

# STA723 Case Study - Group 1

Melody Jiang, Irene Ji, Keru Wu

1/20/2020

## Abstract

This report evaluates association between exposures Dichlorodiphenyldichloroethylene (DDE) & Polychlorinated Biphenyls (PCBs) and the risk of preterm birth, using a subset of data from the National Collaborative Perinatal Project(CPP). We adopted the Generalized Additive Model (GAM) as well as its Bayesian counterpart for analysis. Our approach successfully captured the nonlinearity between interested variables and risk of premature delivery, and it also aligns with known conclusions in epidemiology. Our results demonstrate that exposure to higher concentrations of DDE and PCBs are associated higher risk of preterm birth.

## 1. Introduction

The dataset taken from CPP includes 2380 samples of women and their children which was studied by Longnecker et al. (2001). It is of interest to study how chemical exposures relate to pregnancy since abnormalities in pregnancy such as preterm birth might cause unfavorable developmental outcomes in children. Preterm birth is deliveries occurring earlier than 37 weeks of gestational age. DDE and PCBs are of primary interest among all exposures, both of which are breakdown products in the body of chemicals. These chemicals build up fatty deposits and exist universally in our body, potentially impacting our health. The dataset also contains other possible confounding variables such as cholesterol, triglyceridesage, maternal age, race and smoking status. The goal of our analysis is to assess how DDE and PCBs relate the risk of preterm birth, controlling for counfounding variables.

Our findings demonstrate that there is positive association between two interested exposures and the risk of preterm birth. Higher concentrations of DDE and PCBs are related to higher risk of premature delivery. Although various PCBs show diffrent significance in the relationship, generally they all contribute to higher risk. Some noticable PCBs are pcb\_074, pcb\_105, pcb\_118, pcb\_153 and pcb\_170. Another finding of our analysis is that when concentration of these exposures is below a threshold, both exposures have little effect, while they have a constant effect when they are above a higher threshold.

## 2. Materials & Methods

After grouping Gestational Age into binary response variable (Preterm Delivery or Non-preterm Delivery), in order to allow for non-linear dose-response relationship, we applied Generalized Additive Model (GAM) to fit the data.

Logit transformation is applied to the binary response variable. Numeric covariates, including DDE, Pricipal Components (PCs) 1 to 4 of PCBs, Maternal Age, triglycerides level and cholesterol level are

smoothed to allow for non-linear association. Furthermore, we also adjusted for categorical confounding variables by including them in the model as linear covariates, which includes Smoking Status, Race, Center, etc.

We conducted model checking on the fitted GAM model and examined the effect plots of DDE and PCBs. In order to quantify the association for DDE and PCBs (via PCs), we controlled for all other covariates and computed the change in probability of Preterm Delivery at different levels of DDE and PCBs.

However, frequentist approach may overestimate uncertainty and produce a non-significant p-value. In addition, residual plot of previous GAM model indicates that assumptions of frequentist GAM could be unwarranted. To improve performance of our model, we instead use a Bayesian Generalized Additive Model. We add priors on the common regression coefficients and priors on the standard deviations of the smooth terms. We use default settings in *stan\_gamm4* function in R package *rstanarm*, which adopts a weak informative normal prior for all common regression coefficients and standard deviations of the smooth terms.

### 3. Results

#### 3.1 Exploratory Data Analysis and Preprocessing

On handling missing data, we removed albumin from our dataset because around 93% of this variable is missing. There is only one observation missing DDE and PCBs, and we removed this observation. We imputed missing data in the covariates using the *MICE* R package. We examined the correlation among explanatory variables and found high correlation between PCBs, as shown in Figure 1A. Such high correlation might distort modeling result, so we made PCBs into Principal Components (PCs) as introduced in Section 2.

After data pre-processing, we examined the distribution of DDE and first principle component of PCB, and found that for both DDE and PCB, the concentration of chemical is generally higher in observations that has preterm delivery, as shown in Figure 1B and Figure 1C. Among the possible covariates, most notably, there is a nonlinear relationship between age of mother and proportion of early delivery, as shown in Figure 1D.

#### 3.2 Main Results

Using frequentist GAM, we found that there are slightly positive association between chemical exposures and preterm delivery. Among all covariates, DDE, PC1, As shown in Figure 2 in the appendix, as DDE level increases, the log-odds of preterm delivery increases. Increasing PC1 is also associated with increasing log-odds of preterm delivery. The change in probability of preterm delivery at different levels of DDE and PC1 are summarized in Tables 1 & 2 in the Appendix.

As shown in the tables, higher DDE level is associated with higher probability of preterm delivery. But the rate of increment decreases as DDE level increases, which aligns with our presumption of dose-response effect (the effect may be less significant as chemical level increases). As for PC1, there is also positive association between PC1 and the probability of preterm delivery. As PC1 has positive relationship with PCBs, the PCBs also have positive association with preterm delivery. Hence, we conclude that the chemicals (DDE and PCBs) have positive association with preterm delivery.

Results from our Bayesian Generalized Additive Model align with previous results in frequentist approach. In Figure 4, higher concentration of DDE and PCBs is related to higher risks for pregnant women. In addition, both DDE and the first principle component have significant p-values, indicating the importance of these exposures in the relationship with the response. If we further consier use

estimates of Bayesian GAM to check residual assumptions of GAM, Figure 5 verifies the correctness of Bayesian GAM, which outperforms that of frequentist approach. Estimated effects of DDE and PCBs also have narrower credible intervals compared to frequentist ones. Our model capture some flat regions when concentration is relatively low and high.

Increase of 1 unit in DDE leads to approximately an increase of 0.017 in log odds when DDE has the lowest concentration 2.5 ug/dL. And it decrease to 0.015 when DDE reaches the average concentration 30 ug/dL. But when DDE reaches 120 ug/dL, its increase does not impact the log odds of risk (almost constant risk). This flat region result aligns with domain knowledge of epidemiology that chemical effects become stable after reaching a upper bound. We conclude similar results when analyzing the first principle component of PCBs: when PC1 is lower than 0 or higher than 15, Bayesian GAM shows that change in PC1 has almost no effect. But when PC1 is around its mean value 7.5, one unit increase in PC1 results in an increase of 0.1 in log odds. Note that all loadings for the first PC are positive, where pcb\_74, pcb\_105, pcb\_138, etc. have loadings over 0.3. Therefore after transforming back to original scale, flat region still exist for PCBs, and unit increase in one specific PCB can be attained. (e.g. one unit increase in pcb\_138 leads to an increase of 0.66 in log odds when pcb\_138 is at mean level 0.67 ng/dL)

### 3.3 Sensitivity Analysis

Frequentist GAM and Bayesian GAM give confidence intervals and credible intervals for measuring the effects respectively. Generally speaking, Bayesian GAM has narrower intervals and more significant p-values in anova tests. Both models show one common result that when concentrations of DDE & PCBs become higher, uncertainty increase greatly. This is partially due to the fact that we have limited data for higher level concentrations. Refer to attached figures for detailed CI intervals.

In addition, how to deal with the collinearity of PCBs and carry out dimensionality reduction has great influence in our approach. Our adopted PCA approach has the best performance compared to simple sum and factor analysis, which either have low significance for PCBs, or become unwarranted after model check.

## 4. Discussion

This report has analyzed how DDE and PCBs relate to the risk of premature delivery. After preprocessing data (e.g. impute missing data) and dimensionality reduction (PCA), we build up different models for the data and finally adopt the Generalized Additive Model (GAM) and its Bayesian version. Our approach has advantages that it captures the nonlinearity relationship between exposures and outcomes, and it also fits confounding variables. We conclude that higher exposure to DDE and certain PCBs may be associated with higher risk of premature delivery.

The first extension of our approach is to deal with different centers specially. Our model demonstrates that center 15 and center 37 may deviate from others, which genrally collect samples with higher risks. There are other ways which may perform better in dealing with centers. One can adopt a Bayesian hierarchical model which specifies different variances between centers. Another more direct extension of our GAM model is to include mixed effect. We may use Generalized Additive Mixed Model (GAMM) to consider random effect of centers and it can also be applied to other categorical variables like smoking status and race.

After we find out that DDE and PCBs are related to higher risk of preterm birth, we can examine the trend of exposures effects more accurately. Specifying a special prior (e.g. guarantee monotonicity) may benefit from narrower credible intervals compared to frequentist approach and naive bayes approach.

Furthermore, interaction between chemicals also impacts human health outcomes. Collinearity among PCBs indicates the need for a general dimension reduction method or a variable selection approach. Ferrari and Dunson (2019) build up a bayesian factor model designed for interactions. High correlation between exposure levels can be explained in this flexible dimension reduction approach. Another future research direction is to include penalty in GAM for variable selection.

## References

- Ferrari, Federico, and David B. Dunson. “Bayesian Factor Analysis for Inference on Interactions.” arXiv preprint arXiv:1904.11603 (2019).
- Gabry, Jonah, and Ben Goodrich. “rstanarm: Bayesian applied regression modeling via Stan.” R package version 2.1 (2016).
- Hastie, Trevor J. “Generalized additive models.” Statistical models in S. Routledge, 249-307. (2017).

Appendix A. Figures and Tables

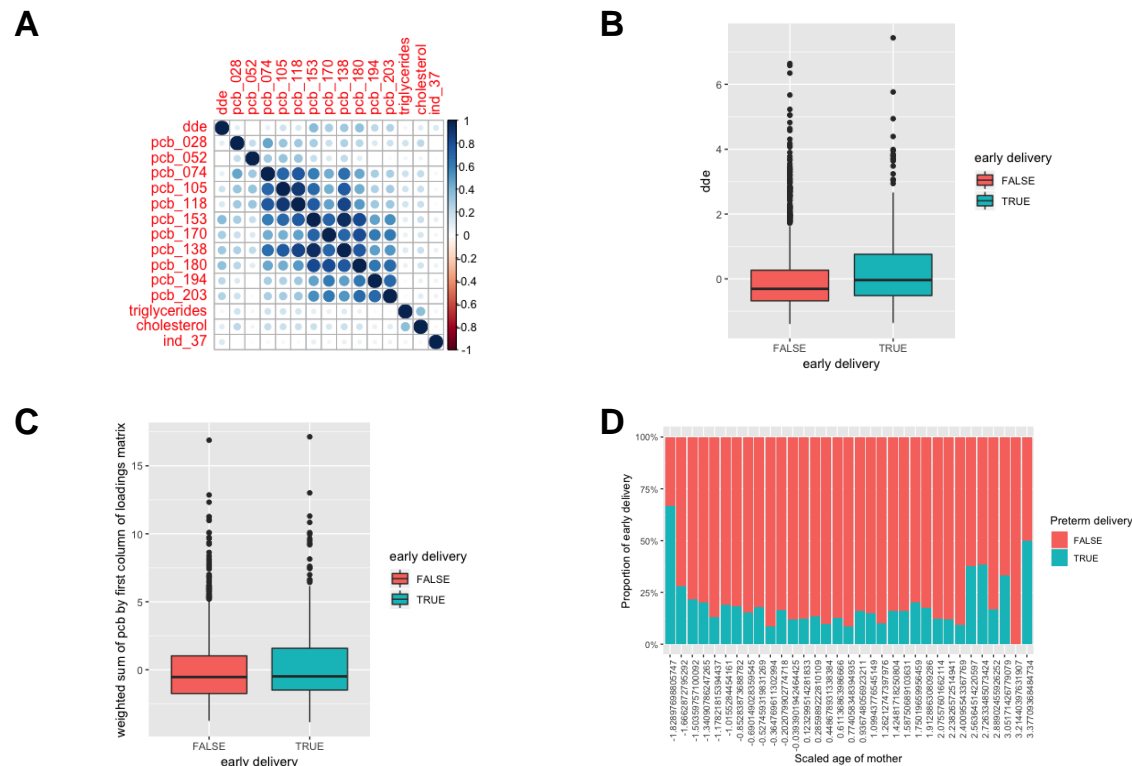


Figure 1. Plots for exploratory data analysis

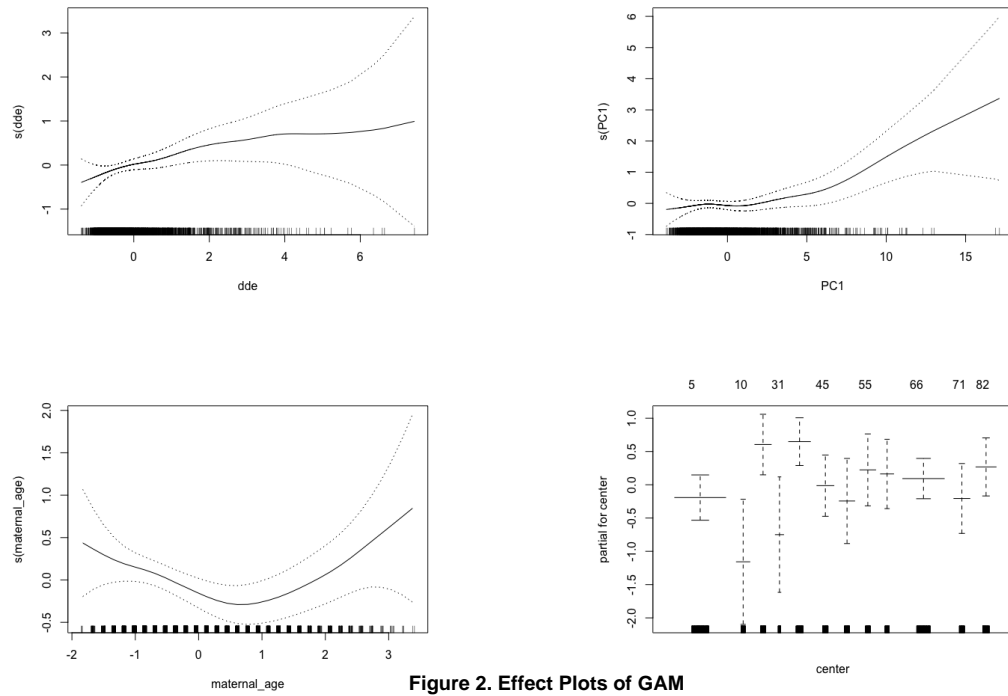


Figure 2. Effect Plots of GAM

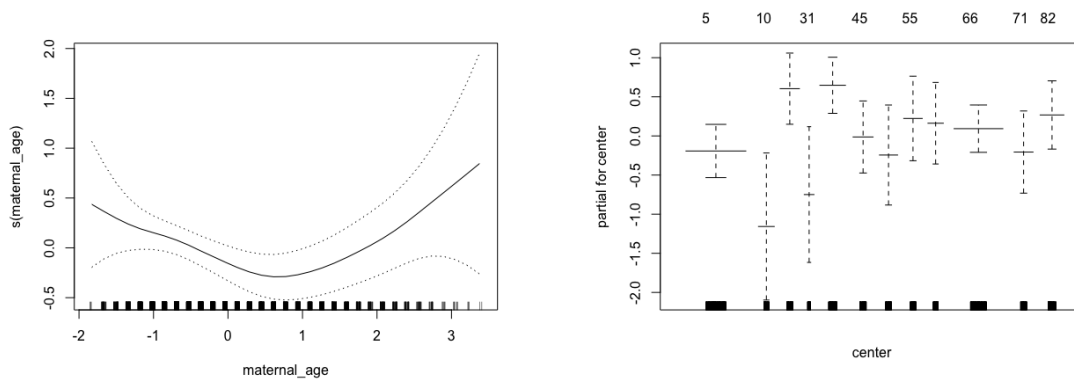


Figure 4 & 5. Effect Plot for Maternal Age and Center

	Scaled DDE	% Change in Probability
1	-1.00	1.43
2	2.00	1.23
3	4.00	0.32

Table 1: Change in Probability of Preterm Delivery at Different DDE Levels

	PC1	% Change in Probability
1	-3.00	0.35
2	3.00	0.66
3	7.00	2.11
4	10.00	4.88

Table 2: Change in Probability of Preterm Delivery at Different PC1 Levels

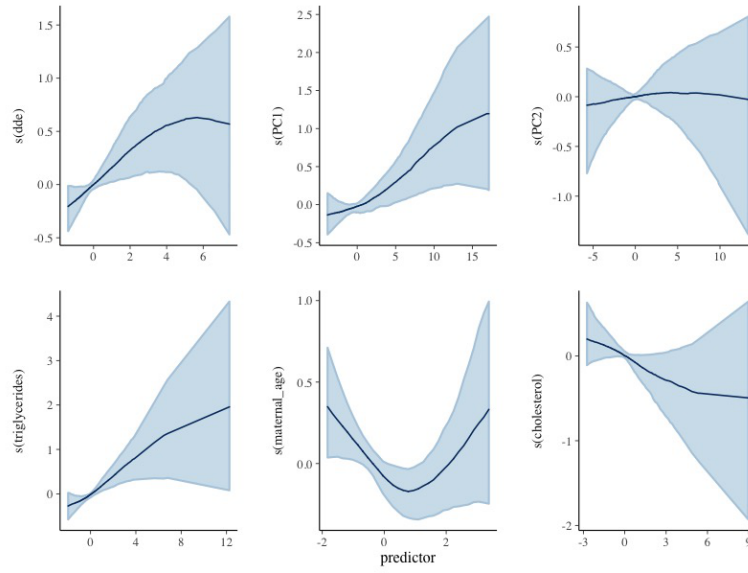
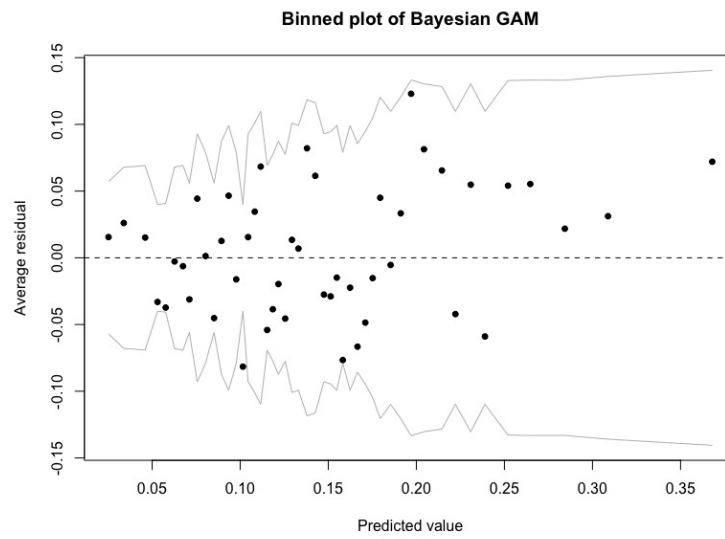


Figure 4. Nonlinear trend of several variables in Bayesian GAM





**Figure 5. Bayesian GAM model check**

## Appendix B. Codes and Outputs

### 1 EDA

#### 1.1

```
dat = readRDS("Longnecker.rds")
dat$center = factor(dat$center)
dat$smoking_status = factor(dat$smoking_status)

dat = dat[-1861,]
library(mice)

## Loading required package: lattice

##
## Attaching package: 'mice'

## The following objects are masked from 'package:base':
##
##      cbind, rbind

dat = dat[,!names(dat) %in% c('albumin')]
imp = mice(dat)

##
## iter imp variable
## 1 1 score_education score_income score_occupation
## 1 2 score_education score_income score_occupation
## 1 3 score_education score_income score_occupation
## 1 4 score_education score_income score_occupation
## 1 5 score_education score_income score_occupation
## 2 1 score_education score_income score_occupation
## 2 2 score_education score_income score_occupation
## 2 3 score_education score_income score_occupation
## 2 4 score_education score_income score_occupation
## 2 5 score_education score_income score_occupation
## 3 1 score_education score_income score_occupation
## 3 2 score_education score_income score_occupation
## 3 3 score_education score_income score_occupation
## 3 4 score_education score_income score_occupation
## 3 5 score_education score_income score_occupation
## 4 1 score_education score_income score_occupation
## 4 2 score_education score_income score_occupation
## 4 3 score_education score_income score_occupation
## 4 4 score_education score_income score_occupation
## 4 5 score_education score_income score_occupation
## 5 1 score_education score_income score_occupation
## 5 2 score_education score_income score_occupation
## 5 3 score_education score_income score_occupation
```

```
## 5 4 score_education score_income score_occupation
## 5 5 score_education score_income score_occupation
```

```
dat = complete(imp)

dat$ind_37 = dat$gestational_age < 37
dat[,c(1:12,13,15,16,17,18,20)] = scale(dat[,c(1:12,13,15,16,17,18,20)])
```

## 2 PCA

### 2.1 PCA results

```
## PCA

pca = princomp(dat[,2:12])
pcb = as.matrix(dat[,2:12])
pcb_pc = pcb %*% pca$loadings
print(pca$loadings)
```

```
##
## Loadings:
##      Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
## pcb_028  0.161  0.243  0.833  0.342  0.154  0.138  0.224
## pcb_052  0.116  0.376  0.223 -0.886
## pcb_074  0.306  0.314      0.189 -0.217 -0.547 -0.580 -0.233      -0.155
## pcb_105  0.320  0.333 -0.208      -0.282  0.243  0.191  0.411 -0.497 -0.350
## pcb_118  0.342  0.306 -0.248      -0.199      0.106 -0.327  0.162 -0.493
## pcb_153  0.376      -0.160      0.332  0.188  0.106 -0.327  0.162 -0.493
## pcb_170  0.325 -0.274      -0.123  0.323 -0.427      0.689  0.193
## pcb_138  0.383      -0.225      0.165  0.117  0.121 -0.213  0.414
## pcb_180  0.344 -0.277      0.375      -0.259 -0.676  0.361
## pcb_194  0.253 -0.419  0.158 -0.100 -0.585 -0.292  0.494 -0.220
## pcb_203  0.268 -0.409  0.203 -0.106 -0.290  0.546 -0.547  0.120  0.114
##      Comp.11
## pcb_028
## pcb_052
## pcb_074
## pcb_105  0.154
## pcb_118 -0.383
## pcb_153 -0.544
## pcb_170
## pcb_138  0.723
## pcb_180
## pcb_194
## pcb_203
##
##      Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## SS loadings    1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var 0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091
## Cumulative Var 0.091  0.182  0.273  0.364  0.455  0.545  0.636  0.727  0.818
```

```
##               Comp.10 Comp.11
## SS loadings      1.000   1.000
## Proportion Var   0.091   0.091
## Cumulative Var   0.909   1.000
```

```
summary(pca)
```

```
## Importance of components:
##               Comp.1   Comp.2   Comp.3   Comp.4   Comp.5
## Standard deviation 2.4458646 1.3261098 0.94105657 0.89065865 0.70742028
## Proportion of Variance 0.5440699 0.1599370 0.08054181 0.07214604 0.04551399
## Cumulative Proportion 0.5440699 0.7040069 0.78454872 0.85669476 0.90220875
##               Comp.6   Comp.7   Comp.8   Comp.9   Comp.10
## Standard deviation 0.58369249 0.52378706 0.46772252 0.35696738 0.284825079
## Proportion of Variance 0.03098547 0.02495166 0.01989603 0.01158903 0.007378131
## Cumulative Proportion 0.93319422 0.95814588 0.97804191 0.98963094 0.997009072
##               Comp.11
## Standard deviation 0.181346021
## Proportion of Variance 0.002990928
## Cumulative Proportion 1.000000000
```

```
dat$PC1 = pcb_pc[,1]
dat$PC2 = pcb_pc[,2]
dat$PC3 = pcb_pc[,3]
dat$PC4 = pcb_pc[,4]
```

### 3 GAM model

#### 3.1 Fit GAM model

```
library(gam)
```

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

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

```
## Loaded gam 1.16.1
```

```
ga1 = gam(ind_37 ~ s(dde) + s(PC1) + s(PC2) + s(PC3) + s(PC4) +
            s(triglycerides) + score_education + score_income + score_occupation +
            s(maternal_age) + s(cholesterol) + smoking_status + center + race,
          family = binomial(link = 'logit'), data = dat)
```

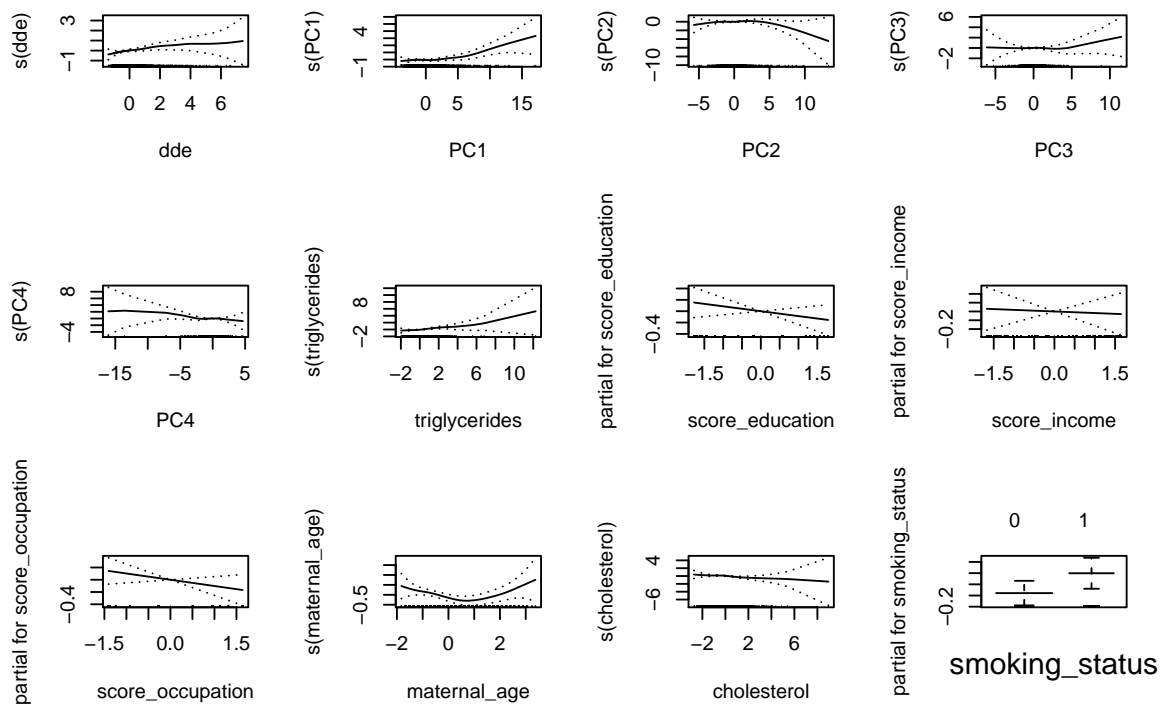
```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
## ignored
```

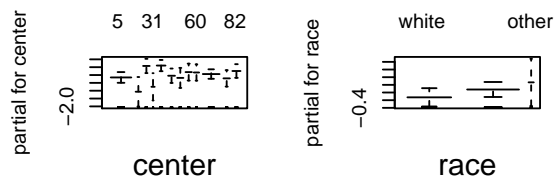
```
summary(ga1)
```

```
##
## Call: gam(formula = ind_37 ~ s(dde) + s(PC1) + s(PC2) + s(PC3) + s(PC4) +
##   s(triglycerides) + score_education + score_income + score_occupation +
##   s(maternal_age) + s(cholesterol) + smoking_status + center +
##   race, family = binomial(link = "logit"), data = dat)
## Deviance Residuals:
##   Min       1Q   Median       3Q      Max
## -1.4841 -0.6158 -0.4749 -0.3401  2.7979
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##   Null Deviance: 2025.59 on 2378 degrees of freedom
## Residual Deviance: 1860.921 on 2329 degrees of freedom
## AIC: 1960.921
##
## Number of Local Scoring Iterations: 8
##
## Anova for Parametric Effects
##           Df Sum Sq Mean Sq F value    Pr(>F)
## s(dde)         1    30.99  30.9911  30.3010 4.105e-08 ***
## s(PC1)         1     4.25   4.2513   4.1567 0.0415837 *
## s(PC2)         1     1.92   1.9170   1.8743 0.1711168
## s(PC3)         1     0.04   0.0373   0.0364 0.8486386
## s(PC4)         1     2.11   2.1116   2.0646 0.1508913
## s(triglycerides) 1     4.12   4.1192   4.0275 0.0448789 *
## score_education 1    12.84  12.8448  12.5588 0.0004021 ***
## score_income    1     3.70   3.6989   3.6165 0.0573323 .
## score_occupation 1     6.04   6.0380   5.9035 0.0151858 *
## s(maternal_age)  1     2.36   2.3582   2.3057 0.1290344
## s(cholesterol)   1     9.58   9.5777   9.3644 0.0022376 **
## smoking_status   1     0.71   0.7076   0.6918 0.4056281
## center          11    32.77   2.9793   2.9129 0.0007957 ***
## race            2     1.50   0.7475   0.7309 0.4816037
## Residuals      2329 2382.04   1.0228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##           Npar Df Npar Chisq  P(Chi)
## (Intercept)
## s(dde)         3     1.8077 0.61328
## s(PC1)         3     9.3846 0.02459 *
## s(PC2)         3     6.9102 0.07481 .
## s(PC3)         3     2.4862 0.47773
## s(PC4)         3     5.2005 0.15769
## s(triglycerides) 3     2.8513 0.41517
## score_education
## score_income
## score_occupation
```

```
## s(maternal_age)          3      10.7705 0.01304 *
## s(cholesterol)           3       2.6608 0.44693
## smoking_status
## center
## race
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow = c(3,4))
plot(gal, se = TRUE)
```





### 3.2 Effect of DDE and PC1

```
# Extract DDE effects
new_data <- new_data_add <- dat[1,]
dde_effect <- function(dde_test){
  new_data$dde <- dde_test
  pred_orig <- predict(ga1, new_data, type = "response")
  # response: prob, link: log odds
  new_data_add$dde <- dde_test + 0.001
  pred_add <- predict(ga1, new_data_add, type = "response")
  return ((pred_add - pred_orig) / 0.001)
}

summary(dat$dde)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.3923 -0.6581 -0.2760  0.0000  0.3177  7.4362
```

```
Scale_DDE <- c(-1,2,4,6)
Percent_Prob_Change <- c(dde_effect(-1)*100,
                         dde_effect(2)*100,
                         dde_effect(4)*100,
                         dde_effect(6)*100)
df <- data.frame(Scale_DDE, Percent_Prob_Change)
```

```
library(knitr)
kable(df)
```

Scale_DDE	Percent_Prob_Change
-1	1.4226332
2	1.2057099
4	0.3301036
6	0.8578243

```
# Extract effects of PC1
new_data <- new_data_add <- dat[1,]
PC1_effect <- function(PC1_test){
  new_data$PC1 <- PC1_test
  pred_orig <- predict(ga1, new_data, type = "response")
  # response: prob, link: log odds
  new_data_add$PC1 <- PC1_test + 0.001
  pred_add <- predict(ga1, new_data_add, type = "response")
  return ((pred_add-pred_orig) / 0.001)
}

summary(dat$PC1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.8573 -1.7107 -0.5345  0.0000  1.0931 17.1345
```

```
PC1_effect(-3)
```

```
##           1
## 0.003364488
```

```
PC1_effect(3)
```

```
##           1
## 0.006481021
```

```
PC1_effect(7)
```

```
##           1
## 0.02045798
```

```
PC1_effect(10)
```

```
##           1
## 0.04761993
```



```
PC1 <- c(-3,3,7,10)
Percent_Prob_Change <- c(PC1_effect(-3)*100,
                          PC1_effect(3)*100,
                          PC1_effect(7)*100,
                          PC1_effect(10)*100)
df <- data.frame(PC1, Percent_Prob_Change)

library(knitr)
kable(df)
```

PC1	Percent_Prob_Change
-3	0.3364488
3	0.6481021
7	2.0457979
10	4.7619934

## 4 Bayesian GAM

Bayesian Generalized Additive Model

$$g(Y_i) = \beta_0 + \sum_{j=1}^m f_j(x_{ij}) + \sum_{k=1}^l \beta_k z_{ik}$$

We add priors on the common regression coefficients, priors on the standard deviations of the smooth terms. The priors are set by default in *rstanarm* package, which is a weak informative normal prior.

### 4.1 Model results

```
library(rstanarm)
```

```
## Loading required package: Rcpp

## rstanarm (Version 2.19.2, packaged: 2019-10-01 20:20:33 UTC)

## - Do not expect the default priors to remain the same in future rstanarm versions.

## Thus, R scripts should specify priors explicitly, even if they are just the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

## options(mc.cores = parallel::detectCores())

## - bayesplot theme set to bayesplot::theme_default()

## * Does _not_ affect other ggplot2 plots

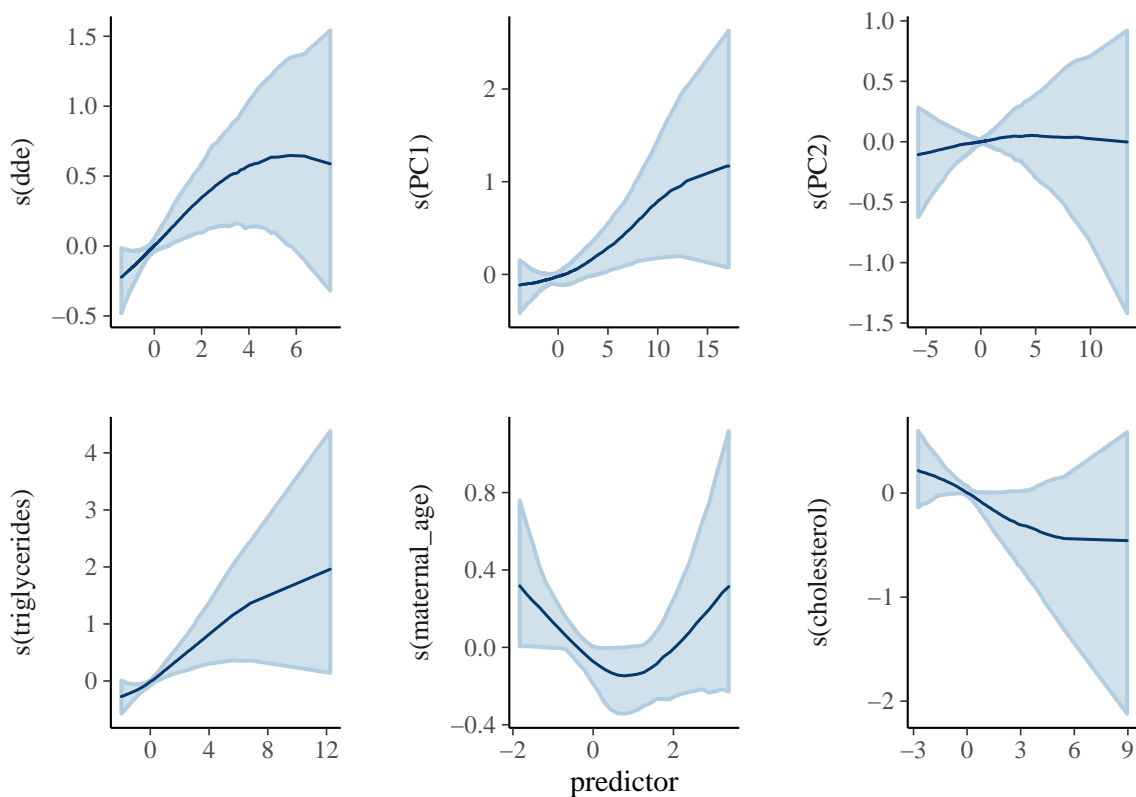
## * See ?bayesplot_theme_set for details on theme setting
```

```
b_ga = stan_gamm4(ind_37 ~ s(dde) + s(PC1) + s(PC2) +
  s(triglycerides) + race + score_education + score_income +
  score_occupation + s(maternal_age) + smoking_status +
  s(cholesterol) + center,
  family = binomial(link = 'logit'), data = dat,
  chain = 1, iter=1000)
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000243 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.43 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration: 1 / 1000 [ 0%] (Warmup)
## Chain 1: Iteration: 100 / 1000 [ 10%] (Warmup)
## Chain 1: Iteration: 200 / 1000 [ 20%] (Warmup)
## Chain 1: Iteration: 300 / 1000 [ 30%] (Warmup)
## Chain 1: Iteration: 400 / 1000 [ 40%] (Warmup)
## Chain 1: Iteration: 500 / 1000 [ 50%] (Warmup)
## Chain 1: Iteration: 501 / 1000 [ 50%] (Sampling)
## Chain 1: Iteration: 600 / 1000 [ 60%] (Sampling)
## Chain 1: Iteration: 700 / 1000 [ 70%] (Sampling)
## Chain 1: Iteration: 800 / 1000 [ 80%] (Sampling)
## Chain 1: Iteration: 900 / 1000 [ 90%] (Sampling)
## Chain 1: Iteration: 1000 / 1000 [100%] (Sampling)
## Chain 1:
## Chain 1: Elapsed Time: 7.34141 seconds (Warm-up)
## Chain 1: 5.71016 seconds (Sampling)
## Chain 1: 13.0516 seconds (Total)
## Chain 1:

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quantiles may be unreliable.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
plot_nonlinear(b_ga)
```



## 4.2 Model check

```
library(arm)
```

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

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

```
## Loading required package: lme4
```

```
##
```

```
## arm (Version 1.10-1, built: 2018-4-12)
```

```
## Working directory is /Users/yiji/Desktop/Duke Statistics PhD/2020 Spring Courses/STA 723 Case Study
```

```
pred.probs_gam <- predict(b_ga, dat, type = 'response') # Calculate predicted probabilities
resid_gam <- residuals(b_ga) # residuals
```

```
binplot(x = pred.probs_gam, y = resid_gam, nclass=NULL,
        xlab="Predicted value", ylab="Average residual",
        main="Binned residual plot: residual vs estimated probabilities for gam",
        cex.pts=0.8, col.pts="gray")
```

**Binned residual plot: residual vs estimated probabilities for gam**

