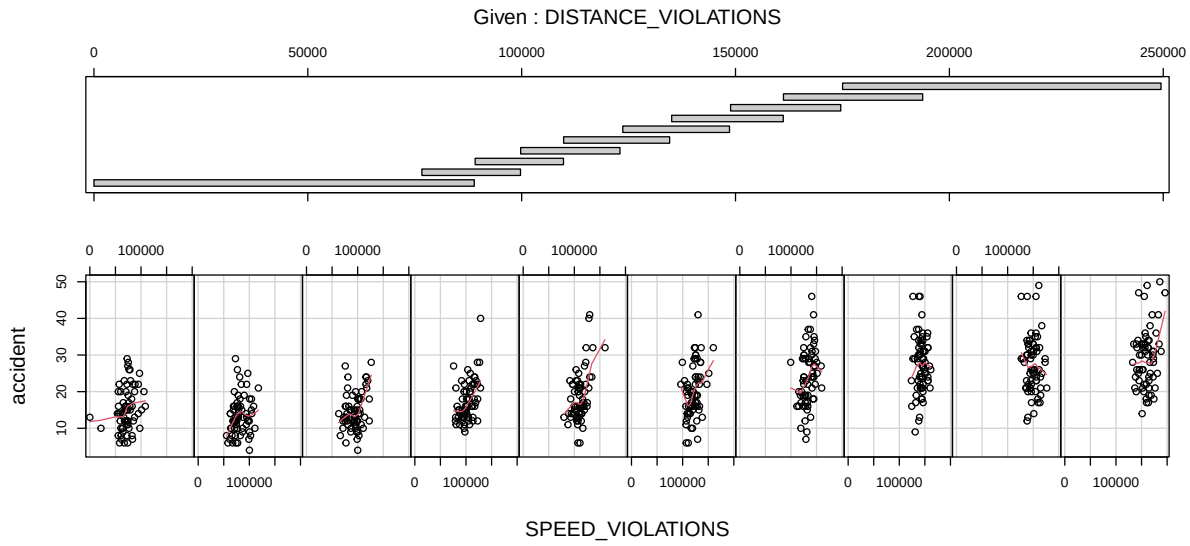
```
smooth <- function(data, mapping, ptcol, ptalpha=1, ptsize=1, linecol, ...) {
  p <- ggplot(data = data, mapping = mapping) +
    geom_point(color=ptcol, alpha=ptalpha, size=ptsize) +
    geom_smooth(color=linecol, ...)
  p
}

ggpairs(df)
```



#intraction analysis

```
coplot(accident ~ SPEED_VIOLATIONS | DISTANCE_VIOLATIONS, data = df,
       number = 10, rows = 1,
       panel = panel.smooth)
```



Model 1

```
mod_gam1 = gam(accident ~ s(SPEED_VIOLATIONS) +
                            s(DISTANCE_VIOLATIONS) +
                             s(HEAVY_VEHICLES) +
                            s(vehicle_class1) + s(police_enforcment) +ti(SPEED_VIOLATIONS,DISTANCE_VIO
                           data = df, select = T, family = 'poisson')

summary(mod_gam1)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## accident ~ s(SPEED_VIOLATIONS) + s(DISTANCE_VIOLATIONS) + s(HEAVY_VEHICLES) +
##     s(vehicle_class1) + s(police_enforcment) + ti(SPEED_VIOLATIONS,
##     DISTANCE_VIOLATIONS, HEAVY_VEHICLES, bs = "fs")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.99534    0.01575   190.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                       edf Ref.df Chi.sq
## s(SPEED_VIOLATIONS)                                 7.582      9  64.62
## s(DISTANCE_VIOLATIONS)                              7.859      9  24.88
## s(HEAVY_VEHICLES)                                   5.869      9  31.92
## s(vehicle_class1)                                   6.566      9  11.28
## s(police_enforcment)                                6.533      9  16.70
## ti(SPEED_VIOLATIONS,DISTANCE_VIOLATIONS,HEAVY_VEHICLES) 17.711     64  46.51
##                                                      p-value
## s(SPEED_VIOLATIONS)                                < 2e-16 ***
## s(DISTANCE_VIOLATIONS)                            0.000544 ***
## s(HEAVY_VEHICLES)                                  < 2e-16 ***
```

```
## s(vehicle_class1)                                        0.068370 .
## s(police_enforcment)                                     0.008482 **
## ti(SPEED_VIOLATIONS,DISTANCE_VIOLATIONS,HEAVY_VEHICLES) 2.83e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.581   Deviance explained = 64.6%
## UBRE = 0.59547  Scale est. = 1         n = 365
```

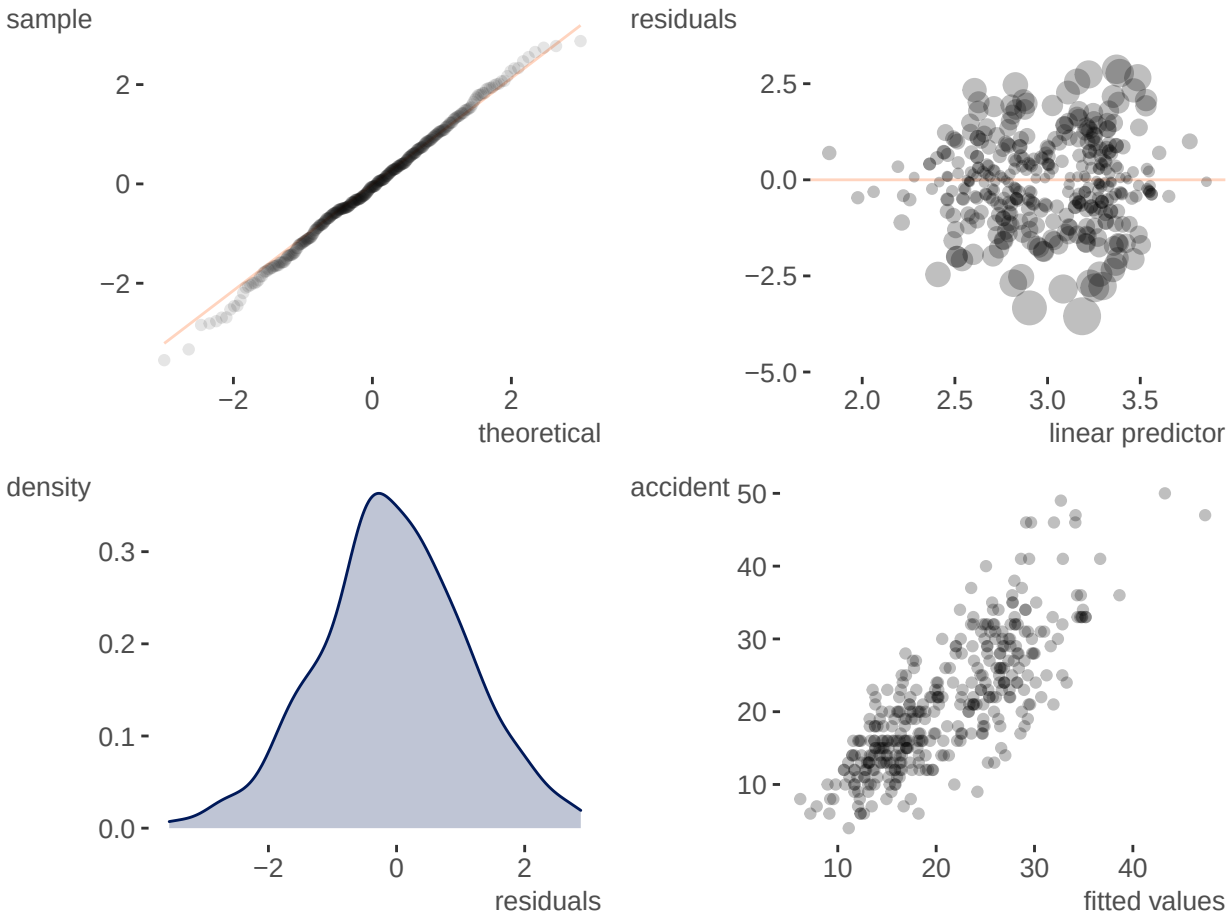
Model checking

```
test = capture.output(gam.check(mod_gam1, k.rep = 1000))
```

```
cat(paste0(test, collapse = '\n'))
```

```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 16 iterations.
## Gradient range [-3.651444e-08,4.421323e-07]
## (score 0.5954737 & scale 1).
## Hessian positive definite, eigenvalue range [5.664891e-11,0.00503388].
## Model rank =  110 / 110
##
## 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
## s(SPEED_VIOLATIONS)                                    9.00  7.58    0.94
## s(DISTANCE_VIOLATIONS)                                 9.00  7.86    0.98
## s(HEAVY_VEHICLES)                                      9.00  5.87    1.02
## s(vehicle_class1)                                      9.00  6.57    0.90
## s(police_enforcment)                                   9.00  6.53    1.05
## ti(SPEED_VIOLATIONS,DISTANCE_VIOLATIONS,HEAVY_VEHICLES) 64.00 17.71    1.01
##                                                         p-value
## s(SPEED_VIOLATIONS)                                       0.124
## s(DISTANCE_VIOLATIONS)                                    0.349
## s(HEAVY_VEHICLES)                                         0.597
## s(vehicle_class1)                                         0.028 *
## s(police_enforcment)                                      0.839
## ti(SPEED_VIOLATIONS,DISTANCE_VIOLATIONS,HEAVY_VEHICLES)   0.607
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot_gam_check(mod_gam1, scatter = TRUE)
```



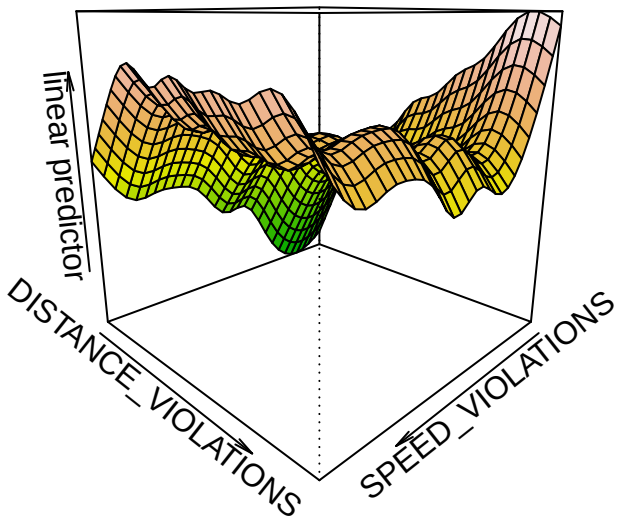
Concurvity

```
knitr::kable(concurvity(mod_gam1) %>%
  as_tibble(rownames = 'type'))
```

type	para	s(SPEED_VIOLATIONS)	s(DISTANCE_VIOLATIONS)	s(HOUR_OF_DAY_VIOLATIONS)	s(WEEKEND_VIOLATIONS)	s(road_type)	s(police_enforcement)	te(SPEED_VIOLATIONS,DISTANCE_VIOL
worst	0.9812849	0.9996554	0.9983304	0.9998456	0.9993387	0.8305681	0.9999042	
observed	0.9812849	0.9627623	0.8628428	0.9682509	0.8131498	0.6580489	0.7858729	
estimate	0.9812849	0.9656921	0.9710598	0.9875286	0.8918682	0.6459268	0.4971377	

##Model plot

```
vis.gam(mod_gam1, plot.type = "persp", color = "terrain",
        # changes the perspective
        main = "", theta=135)
```


linear predictor
DISTANCE_VIOLATIONS
SPEED_VIOLATIONS