Appendix for STA723 Case Study - Group 1

Melody Jiang, Irene Ji, Keru Wu

1/22/2020

Contents

This appendix mainly contains codes and additional 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)
```

```
## Warning: package 'mice' was built under R version 3.5.2
```

```
## 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
## 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
## pcb_105  0.320  0.333 -0.208        -0.282  0.243  0.191  0.411 -0.497
## pcb_118  0.342  0.306 -0.248        -0.199                        0.202
## pcb_153  0.376        -0.160         0.332  0.188  0.106 -0.327  0.162
## 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
## 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.10 Comp.11
```

```
## pcb_028
## pcb_052
## pcb_074 -0.155
## pcb_105 -0.350   0.154
## pcb_118  0.686  -0.383
## pcb_153 -0.493  -0.544
## pcb_170
## pcb_138          0.723
## pcb_180  0.361
## pcb_194
## pcb_203
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## SS loadings     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
## Cumulative Var  0.091  0.182  0.273  0.364  0.455  0.545  0.636  0.727
##                Comp.9 Comp.10 Comp.11
## SS loadings     1.000   1.000   1.000
## Proportion Var  0.091   0.091   0.091
## Cumulative Var  0.818   0.909   1.000
```

```
summary(pca)
```

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

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)
```

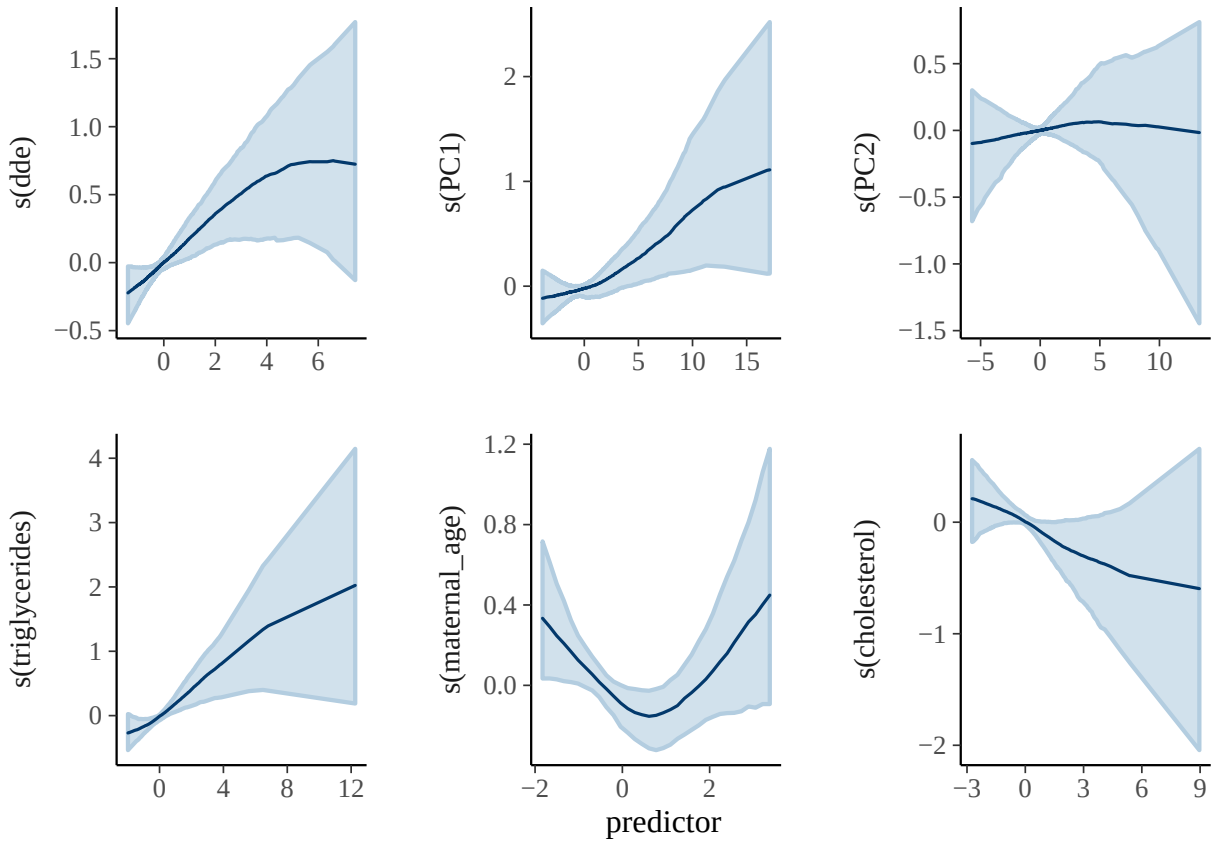
```
## Warning: package 'rstanarm' was built under R version 3.5.2
## Loading required package: Rcpp
## Warning: package 'Rcpp' was built under R version 3.5.2
## rstanarm (Version 2.19.2, packaged: 2019-10-01 20:20:33 UTC)
## - Do not expect the default priors to remain the same in future rstanarm versions.
## 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 0.000203 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.03 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: 6.27895 seconds (Warm-up)
## Chain 1:                2.73482 seconds (Sampling)
## Chain 1:                9.01377 seconds (Total)
## Chain 1:
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.95 may help
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
plot_nonlinear(b_ga)
```



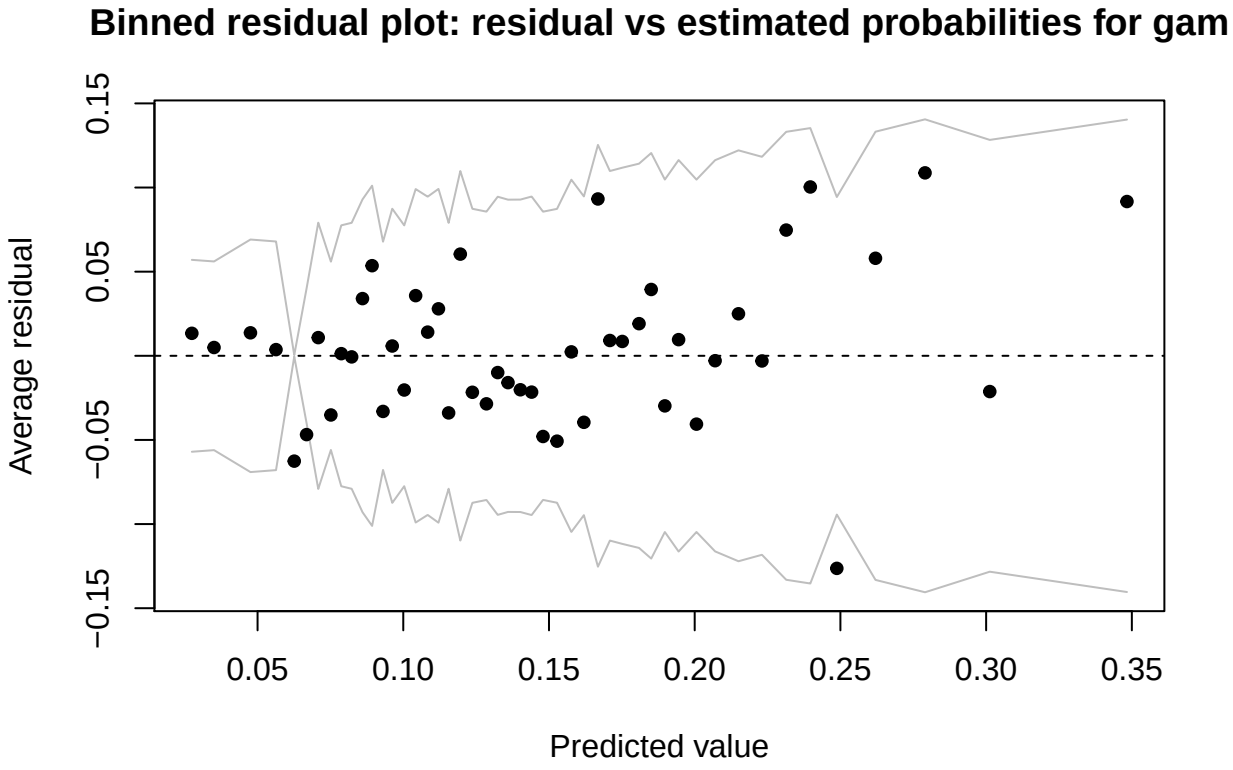
4.2 Model check

```
library(arm)
```

```
## Loading required package: MASS
## Loading required package: Matrix
## Warning: package 'Matrix' was built under R version 3.5.2
## Loading required package: lme4
##
## arm (Version 1.10-1, built: 2018-4-12)
## Working directory is /Users/true/Documents/Files/Duke/STA723/Projects/1/Appendix
```

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

binnedplot(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=1, col.int="gray")
```


Binned residual plot: residual vs estimated probabilities for gam
Average residual
0.15
0.05
−0.05
−0.15
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Predicted value