HGAM trial

```
library(tidyverse)
library(readxl)
library(mgcv)
library(MASS)
library(gratia)

# dvs <- c('CHD', 'Depression', 'COPD', 'DIABETES', 'HIGHCHOL', 'BPHIGH',
'Obesity')

data <- read_excel("merged_excel.xlsx") # non-cleaned data
```

Basic data cleaning - not fully implemented:

```
rename(data, PAProp = "Park Area - Proportion", StopsSqMile = "Stops per
Sq Mile")
```

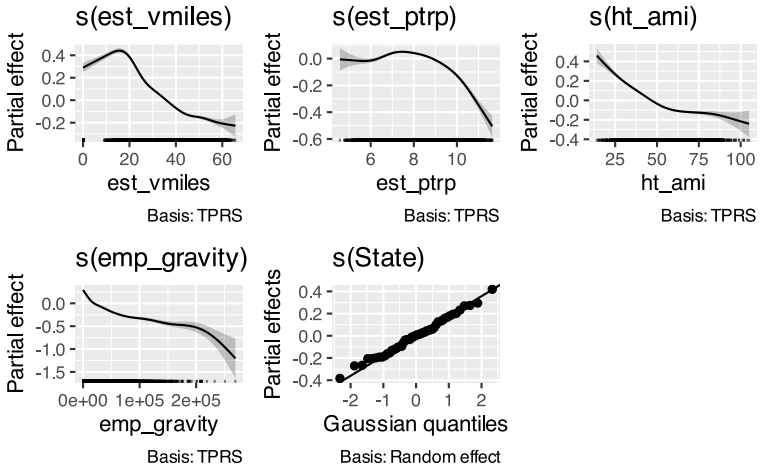
```
# A tibble: 50,080 × 27
      TractID Obesity StopsSqMile    PAProp lapophalfshare lapop1share est_ptrp
        <dbl>   <dbl>       <dbl>     <dbl>          <dbl>       <dbl>    <dbl>
 1 39003010100    33.3           0   2.44e-2           39.8        13.4     8.71
 2 39003010900    38.6           0   0                 55.9         3.14    7.73
 3 39003011000    42.1           0   0                 81.9        30.9     7.67
 4 39003011200    44.8           0   0                 98.2        14.7     8.33
 5 39003011800    38.5           0   9.41e-6           73.7        28.3     8.22
 6 39003011900    39.5           0   2.85e-2           89.5        78.6     8.14
 7 39003012000    37.2           0   0                100.         99.9     8.37
 8 39003012200    46.8           0   0                 48.2        NA       8.07
 9 39003012300    45             0   4.38e-2           42.4        NA       7.97
10 39003012400    46.7           0   0                 25.4        NA       7.54
# ℹ 50,070 more rows
# ℹ 20 more variables: est_vmiles <dbl>, CHD <dbl>, Depression <dbl>,
#   COPD <dbl>, DIABETES <dbl>, HIGHCHOL <dbl>, BPHIGH <dbl>,
#   lakids1share <dbl>, lahunv1share <dbl>, Lat <dbl>, Long <dbl>,
#   ht_ami <dbl>, emp_gravity <dbl>, `Job Access By Transit` <dbl>,
#   State <chr>, Subgroup <dbl>, `Unnamed: 0` <lgl>, `Unnamed: 1` <lgl>,
#   `Unnamed: 2` <lgl>, `Unnamed: 3` <lgl>
```

```
data <- data |>
    dplyr::select(!starts_with("Unnamed")) |> # remove extra columns
    filter(State != "FL") |> # remove Florida data
```

```
    filter(emp_gravity < 300000) # remove emp_gravity single outlier

data$est_ptrp <- na_if(data$est_ptrp, 0) # replace 0 with NA
data$ht_ami <- na_if(data$ht_ami, 0)
data$TractID <- factor(data$TractID) # convert to factor variable
data$State <- factor(data$State)

data$COPD <- data$COPD / 100 # convert to percentage 0-1

data <- drop_na(data) # drop missing data

# Not worrying about train/test split at the moment
```

Single common (global) smoother for all observations (non MLM?)

```
COPD_modG <- bam(COPD ~ s(est_vmiles, bs="tp") + s(est_ptrp, bs="tp") +
s(ht_ami, bs="tp") + s(emp_gravity, bs="tp") + s(State, bs="re"), data=data,
method="fREML", family="quasibinomial")

draw(COPD_modG)
```

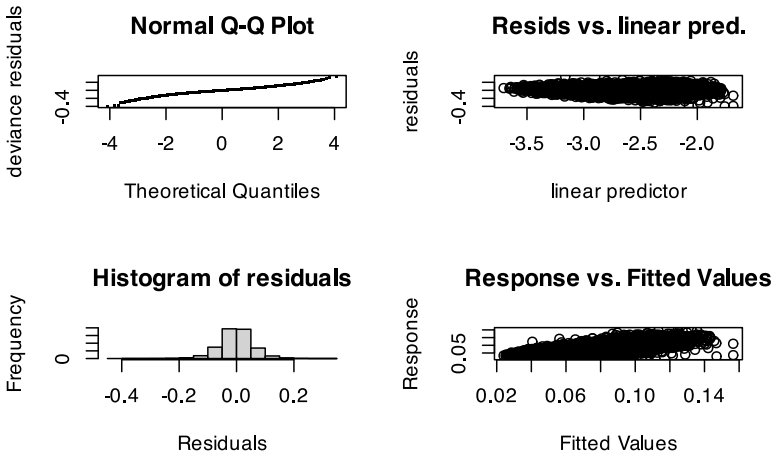



```
summary(COPD_modG)
```

```

Family: quasibinomial
Link function: logit

Formula:
COPD ~ s(est_vmiles, bs = "tp") + s(est_ptrp, bs = "tp") + s(ht_ami,
```

```
    bs = "tp") + s(emp_gravity, bs = "tp") + s(State, bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.73409    0.02395  -114.1   <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(est_vmiles)   8.476  8.908 358.60  <2e-16 ***
s(est_ptrp)     7.027  8.019  93.29  <2e-16 ***
s(ht_ami)       5.830  6.840 224.43  <2e-16 ***
s(emp_gravity)  8.265  8.832 737.18  <2e-16 ***
s(State)       48.268 49.000 146.51  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =   0.65   Deviance explained = 65.6%
fREML = -33134  Scale est. = 0.0031695  n = 22855
```

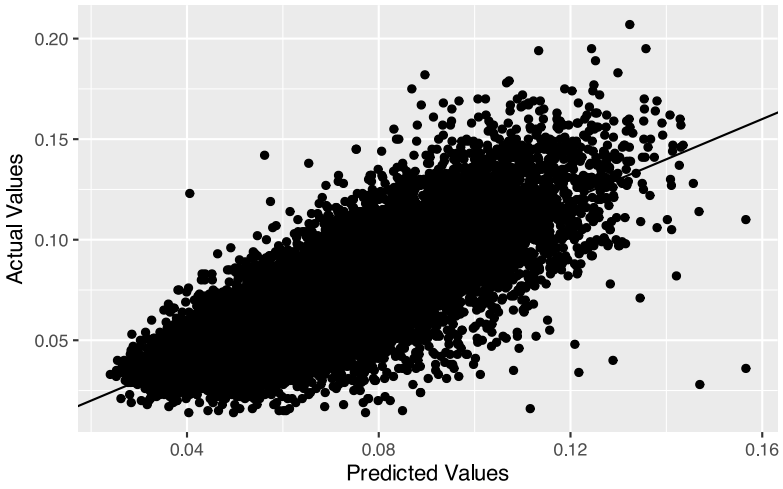
```
gam.check(COPD_modG)
```




```
Method: fREML   Optimizer: perf newton
full convergence after 6 iterations.
Gradient range [-0.0003888438,0.0001166878]
(score -33133.89 & scale 0.003169548).
Hessian positive definite, eigenvalue range [1.808958,11425.06].
Model rank =  87 / 87
```

```
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(est_vmiles)   9.00  8.48    0.98    0.04 *
s(est_ptrp)     9.00  7.03    0.98    0.12
s(ht_ami)       9.00  5.83    1.00    0.53
s(emp_gravity)  9.00  8.27    1.00    0.52
s(State)       50.00 48.27      NA      NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pred_modG <- predict(COPD_modG, data, type="response", se.fit=TRUE)

ggplot(data, aes(x=pred_modG$fit, y=COPD)) +
    geom_point() +
    geom_abline(intercept = 0, slope = 1) +
    labs(x='Predicted Values', y='Actual Values')
```

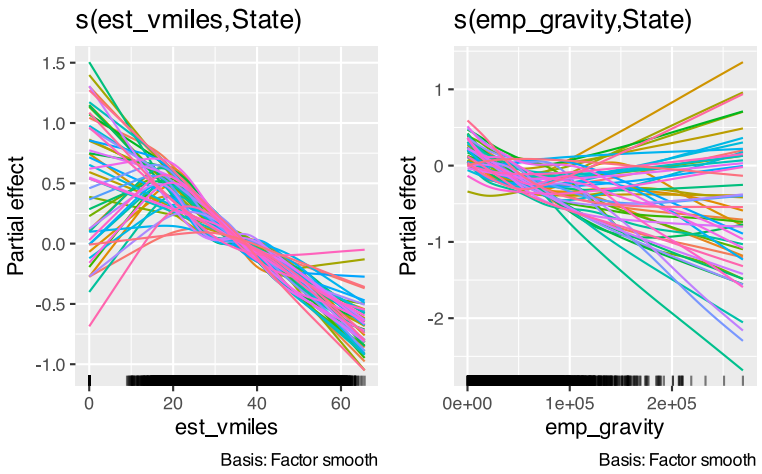


A single common smoother plus group-level smoothers that have the same wiggliness (model GS)

```
COPD_modGS <- bam(COPD ~ s(est_vmiles, State, bs="fs") + s(emp_gravity, State,
bs="fs"), data=data, method="fREML", family="quasibinomial")
```

```
Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has repeated
1-d smooths of same variable.
```

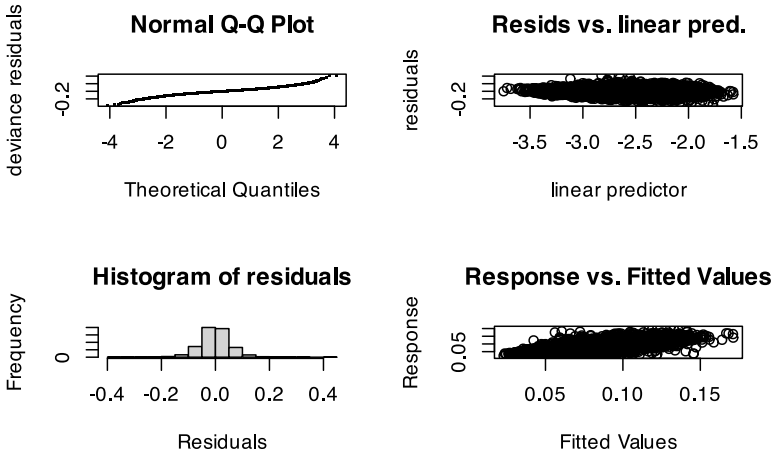
```
draw(COPD_modGS)
```




```
summary(COPD_modGS)
```

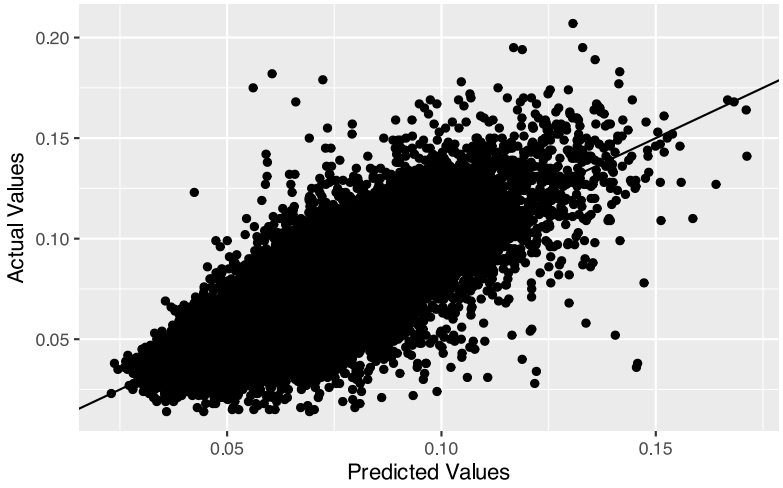
```
Family: quasibinomial
Link function: logit

Formula:
COPD ~ s(est_vmiles, State, bs = "fs") + s(emp_gravity, State,
    bs = "fs")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.71545    0.02136  -127.1   <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(est_vmiles,State)  229.4    499 45.1  <2e-16 ***
s(emp_gravity,State) 199.0    476 27.0  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.653   Deviance explained = 66.4%
fREML = -32638  Scale est. = 0.0031627  n = 22855
```

```
gam.check(COPD_modGS)
```




```
Method: fREML   Optimizer: perf newton
full convergence after 14 iterations.
Gradient range [-0.0001274781,0.0001395328]
(score -32638.03 & scale 0.003162721).
Hessian positive definite, eigenvalue range [5.232851e-05,11428.13].
Model rank =  1001 / 1001

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(est_vmiles,State)  500 229       1    0.50
s(emp_gravity,State) 500 199       1    0.53
```

```
pred_modGS <- predict(COPD_modGS, data, type="response", se.fit=TRUE)

ggplot(data, aes(x=pred_modGS$fit, y=COPD)) +
    geom_point() +
    geom_abline(intercept = 0, slope = 1) +
    labs(x='Predicted Values', y='Actual Values')
```


0.20
0.15
0.10
0.05
Actual Values
0.05
0.10
0.15
Predicted Values