GAMs and NonLinearity

Sar, North, Henry, Quinn

3/8/2022

```
library(ISLR)
library(dplyr)
```

```
## Warning: replacing previous import ’lifecycle::last_warnings’ by
## ’rlang::last_warnings’ when loading ’pillar’
```

```
##
## 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
```

```
library(readr)
```

```
## Warning: replacing previous import ’lifecycle::last_warnings’ by
## ’rlang::last_warnings’ when loading ’hms’
```

```
library(broom)
```

```
## Warning: package ’broom’ was built under R version 4.1.2
```

```
library(ggplot2)
library(splines)
library(tidymodels)
```

```
## Registered S3 method overwritten by ’tune’:
##   method                   from
##   required_pkgs.model_spec parsnip
```

```
## -- Attaching packages -------------------------------------- tidymodels 0.1.4 --
## v dials        0.0.10     v tibble       3.1.6
## v infer        1.0.0      v tidyr        1.1.4
## v modeldata    0.1.1      v tune         0.1.6
## v parsnip      0.1.7      v workflows    0.2.4
## v purrr        0.3.4      v workflowsets 0.1.0
## v recipes      0.1.17     v yardstick    0.0.9
## v rsample      0.1.1
```

```
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard()  masks scales::discard()
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```
tidymodels_prefer()
```

```
COVID_State <- read.csv("COVID - State - Daily.csv", na.strings = ".")

Employment_State <- read.csv("Employment - State - Daily.csv", na.strings = ".")

Mobility_State <- read.csv("Google Mobility - State - Daily.csv", na.strings = ".")

Spending_State <- read.csv("Affinity - State - Daily.csv", na.strings = ".")
```

```
COVID_State$Date<-as.Date(with(COVID_State,paste(year,month,day,sep="-")),"%Y-%m-%d")

Employment_State$Date<-as.Date(with(Employment_State,paste(year,month,day,sep="-")),"%Y-%m-%d")

Mobility_State$Date<-as.Date(with(Mobility_State,paste(year,month,day,sep="-")),"%Y-%m-%d")

Spending_State$Date<-as.Date(with(Spending_State,paste(year,month,day,sep="-")),"%Y-%m-%d")

full_data <- merge(merge(merge(COVID_State, Employment_State, by=c("Date","statefips")), Mobility_State
```

```
## Warning in merge.data.frame(merge(merge(COVID_State, Employment_State, by =
## c("Date", : column names 'year.x', 'month.x', 'day.x', 'year.y', 'month.y',
## 'day.y' are duplicated in the result
```

```
head(full_data)
```

```
##         Date statefips year.x month.x day.x new_case_count new_death_count
## 1 2020-02-24         1   2020       2    24             NA              NA
## 2 2020-02-24        10   2020       2    24             NA              NA
## 3 2020-02-24        11   2020       2    24             NA              NA
## 4 2020-02-24        12   2020       2    24             NA              NA
## 5 2020-02-24        13   2020       2    24             NA              NA
## 6 2020-02-24        15   2020       2    24             NA              NA
##   case_count death_count vaccine_count fullvaccine_count booster_first_count
## 1         NA          NA            NA                NA                  NA
## 2         NA          NA            NA                NA                  NA
## 3         NA          NA            NA                NA                  NA
## 4         NA          NA            NA                NA                  NA
## 5         NA          NA            NA                NA                  NA
## 6         NA          NA            NA                NA                  NA
##   new_vaccine_count new_fullvaccine_count new_booster_first_count
## 1                NA                    NA                      NA
## 2                NA                    NA                      NA
## 3                NA                    NA                      NA
## 4                NA                    NA                      NA
## 5                NA                    NA                      NA
## 6                NA                    NA                      NA
##   new_test_count test_count hospitalized_count new_case_rate case_rate
## 1             NA         NA                 NA            NA        NA
## 2             NA         NA                 NA            NA        NA
## 3             NA         NA                 NA            NA        NA
## 4             NA         NA                 NA            NA        NA
## 5             NA         NA                 NA            NA        NA
## 6             NA         NA                  0            NA        NA
##   new_death_rate death_rate new_test_rate test_rate new_vaccine_rate
## 1             NA         NA            NA        NA               NA
## 2             NA         NA            NA        NA               NA
## 3             NA         NA            NA        NA               NA
## 4             NA         NA            NA        NA               NA
## 5             NA         NA            NA        NA               NA
## 6             NA         NA            NA        NA               NA
##   vaccine_rate new_fullvaccine_rate fullvaccine_rate new_booster_first_rate
## 1           NA                   NA               NA                     NA
## 2           NA                   NA               NA                     NA
## 3           NA                   NA               NA                     NA
## 4           NA                   NA               NA                     NA
## 5           NA                   NA               NA                     NA
## 6           NA                   NA               NA                     NA
##   booster_first_rate hospitalized_rate year.y month.y day.y      emp emp_incq1
## 1                 NA                NA   2020       2    24  0.01580   0.00751
## 2                 NA                NA   2020       2    24  0.00537  -0.02670
## 3                 NA                NA   2020       2    24       NA        NA
## 4                 NA                NA   2020       2    24  0.00448  -0.00263
## 5                 NA                NA   2020       2    24  0.00532  -0.00537
## 6                 NA                 0   2020       2    24 -0.03530  -0.07190
##   emp_incq2 emp_incq3 emp_incq4 emp_incmiddle emp_incbelowmed emp_incabovemed
## 1   0.02320   0.01680        NA       0.01960        0.013600          0.0183
## 2   0.00570   0.01680    0.0242       0.01170       -0.011400          0.0206
## 3        NA        NA        NA            NA              NA              NA
## 4  -0.00458   0.01070    0.0164       0.00324       -0.003550          0.0133
## 5   0.00520   0.00873    0.0140       0.00710       -0.000838          0.0114
## 6  -0.04920  -0.00520        NA      -0.02980       -0.058300         -0.0112
##    emp_ss40 emp_ss60 emp_ss65 emp_ss70 year.x month.x day.x
## 1  0.001540 -0.00399  0.05300 -0.01620   2020       2    24
## 2  0.015400  0.01340  0.01030 -0.05550   2020       2    24
## 3        NA       NA       NA       NA   2020       2    24
## 4 -0.002320  0.00134  0.00576  0.01620   2020       2    24
## 5 -0.000237  0.00168  0.00889  0.00964   2020       2    24
## 6  0.054800       NA       NA -0.01530   2020       2    24
##   gps_retail_and_recreation gps_grocery_and_pharmacy gps_parks
## 1                   0.00286                 -0.00714    0.0557
## 2                   0.03710                  0.01290    0.2340
## 3                  -0.01140                 -0.03290    0.1400
## 4                   0.02710                  0.00714    0.0943
## 5                  -0.00571                 -0.02290    0.0186
## 6                   0.01140                 -0.00571    0.0814
##   gps_transit_stations gps_workplaces gps_residential gps_away_from_home year.y
## 1              0.06000        0.01290         0.00857           -0.00798   2020
## 2              0.07000        0.02860        -0.00571            0.00850   2020
## 3              0.00571       -0.01430         0.00714           -0.00492   2020
## 4              0.03430        0.01000         0.00143           -0.00138   2020
## 5              0.01710       -0.01140         0.01000           -0.00781   2020
## 6              0.02570        0.00714         0.00143           -0.00049   2020
##   month.y day.y freq spend_all spend_aap spend_acf spend_aer spend_apg
## 1       2    24    d   -0.0198   -0.1320   -0.0220   -0.1000   -0.0810
## 2       2    24    d   -0.0461    0.1130   -0.0279   -0.6280    0.4140
## 3       2    24    d    0.0192   -0.1280   -0.0113    0.0740   -0.0855
## 4       2    24    d   -0.0452   -0.0847   -0.0493   -0.1020   -0.0675
## 5       2    24    d   -0.0163   -0.0321   -0.0334    0.0287   -0.0308
## 6       2    24    d   -0.0504   -0.1210   -0.0447   -0.1650   -0.0851
##   spend_durables spend_nondurables spend_grf spend_gen spend_hic spend_hcs
## 1        -0.0317          -0.04750   -0.0223  -0.01050  -0.06180  -0.07310
## 2         0.0208           0.13400   -0.0284   0.63600   0.13400  -0.01060
## 3         0.0311          -0.00364    0.0294   0.00856   0.59500   0.02630
## 4        -0.0492          -0.04720   -0.0468  -0.03810  -0.08320   0.00175
## 5        -0.0164          -0.02450   -0.0110  -0.03000  -0.00361  -0.02010
## 6        -0.0118          -0.04380   -0.0173  -0.04770   0.16600  -0.08730
##   spend_inpersonmisc spend_remoteservices spend_sgh spend_tws
## 1             0.0062              0.02110   -0.0453   -0.1020
## 2            -0.1380             -0.15500   -0.1540   -0.0929
## 3             0.2100             -0.03610   -0.1230   -0.1360
## 4            -0.0815             -0.04600   -0.0426   -0.1030
## 5            -0.0658             -0.00774    0.0940   -0.1060
## 6            -0.0645             -0.04000   -0.2270   -0.0909
##   spend_retail_w_grocery spend_retail_no_grocery spend_all_incmiddle
## 1               -0.03910                 -0.0459            -0.02970
## 2                0.10200                  0.1560            -0.06480
## 3               -0.00169                 -0.0124            -0.06430
## 4               -0.04390                 -0.0421            -0.03880
## 5               -0.01640                 -0.0176            -0.01870
## 6               -0.03610                 -0.0498             0.00268
##   spend_all_q1 spend_all_q2 spend_all_q3 spend_all_q4 provisional
## 1      -0.0158      -0.0717     0.036100     0.009840           0
## 2       0.2240      -0.0565    -0.068700    -0.016000           0
## 3      -0.0265      -0.5850    -0.047300     0.039400           0
## 4      -0.0677      -0.0420    -0.035100    -0.035700           0
## 5      -0.0386      -0.0234    -0.015600    -0.000937           0
## 6           NA       0.0134     0.000257    -0.076700           0
```

```
full_data1 <- full_data %>%
  select(-year.x, -month.x, -day.x, - year.y, -month.y, -day.y, -year.x )


minnesota <- full_data1 %>%
  filter(statefips==27)
```

```
minnesota_cut <- minnesota %>%
  filter(Date > "2020-04-13")
```

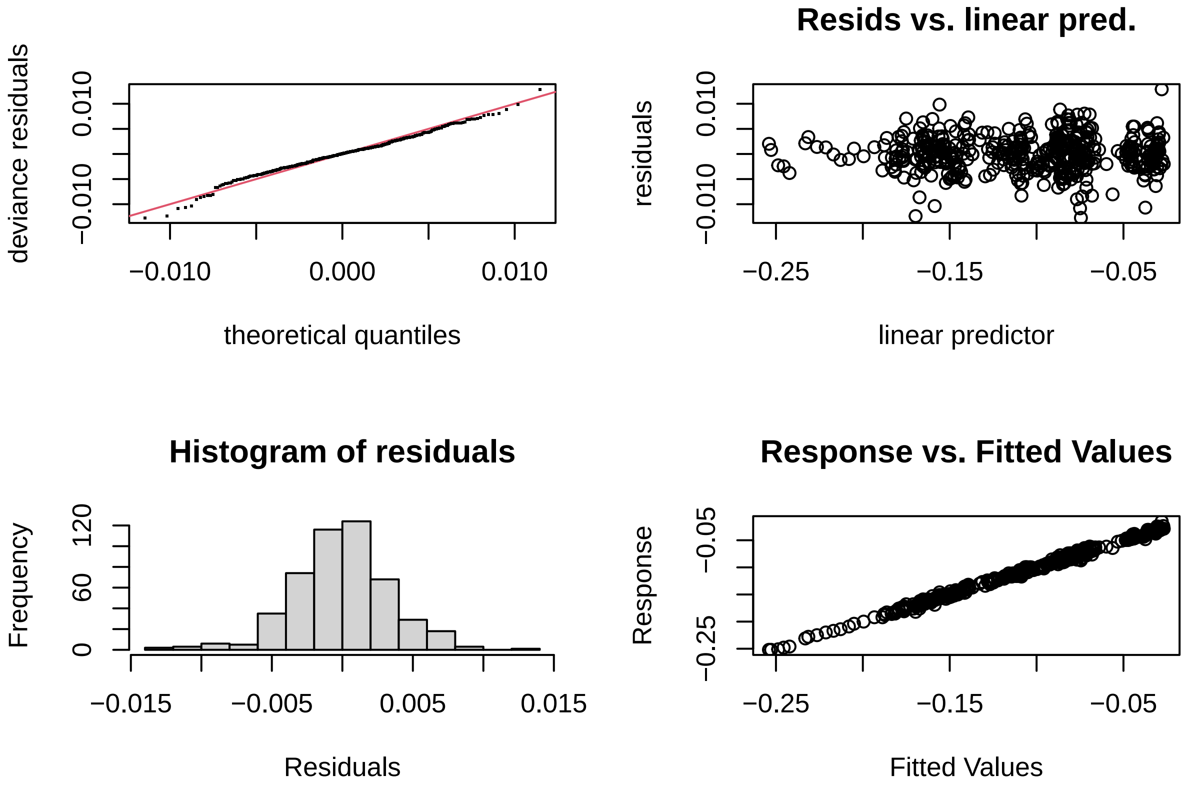
```
set.seed(123)

# Don't necessarily need to use gam_spec, can use lm_spec instead
gam_spec <-
  gen_additive_mod() %>%
  set_engine(engine = 'mgcv') %>%
  set_mode('regression')

lm_spec <-
  linear_reg() %>%
  set_engine(engine = 'lm') %>%
  set_mode('regression')

gam_mod <- fit(gam_spec,
           gps_away_from_home ~ s(case_rate) + s(hospitalized_rate, k=20) + s(emp_incq1, k=20) + s(emp_
    data = minnesota_cut)
```

```
# Diagnostics: Check to see if the number of knots is large enough
par(mfrow=c(2,2))
gam_mod %>% pluck('fit') %>% mgcv::gam.check()
```

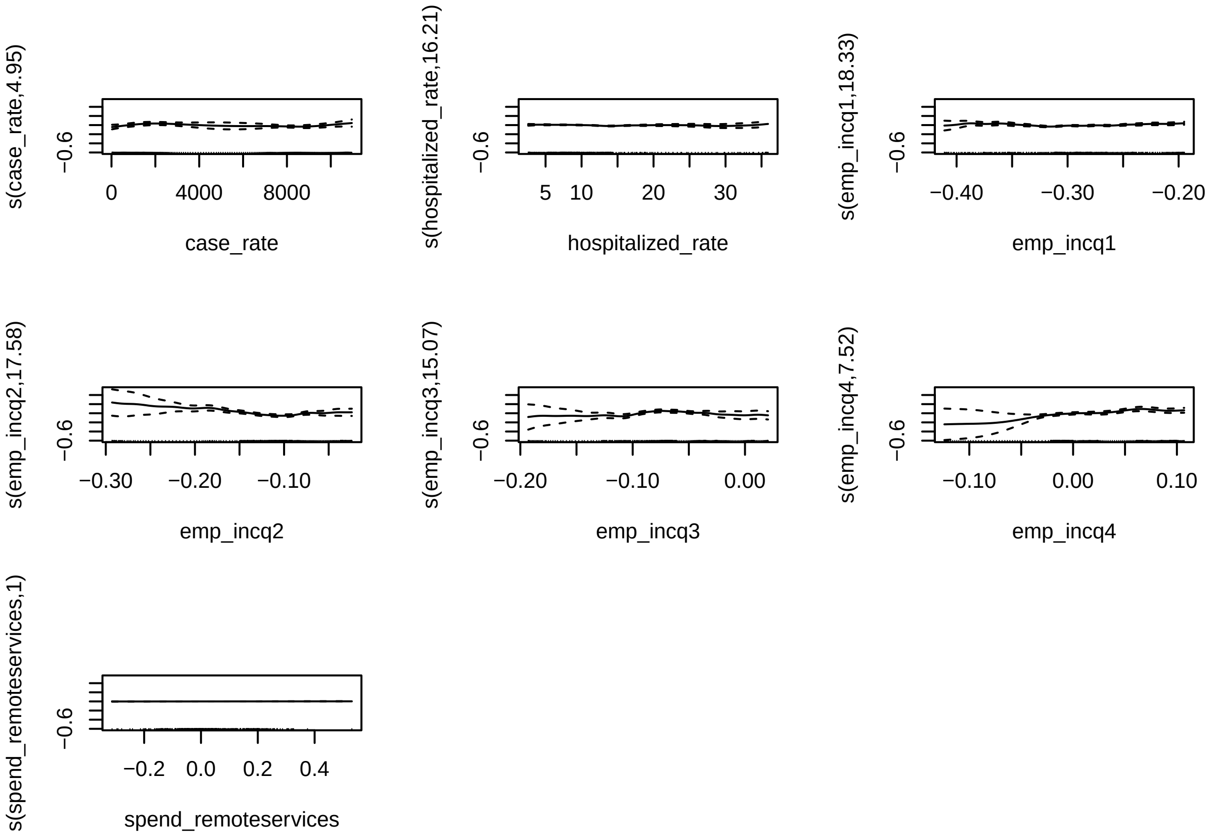


```
## 
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 17 iterations.
## The RMS GCV score gradient at convergence was 3.618495e-08 .
## The Hessian was not positive definite.
## Model rank =  104 / 104 
## 
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k’.
## 
##                             k’   edf k-index p-value    
## s(case_rate)              9.00  4.95    0.51  <2e-16 ***
## s(hospitalized_rate)     19.00 16.21    1.03    0.65    
## s(emp_incq1)             19.00 18.33    0.76  <2e-16 ***
## s(emp_incq2)             19.00 17.58    0.80  <2e-16 ***
## s(emp_incq3)             19.00 15.07    0.82  <2e-16 ***
## s(emp_incq4)              9.00  7.52    0.81  <2e-16 ***
## s(spend_remoteservices)  9.00  1.00    0.97    0.21    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
# Parameter (linear) estimates and then Smooth Terms (H0: no relationship)
gam_mod %>% pluck('fit') %>% summary()
```

```
## 
## Family: gaussian 
## Link function: identity 
## 
## Formula:
## gps_away_from_home ~ s(case_rate) + s(hospitalized_rate, k = 20) + 
##     s(emp_incq1, k = 20) + s(emp_incq2, k = 20) + s(emp_incq3, 
##     k = 20) + s(emp_incq4) + s(spend_remoteservices)
## 
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.104091   0.000169  -615.9   <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(case_rate)             4.95  5.911 3.199 0.004059 ** 
## s(hospitalized_rate)    16.21 18.004 3.976 7.14e-07 ***
## s(emp_incq1)            18.33 18.799 9.028  < 2e-16 ***
## s(emp_incq2)            17.58 18.313 7.447  < 2e-16 ***
## s(emp_incq3)            15.07 16.728 4.682  < 2e-16 ***
## s(emp_incq4)             7.52  8.058 4.203 0.000407 ***
## s(spend_remoteservices)  1.00  1.001 1.368 0.242846    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## R-sq.(adj) =  0.995   Deviance explained = 99.5%
## GCV = 1.6631e-05  Scale est. = 1.3826e-05  n = 484
```

```
# Looking at possible non-linear functions
gam_mod %>% pluck('fit') %>% plot(all.terms = TRUE, pages = 1)
```



```
formula = gps_away_from_home ~ case_rate + hospitalized_rate + emp_incq1 + emp_incq2 + emp_incq3 + emp_
gam_rec <- recipe(formula, data=minnesota_cut)

gam_rec_new <- gam_rec %>%
     step_ns(case_rate, deg_free = 6) %>%
     step_ns(hospitalized_rate, deg_free = 9) %>%
     step_ns(emp_incq1, deg_free = 8) %>%
     step_ns(emp_incq2, deg_free = 9) %>%
     step_ns(emp_incq3, deg_free = 9) %>%
     step_ns(emp_incq4, deg_free = 7) %>%
     step_ns(spend_remoteservices, deg_free = 6)
```

```
data_cv8 <- minnesota_cut %>%
    vfold_cv(v = 8)

gam_wf <- workflow() %>%
    add_model(lm_spec) %>%
    add_recipe(gam_rec)

fit_resamples(
    gam_wf,
    resamples = data_cv8, # cv folds
    metrics = metric_set(mae,rmse,rsq)
) %>% collect_metrics()
```

```
## Warning: package 'rlang' was built under R version 4.1.2
```

```
## # A tibble: 3 x 6
##   .metric .estimator   mean     n  std_err .config
##   <chr>   <chr>       <dbl> <int>    <dbl> <chr>
## 1 mae     standard   0.0115     8 0.000386 Preprocessor1_Model1
## 2 rmse    standard   0.0145     8 0.000523 Preprocessor1_Model1
## 3 rsq     standard   0.917      8 0.00669  Preprocessor1_Model1
```

```
gam_new_wf <- workflow() %>%
    add_model(lm_spec) %>%
    add_recipe(gam_rec_new)

fit_resamples(
    gam_new_wf,
    resamples = data_cv8, # cv folds
    metrics = metric_set(mae,rmse,rsq)
) %>% collect_metrics()
```

```
## # A tibble: 3 x 6
##   .metric .estimator    mean     n  std_err .config
##   <chr>   <chr>        <dbl> <int>    <dbl> <chr>
## 1 mae     standard   0.00424     8 0.000173 Preprocessor1_Model1
## 2 rmse    standard   0.00555     8 0.000198 Preprocessor1_Model1
## 3 rsq     standard   0.988       8 0.000842 Preprocessor1_Model1
```