

Basic CAR Model

Alvin Sheng

```
library(here)

## Warning in readLines(f, n): line 1 appears to contain an embedded nul
## Warning in readLines(f, n): incomplete final line found on '/Volumes/
## ALVINDRIVE2/flood-risk-health-effects/._flood-risk-health-effects.Rproj'
## here() starts at /Volumes/ALVINDRIVE2/flood-risk-health-effects

library(coda)
library(CARBayes)

## Loading required package: MASS
## Loading required package: Rcpp
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
library(ggplot2)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v tibble  3.1.6     v dplyr    1.0.7
## v tidyr   1.1.4     v stringr  1.4.0
## v readr    2.1.1     vforcats  0.5.1
## v purrr   0.3.4

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
## x dplyr::select() masks MASS::select()

i_am("reports/basic_CAR_model_all_census_tract.Rmd")

## here() starts at /Volumes/ALVINDRIVE2/flood-risk-health-effects
fhs_model_df <- readRDS("intermediary_data/fhs_model_df_all_census_tract_pc.rds")
```

CAR model results, Coronary Heart Disease

Inference is based on 3 markov chains, each of which has been run for 110000 samples, the first 10000 of which has been removed for burn-in. The remaining 100000 samples are thinned by 2, resulting in 150000 samples for inference across the 3 Markov chains.

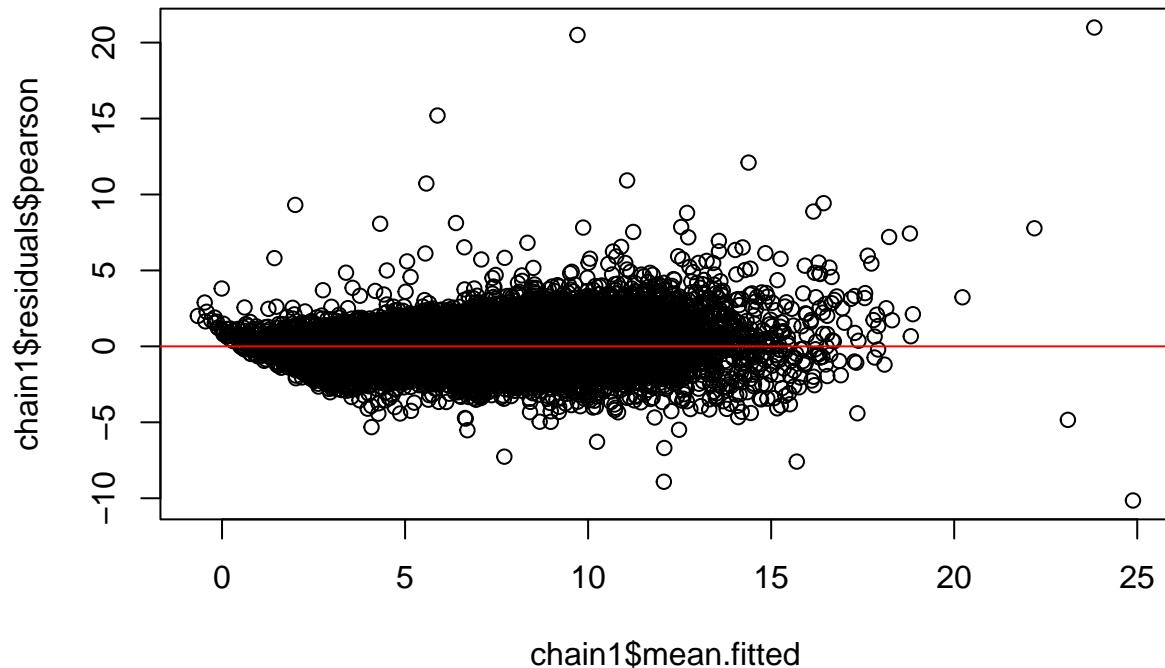
```
load(here("modeling_files/all_census_tract_intrinsic.RData"))
```

Model Diagnostics

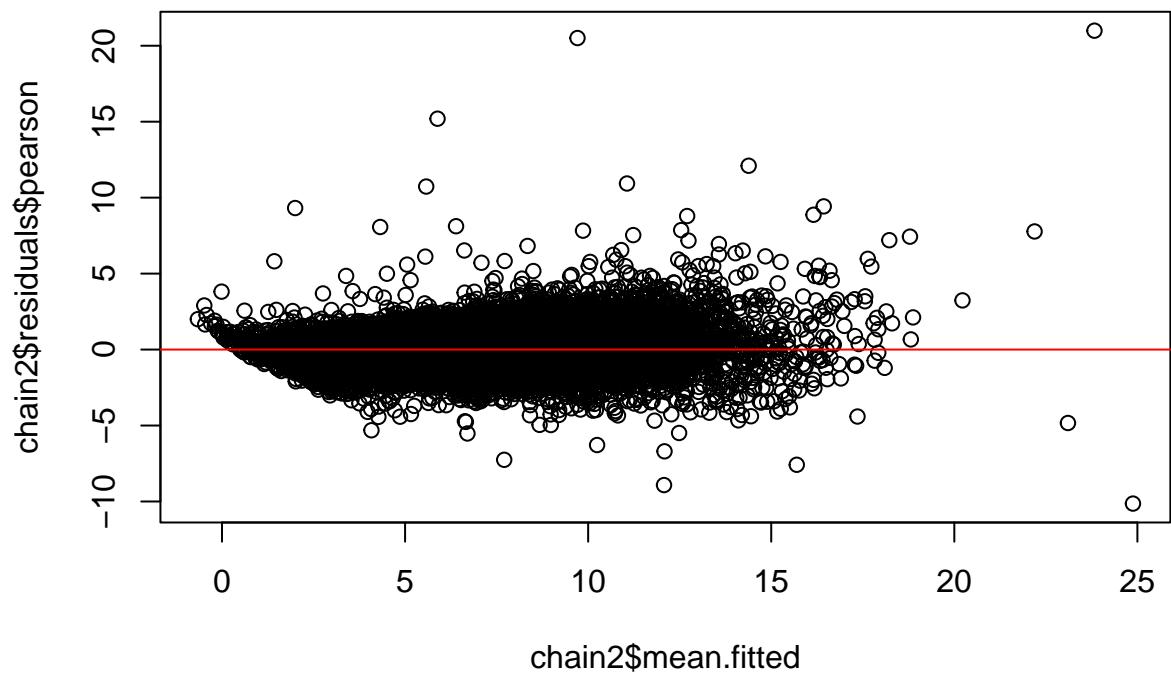
Residual Plots

For each chain

```
plot(chain1$mean.fitted, chain1$residuals$pearson)  
abline(h = 0, col = "red")
```

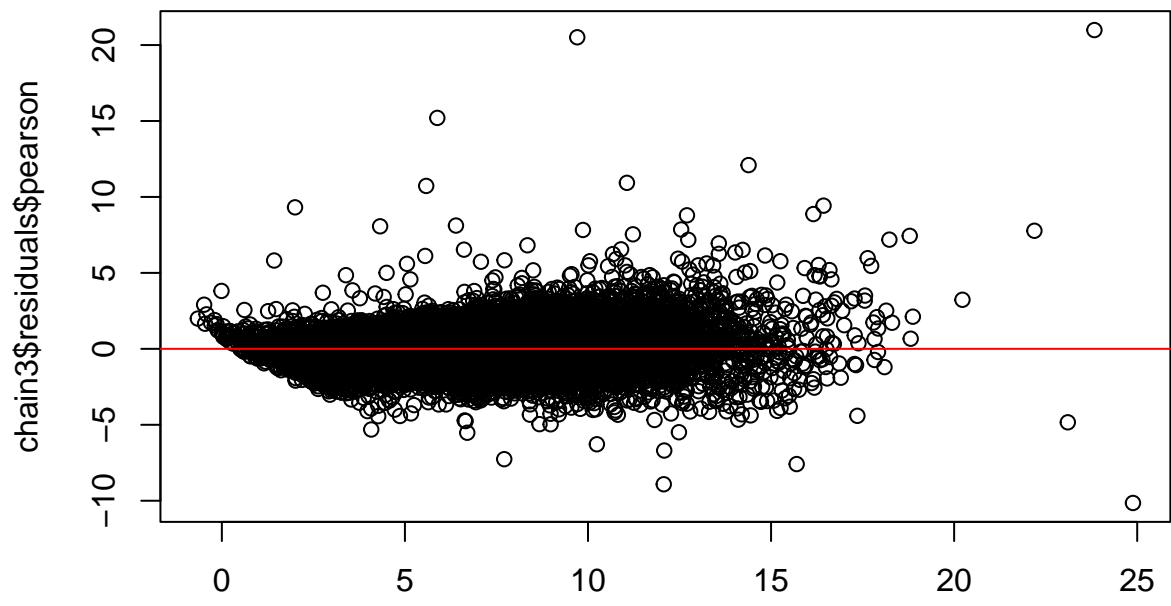


```
plot(chain2$mean.fitted, chain2$residuals$pearson)  
abline(h = 0, col = "red")
```



```
plot(chain2$mean.fitted, chain2$residuals$pearson)
```

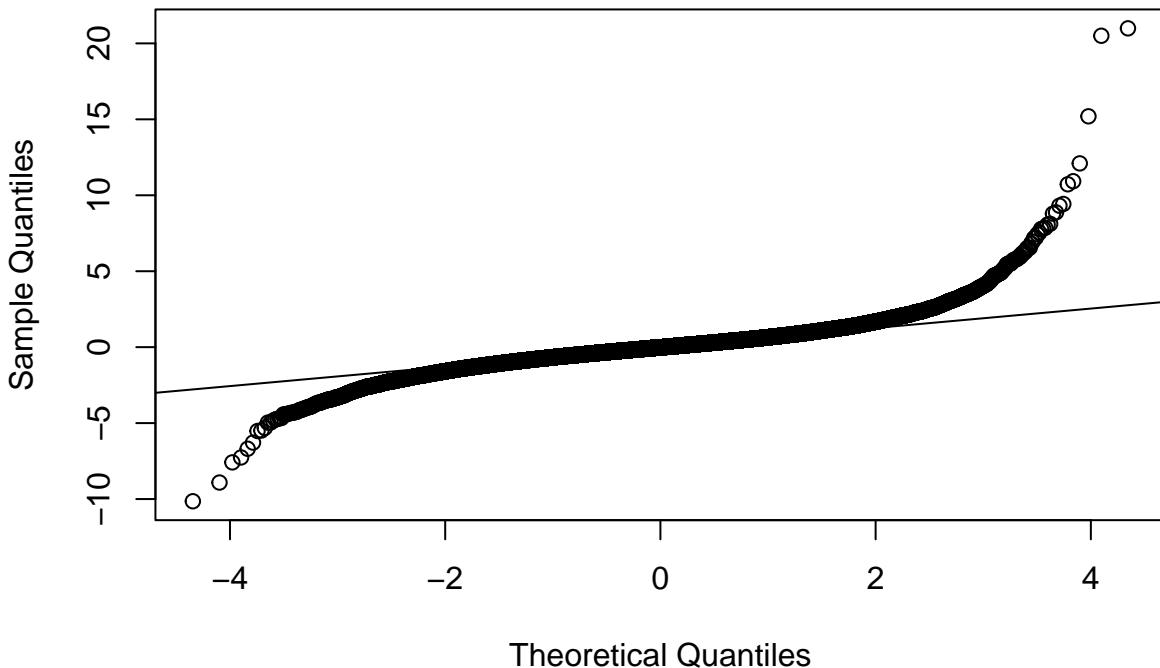
```
abline(h = 0, col = "red")
```



```
qqnorm(chain1$residuals$pearson)
```

```
qqline(chain1$residuals$pearson)
```

Normal Q-Q Plot



What's the proportion of predicted values that correspond to negative counts? Which would not make sense

```
mean(chain1$mean.fitted < 0, na.rm = T)
```

```
## [1] 0.0001654328
```

Small proportion seems reasonable.

Beta samples

```
beta_samples <- mcmc.list(chain1$samples$beta, chain2$samples$beta,
                           chain3$samples$beta)

effectiveSize(beta_samples)

##      var1      var2      var3      var4      var5      var6
## 125499.7838 27191.4912 27917.6198 39362.1706 65711.0338 55823.3470
##      var7      var8      var9      var10     var11     var12
## 44122.5743 73285.2857 30773.3297 41756.4302 35046.8267 49915.2278
##      var13     var14     var15     var16     var17     var18
## 48128.8828 73620.2014 23030.0734 37202.4283 38419.5744 32228.9658
##      var19     var20     var21     var22     var23     var24
## 53846.8401 28622.0060 73382.2380 40466.2562 8376.6144 4583.9169
##      var25     var26     var27     var28     var29     var30
## 325.6080 3749.4412 2028.5671 2659.8994 745.1913 120.0038
##      var31     var32     var33
## 804.8586 1203.0334 20687.3398

plot(beta_samples)
```

```

gelman.diag(beta_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]    1.00    1.00
## [2,]    1.00    1.00
## [3,]    1.00    1.00
## [4,]    1.00    1.00
## [5,]    1.00    1.00
## [6,]    1.00    1.00
## [7,]    1.00    1.00
## [8,]    1.00    1.00
## [9,]    1.00    1.00
## [10,]   1.00    1.00
## [11,]   1.00    1.00
## [12,]   1.00    1.00
## [13,]   1.00    1.00
## [14,]   1.00    1.00
## [15,]   1.00    1.00
## [16,]   1.00    1.00
## [17,]   1.00    1.00
## [18,]   1.00    1.00
## [19,]   1.00    1.00
## [20,]   1.00    1.00
## [21,]   1.00    1.00
## [22,]   1.00    1.00
## [23,]   1.00    1.01
## [24,]   1.00    1.00
## [25,]   1.02    1.07
## [26,]   1.00    1.01
## [27,]   1.01    1.05
## [28,]   1.00    1.01
## [29,]   1.03    1.09
## [30,]   1.09    1.27
## [31,]   1.00    1.00
## [32,]   1.00    1.01
## [33,]   1.00    1.00
##
## Multivariate psrf
##
## 1.08

```

Examining sigma2, nu2, rho

```

sigma2_samples <- mcmc.list(chain1$samples$sigma2, chain2$samples$sigma2,
                             chain3$samples$sigma2)

nu2_samples <- mcmc.list(chain1$samples$nu2, chain2$samples$nu2,
                         chain3$samples$nu2)

plot(sigma2_samples)

```

```

plot(nu2_samples)

gelman.diag(sigma2_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1
gelman.diag(nu2_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1

```

Examining a sample of the 3108 phi parameters

```

phi_samples <- mcmc.list(chain1$samples$phi, chain2$samples$phi, chain3$samples$phi)

set.seed(1157, kind = "Mersenne-Twister", normal.kind = "Inversion", sample.kind = "Rejection")

phi_subset_idx <- sample(1:ncol(phi_samples[[1]]), size = 10)

phi_samples_subset <- phi_samples[, phi_subset_idx]

plot(phi_samples_subset)

gelman.diag(phi_samples_subset)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]    1.00    1.01
## [2,]    1.01    1.02
## [3,]    1.01    1.04
## [4,]    1.00    1.00
## [5,]    1.00    1.00
## [6,]    1.00    1.00
## [7,]    1.00    1.00
## [8,]    1.00    1.00
## [9,]    1.00    1.00
## [10,]   1.01    1.02
##
## Multivariate psrf
##
## 1.02

```

Inference

```

beta_samples_matrix <- rbind(chain1$samples$beta, chain2$samples$beta, chain3$samples$beta)

colnames(beta_samples_matrix) <- c("Intercept", names(fhs_model_df[, 19:(ncol(fhs_model_df) - 4)]))

```

```
(beta_inference <- round(t(apply(beta_samples_matrix, 2, quantile, c(0.5, 0.025, 0.975))),5))

##                               50%      2.5%     97.5%
## Intercept              6.66096  6.65668  6.66524
## flood_risk_pc1         -0.01108 -0.02060 -0.00158
## flood_risk_pc2          0.00533 -0.00385  0.01462
## flood_risk_pc3         -0.00927 -0.01673 -0.00186
## flood_risk_pc4         -0.00485 -0.01107  0.00141
## flood_risk_pc5          0.00186 -0.00478  0.00853
## EP_POV                  0.31408  0.30180  0.32623
## EP_UNEMP                 0.03009  0.02211  0.03811
## EP_PCI                  -0.03555 -0.04750 -0.02370
## EP_NOHSDP                0.19426  0.17873  0.20985
## EP_AGE65                 1.38103  1.37101  1.39114
## EP_AGE17                 0.27865  0.26803  0.28926
## EP_DISABL                0.27011  0.26004  0.28016
## EP_SNGPNT                -0.06513 -0.07436 -0.05592
## EP_MINRTY                -0.03937 -0.05473 -0.02402
## EP_LIMENG                -0.06162 -0.07557 -0.04777
## EP_MUNIT                 -0.05726 -0.06628 -0.04830
## EP_MOBILE                 0.07929  0.07110  0.08751
## EP_CROWD                 -0.04639 -0.05689 -0.03592
## EP_NOVEH                  0.12667  0.11327  0.14012
## EP_GROUPQ                 -0.09433 -0.10135 -0.08729
## EP_UNINSUR                0.10416  0.09374  0.11453
## co                        -0.13988 -0.17665 -0.10305
## no2                       -0.07051 -0.11876 -0.02338
## o3                         -0.14850 -0.22588 -0.07081
## pm10                      -0.15838 -0.19033 -0.12650
## pm25                      0.46131  0.41293  0.50905
## so2                        0.03603  0.00335  0.06854
## summer_tmmx                0.07514  0.01949  0.12947
## winter_tmmx                -0.26586 -0.41929 -0.10301
## summer_rmax                -0.04452 -0.11059  0.01935
## winter_rmax                 0.04645 -0.00199  0.09661
## Data_Value_CSMOKING       0.78630  0.76618  0.80626
```

List of significant beta coefficients:

```
colnames(beta_samples_matrix)[sign(beta_inference[, 2]) == sign(beta_inference[, 3])]

## [1] "Intercept"           "flood_risk_pc1"        "flood_risk_pc3"
## [4] "EP_POV"               "EP_UNEMP"             "EP_PCI"
## [7] "EP_NOHSDP"            "EP_AGE65"              "EP_AGE17"
## [10] "EP_DISABL"            "EP_SNGPNT"             "EP_MINRTY"
## [13] "EP_LIMENG"             "EP_MUNIT"              "EP_MOBILE"
## [16] "EP_CROWD"              "EP_NOVEH"              "EP_GROUPQ"
## [19] "EP_UNINSUR"            "co"                   "no2"
## [22] "o3"                   "pm10"                 "pm25"
## [25] "so2"                  "summer_tmmx"            "winter_tmmx"
## [28] "Data_Value_CSMOKING"
```

Credible Interval plots for the coefficients, in ggplot

```
# first, process the beta_inference matrix in a form ggplot can understand

beta_inference_df <- as.data.frame(beta_inference)

beta_inference_df <- mutate(beta_inference_df, var_name = row.names(beta_inference_df))

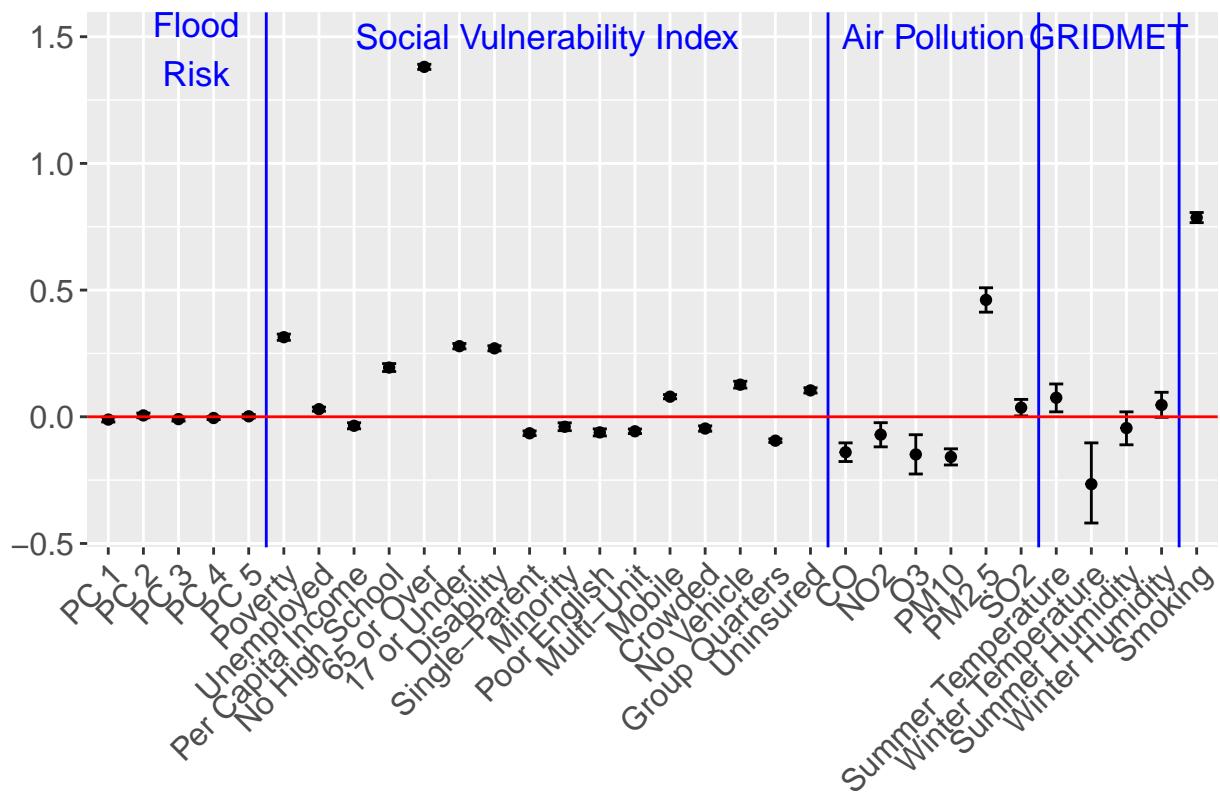
beta_inference_df <- rename(beta_inference_df,
                           post_median = `50%`,
                           post_2.5 = `2.5%`,
                           post_97.5 = `97.5%`)

beta_inference_df$var_name <- factor(beta_inference_df$var_name, levels = beta_inference_df$var_name)

p <- ggplot(beta_inference_df[-1, ], aes(x = var_name, y = post_median)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(),
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 10)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
  geom_vline(xintercept = c(5.5, 21.5, 27.5, 31.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3.5, y = 1.45, label = "Flood\\nRisk",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 13.5, y = 1.5, label = "Social Vulnerability Index",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 24.5, y = 1.5, label = "Air Pollution",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 29.5, y = 1.5, label = "GRIDMET",
           col = "blue", size = 4.5) +
  scale_x_discrete(labels = c("PC 1", "PC 2", "PC 3", "PC 4", "PC 5",
                             "Poverty", "Unemployed", "Per Capita Income", "No High School",
                             "65 or Over", "17 or Under", "Disability",
                             "Single-Parent", "Minority", "Poor English",
                             "Multi-Unit", "Mobile", "Crowded",
                             "No Vehicle", "Group Quarters", "Uninsured",
                             "CO", "NO2", "O3", "PM10", "PM2.5", "SO2",
                             "Summer Temperature", "Winter Temperature", "Summer Humidity", "Winter Humidity",
                             "Smoking")) + ggtitle("95% Credible Intervals of Coefficients, Coronary Heart Disease Risk Factors")
```

p

95% Credible Intervals of Coefficients, Coronary Heart Disease



CAR model results, High Blood Pressure

Inference is based on 3 markov chains, each of which has been run for 110000 samples, the first 10000 of which has been removed for burn-in. The remaining 100000 samples are thinned by 2, resulting in 150000 samples for inference across the 3 Markov chains.

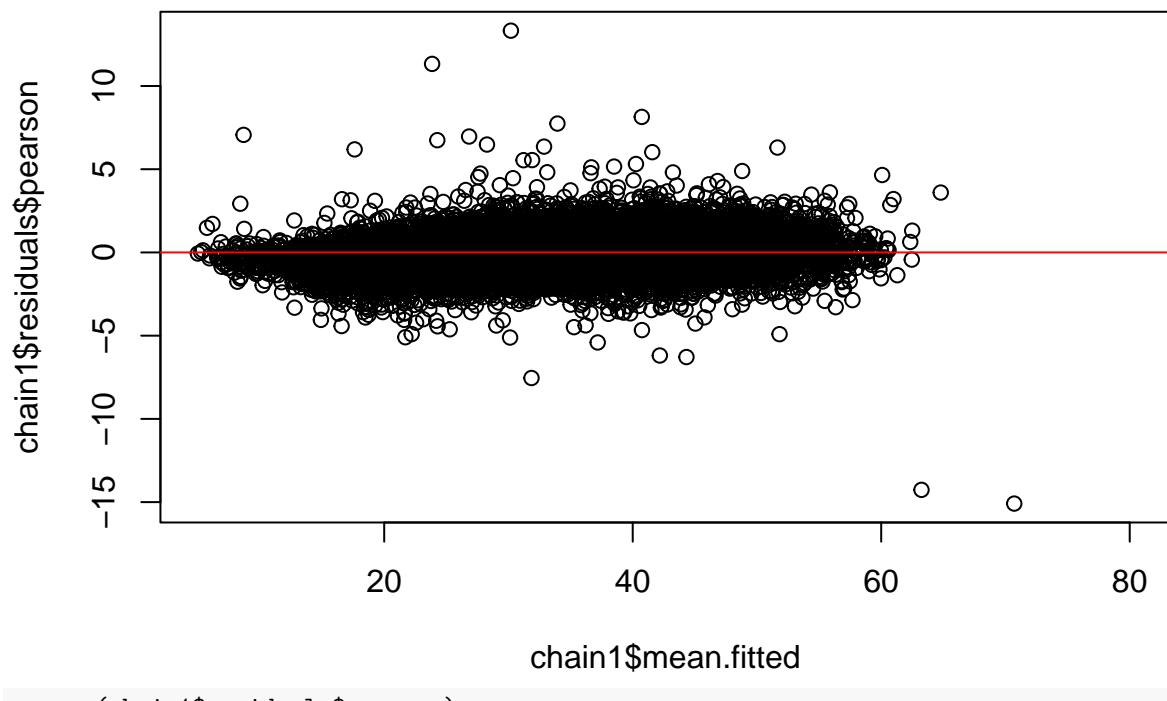
```
load(here("modeling_files/all_census_tract_BPHIGH.RData"))
```

Model Diagnostics

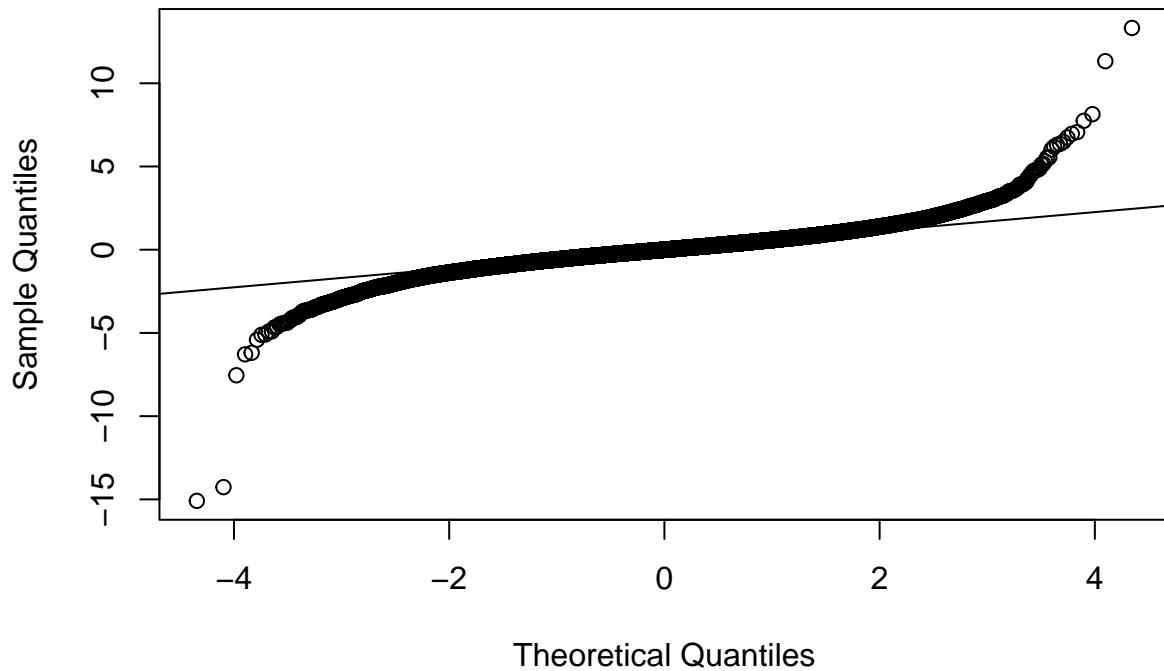
Residual plot

```
plot(chain1$mean.fitted, chain1$residuals$pearson)

abline(h = 0, col = "red")
```



Normal Q-Q Plot



Beta samples

```
beta_samples <- mcmc.list(chain1$samples$beta, chain2$samples$beta,
                           chain3$samples$beta)

effectiveSize(beta_samples)

##      var1      var2      var3      var4      var5      var6
## 81159.39421 24797.38439 22842.06842 31691.54811 50955.85839 44921.74636
##      var7      var8      var9      var10     var11     var12
## 35310.18055 55763.85442 25266.52388 34381.79558 28371.27269 39397.57023
##      var13     var14     var15     var16     var17     var18
## 42868.30093 59467.67421 18147.72148 30307.66492 30151.71545 28634.59239
##      var19     var20     var21     var22     var23     var24
## 43263.04289 25138.27409 55437.66800 34378.03414 7323.09328 3801.62286
##      var25     var26     var27     var28     var29     var30
## 266.03192 3197.40696 1758.34444 2176.67299 616.29419 94.74895
##      var31     var32     var33
## 688.93589 976.37674 17603.92651

plot(beta_samples)

gelman.diag(beta_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,] 1.00 1.00
## [2,] 1.00 1.00
## [3,] 1.00 1.00
## [4,] 1.00 1.00
## [5,] 1.00 1.00
## [6,] 1.00 1.00
## [7,] 1.00 1.00
## [8,] 1.00 1.00
## [9,] 1.00 1.00
## [10,] 1.00 1.00
## [11,] 1.00 1.00
## [12,] 1.00 1.00
## [13,] 1.00 1.00
## [14,] 1.00 1.00
## [15,] 1.00 1.00
## [16,] 1.00 1.00
## [17,] 1.00 1.00
## [18,] 1.00 1.00
## [19,] 1.00 1.00
## [20,] 1.00 1.00
## [21,] 1.00 1.00
## [22,] 1.00 1.00
## [23,] 1.00 1.01
## [24,] 1.00 1.01
## [25,] 1.03 1.10
## [26,] 1.00 1.02
## [27,] 1.02 1.06
## [28,] 1.00 1.01
```

```

## [29,]      1.03      1.10
## [30,]      1.10      1.30
## [31,]      1.00      1.00
## [32,]      1.01      1.01
## [33,]      1.00      1.00
##
## Multivariate psrf
##
## 1.1

```

Examining sigma2, nu2, rho

```

sigma2_samples <- mcmc.list(chain1$samples$sigma2, chain2$samples$sigma2,
                             chain3$samples$sigma2)

nu2_samples <- mcmc.list(chain1$samples$nu2, chain2$samples$nu2,
                         chain3$samples$nu2)

plot(sigma2_samples)

plot(nu2_samples)

gelman.diag(sigma2_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1      1
gelman.diag(nu2_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1      1

```

Examining a sample of the 3108 phi parameters

```

phi_samples <- mcmc.list(chain1$samples$phi, chain2$samples$phi, chain3$samples$phi)

set.seed(1157, kind = "Mersenne-Twister", normal.kind = "Inversion", sample.kind = "Rejection")

phi_subset_idx <- sample(1:ncol(phi_samples[[1]]), size = 10)

phi_samples_subset <- phi_samples[, phi_subset_idx]

plot(phi_samples_subset)

gelman.diag(phi_samples_subset)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1.01      1.02

```

```

## [2,]    1.01    1.02
## [3,]    1.02    1.06
## [4,]    1.00    1.01
## [5,]    1.00    1.00
## [6,]    1.00    1.00
## [7,]    1.00    1.00
## [8,]    1.00    1.00
## [9,]    1.00    1.00
## [10,]   1.01    1.03
##
## Multivariate psrf
##
## 1.02

```

Inference

```

beta_samples_matrix <- rbind(chain1$samples$beta, chain2$samples$beta, chain3$samples$beta)

colnames(beta_samples_matrix) <- c("Intercept", names(fhs_model_df[, 19:(ncol(fhs_model_df) - 4)]))

(beta_inference <- round(t(apply(beta_samples_matrix, 2, quantile, c(0.5, 0.025, 0.975))),5))

##                               50%     2.5%    97.5%
## Intercept            32.32388 32.31252 32.33529
## flood_risk_pc1      -0.07969 -0.11030 -0.04904
## flood_risk_pc2      -0.03276 -0.06277 -0.00256
## flood_risk_pc3      -0.04913 -0.07305 -0.02528
## flood_risk_pc4      -0.00216 -0.02192  0.01771
## flood_risk_pc5      0.01443 -0.00666  0.03566
## EP_POV               0.03682 -0.00246  0.07601
## EP_UNEMP              0.12881  0.10374  0.15404
## EP_PCI                0.25441  0.21584  0.29286
## EP_NOHSDP             -0.00369 -0.05329  0.04586
## EP_AGE65              4.06691  4.03477  4.09943
## EP_AGE17              0.60888  0.57506  0.64255
## EP_DISABL             0.74122  0.70934  0.77295
## EP_SNGPNT             -0.04916 -0.07820 -0.02020
## EP_MINRTY              2.64271  2.59171  2.69325
## EP_LIMENG             -0.88639 -0.93131 -0.84186
## EP_MUNIT              -0.60855 -0.63740 -0.58006
## EP_MOBILE              0.11879  0.09277  0.14494
## EP_CROWD              -0.12461 -0.15783 -0.09133
## EP_NOVEH               0.61229  0.56931  0.65531
## EP_GROUPQ              -0.58982 -0.61207 -0.56746
## EP_UNINSUR             0.23368  0.20066  0.26656
## co                     -0.69316 -0.81604 -0.56996
## no2                    -0.67967 -0.84367 -0.51812
## o3                     -0.48488 -0.76167 -0.19597
## pm10                  -0.49465 -0.60402 -0.38501
## pm25                   1.18594  1.01921  1.34973
## so2                     0.03861 -0.07447  0.15169
## summer_tmmx            0.09132 -0.11130  0.28875
## winter_tmmx           -0.69387 -1.25780 -0.09207

```

```

## summer_rmax      -0.27015 -0.50522 -0.04691
## winter_rmax       0.17283  0.00036  0.35383
## Data_Value_CSMOKING  2.52645  2.46123  2.59127

```

List of significant beta coefficients:

```

colnames(beta_samples_matrix)[sign(beta_inference[, 2]) == sign(beta_inference[, 3])]

## [1] "Intercept"           "flood_risk_pc1"      "flood_risk_pc2"
## [4] "flood_risk_pc3"      "EP_UNEMP"            "EP_PCI"
## [7] "EP_AGE65"             "EP_AGE17"             "EP_DISABL"
## [10] "EP_SNGPNT"            "EP_MINRTY"            "EP_LIMENG"
## [13] "EP_MUNIT"              "EP_MOBILE"             "EP_CROWD"
## [16] "EP_NOVEH"              "EP_GROUPQ"             "EP_UNINSUR"
## [19] "co"                   "no2"                  "o3"
## [22] "pm10"                 "pm25"                 "winter_tmmx"
## [25] "summer_rmax"           "winter_rmax"           "Data_Value_CSMOKING"

```

Credible Interval plots for the coefficients, in ggplot

```

# first, process the beta_inference matrix in a form ggplot can understand

beta_inference_df <- as.data.frame(beta_inference)

beta_inference_df <- mutate(beta_inference_df, var_name = row.names(beta_inference_df))

beta_inference_df <- rename(beta_inference_df,
                             post_median = `50%`,
                             post_2.5 = `2.5%`,
                             post_97.5 = `97.5%`)

beta_inference_df$var_name <- factor(beta_inference_df$var_name, levels = beta_inference_df$var_name)

p <- ggplot(beta_inference_df[-1, ], aes(x = var_name, y = post_median)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(),
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 10)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
  geom_vline(xintercept = c(5.5, 21.5, 27.5, 31.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3.5, y = 1.45, label = "Flood\nRisk",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 13.5, y = 1.5, label = "Social Vulnerability Index",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 24.5, y = 1.5, label = "Air Pollution",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 29.5, y = 1.5, label = "GRIDMET",
           col = "blue", size = 4.5) +
  scale_x_discrete(labels = c("PC 1", "PC 2", "PC 3", "PC 4", "PC 5",
                             "Poverty", "Unemployed", "Per Capita Income", "No High School",
                             "65 or Over", "17 or Under", "Disability",
                             "Single-Parent", "Minority", "Poor English",
                             "Multi-Unit", "Mobile", "Crowded"),
                    guide = "none")

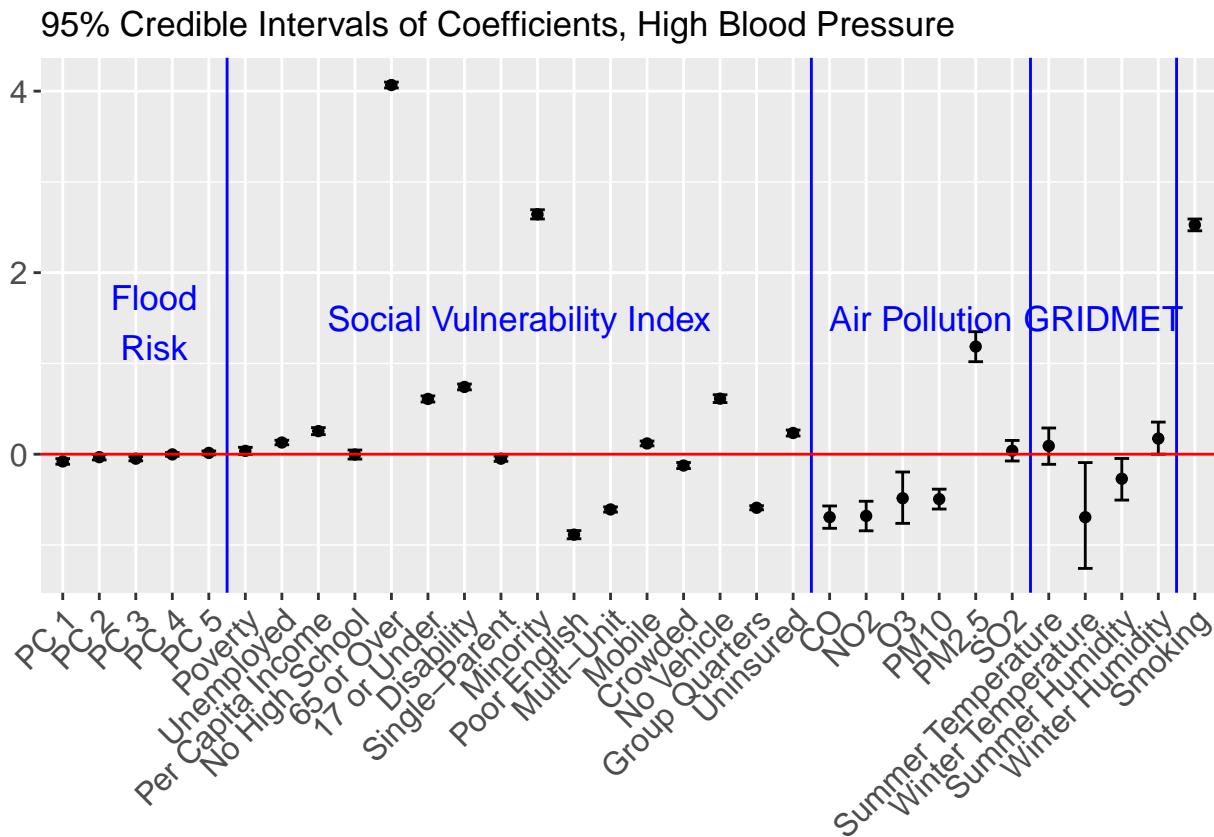
```

```

"No Vehicle", "Group Quarters", "Uninsured",
"CO", "NO2", "O3", "PM10", "PM2.5", "SO2",
"Summer Temperature", "Winter Temperature", "Summer Humidity", "Winter Humidity",
"Smoking")) + ggtitle("95% Credible Intervals of Coefficients, High Blood Pressure")

```

p



CAR model results, Asthma

Inference is based on 3 markov chains, each of which has been run for 110000 samples, the first 10000 of which has been removed for burn-in. The remaining 100000 samples are thinned by 2, resulting in 150000 samples for inference across the 3 Markov chains.

```
load(here("modeling_files/all_census_tract_CASTHMA.RData"))
```

Model Diagnostics

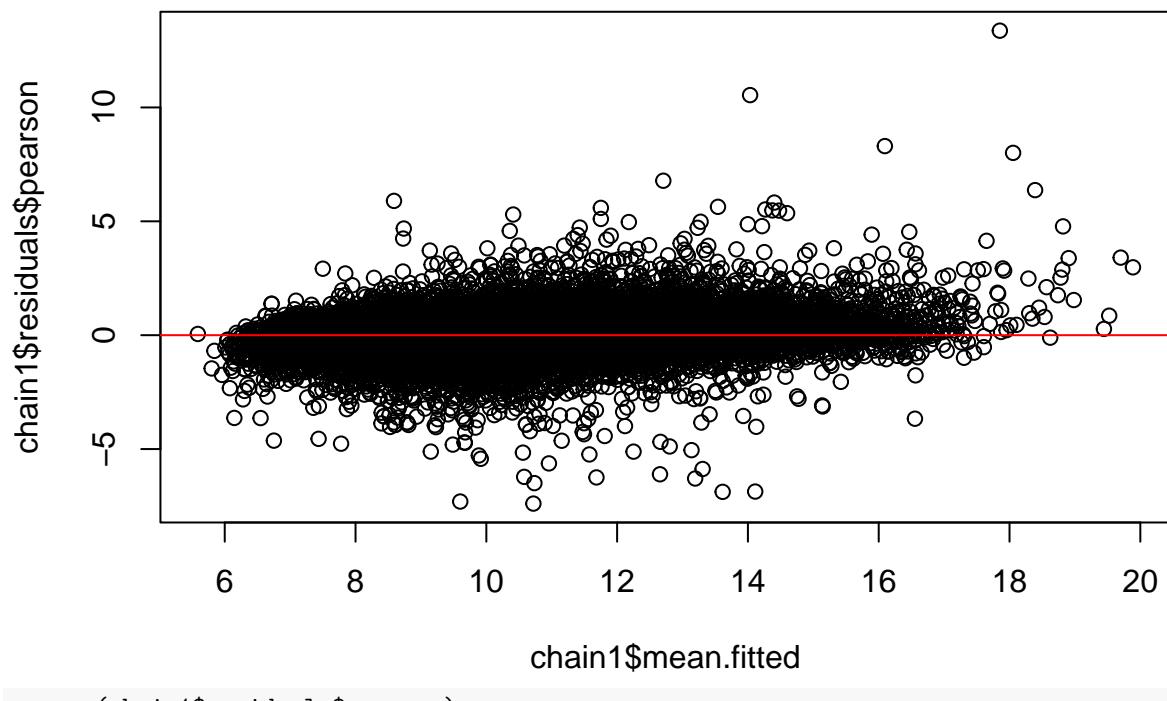
Residual plot

```

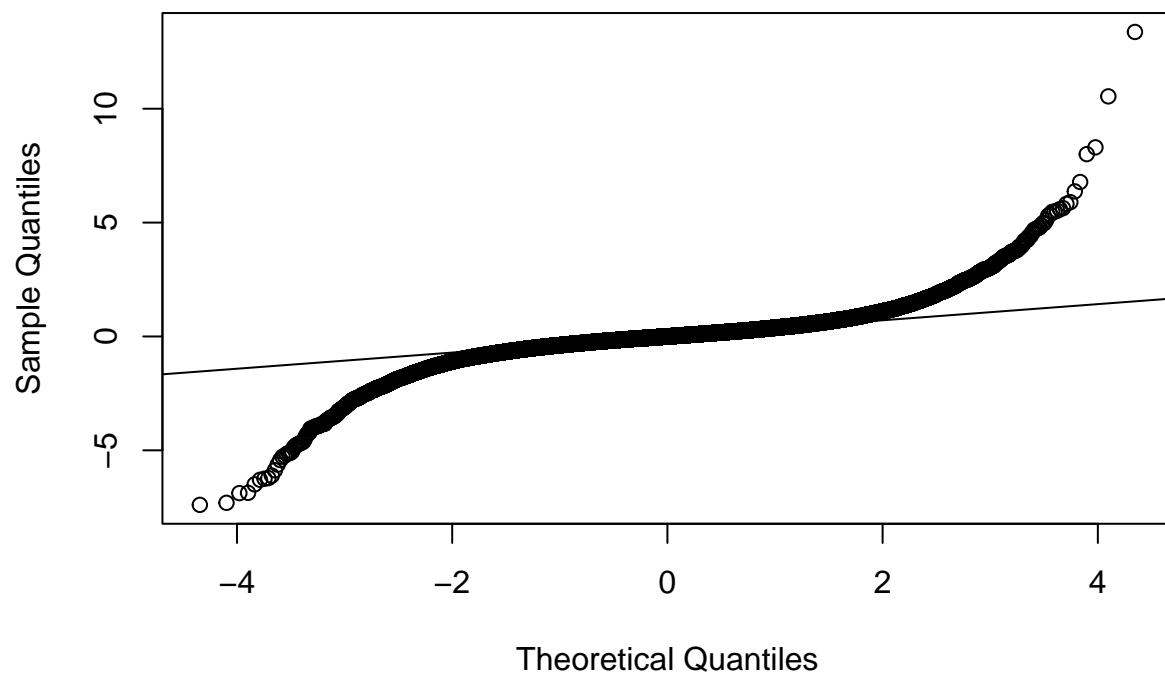
plot(chain1$mean.fitted, chain1$residuals$pearson)

abline(h = 0, col = "red")

```



Normal Q-Q Plot



Theoretical Quantiles

Beta samples

```
beta_samples <- mcmc.list(chain1$samples$beta, chain2$samples$beta,
                           chain3$samples$beta)

effectiveSize(beta_samples)

##      var1      var2      var3      var4      var5      var6
## 51799.97269 18277.87075 15540.70380 21555.95317 33484.15962 29836.00531
##      var7      var8      var9      var10     var11     var12
## 23908.19500 36228.71569 17809.49430 23496.36014 22786.68571 25989.89614
##      var13     var14     var15     var16     var17     var18
## 30241.63649 38687.14389 12911.44927 20176.37567 21229.14162 20616.19718
##      var19     var20     var21     var22     var23     var24
## 28315.31888 18102.91363 35584.90286 23687.47521 5184.70595 2799.31708
##      var25     var26     var27     var28     var29     var30
## 209.19229  2269.65724 1276.60019 1471.32388 450.24784   66.25815
##      var31     var32     var33
## 505.50811  734.99869 12636.84764

plot(beta_samples)

gelman.diag(beta_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]    1.00    1.00
## [2,]    1.00    1.00
## [3,]    1.00    1.00
## [4,]    1.00    1.00
## [5,]    1.00    1.00
## [6,]    1.00    1.00
## [7,]    1.00    1.00
## [8,]    1.00    1.00
## [9,]    1.00    1.00
## [10,]   1.00    1.00
## [11,]   1.00    1.00
## [12,]   1.00    1.00
## [13,]   1.00    1.00
## [14,]   1.00    1.00
## [15,]   1.00    1.00
## [16,]   1.00    1.00
## [17,]   1.00    1.00
## [18,]   1.00    1.00
## [19,]   1.00    1.00
## [20,]   1.00    1.00
## [21,]   1.00    1.00
## [22,]   1.00    1.00
## [23,]   1.00    1.01
## [24,]   1.00    1.01
## [25,]   1.05    1.16
## [26,]   1.01    1.02
## [27,]   1.02    1.08
## [28,]   1.01    1.02
```

```

## [29,]      1.04      1.13
## [30,]      1.13      1.38
## [31,]      1.00      1.00
## [32,]      1.01      1.02
## [33,]      1.00      1.00
##
## Multivariate psrf
##
## 1.12

```

Examining sigma2, nu2, rho

```

sigma2_samples <- mcmc.list(chain1$samples$sigma2, chain2$samples$sigma2,
                             chain3$samples$sigma2)

nu2_samples <- mcmc.list(chain1$samples$nu2, chain2$samples$nu2,
                         chain3$samples$nu2)

plot(sigma2_samples)

plot(nu2_samples)

gelman.diag(sigma2_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1      1
gelman.diag(nu2_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1      1

```

Examining a sample of the 3108 phi parameters

```

phi_samples <- mcmc.list(chain1$samples$phi, chain2$samples$phi, chain3$samples$phi)

set.seed(1157, kind = "Mersenne-Twister", normal.kind = "Inversion", sample.kind = "Rejection")

phi_subset_idx <- sample(1:ncol(phi_samples[[1]]), size = 10)

phi_samples_subset <- phi_samples[, phi_subset_idx]

plot(phi_samples_subset)

gelman.diag(phi_samples_subset)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1.01      1.03

```

```

## [2,]    1.01    1.03
## [3,]    1.04    1.12
## [4,]    1.00    1.01
## [5,]    1.00    1.00
## [6,]    1.00    1.00
## [7,]    1.00    1.00
## [8,]    1.00    1.00
## [9,]    1.00    1.00
## [10,]   1.01    1.04
##
## Multivariate psrf
##
## 1.04

```

Inference

```

beta_samples_matrix <- rbind(chain1$samples$beta, chain2$samples$beta, chain3$samples$beta)

colnames(beta_samples_matrix) <- c("Intercept", names(fhs_model_df[, 19:(ncol(fhs_model_df) - 4)]))

(beta_inference <- round(t(apply(beta_samples_matrix, 2, quantile, c(0.5, 0.025, 0.975))),5))

##                               50%     2.5%    97.5%
## Intercept                9.89827  9.89672  9.89982
## flood_risk_pc1           -0.00310 -0.00834  0.00211
## flood_risk_pc2           -0.00344 -0.00863  0.00175
## flood_risk_pc3            0.00743  0.00335  0.01149
## flood_risk_pc4            0.00364  0.00030  0.00701
## flood_risk_pc5           -0.00002 -0.00361  0.00358
## EP_POV                    0.28271  0.27605  0.28937
## EP_UNEMP                  0.07064  0.06643  0.07489
## EP_PCI                    -0.06379 -0.07033 -0.05722
## EP_NOHSDP                 0.07433  0.06592  0.08268
## EP_AGE65                  0.09781  0.09244  0.10329
## EP_AGE17                  0.00971  0.00399  0.01539
## EP_DISABL                 -0.06089 -0.06627 -0.05555
## EP_SNGPNT                 0.03879  0.03392  0.04367
## EP_MINRTY                 0.32423  0.31547  0.33285
## EP_LIMENG                 -0.25511 -0.26274 -0.24747
## EP_MUNIT                  -0.02226 -0.02714 -0.01742
## EP_MOBILE                 -0.01532 -0.01972 -0.01087
## EP_CROWD                  -0.02058 -0.02620 -0.01494
## EP_NOVEH                  0.15009  0.14279  0.15735
## EP_GROUPQ                 -0.04561 -0.04936 -0.04181
## EP_UNINSUR                -0.02820 -0.03378 -0.02264
## co                        -0.06490 -0.08637 -0.04337
## no2                       -0.16867 -0.19762 -0.14004
## o3                         0.00868 -0.04108  0.06300
## pm10                      -0.16166 -0.18092 -0.14225
## pm25                      0.29897  0.26926  0.32794
## so2                        -0.02433 -0.04457 -0.00391
## summer_tmmx                0.03313 -0.00456  0.06984
## winter_tmmx               -0.08871 -0.19225  0.02380

```

```

## summer_rmax      0.00599 -0.03640  0.04603
## winter_rmax     -0.05773 -0.08897 -0.02488
## Data_Value_CSMOKING  0.85564  0.84452  0.86663

```

List of significant beta coefficients:

```

colnames(beta_samples_matrix)[sign(beta_inference[, 2]) == sign(beta_inference[, 3])]

## [1] "Intercept"           "flood_risk_pc3"       "flood_risk_pc4"
## [4] "EP_POV"              "EP_UNEMP"            "EP_PCI"
## [7] "EP_NOHSDP"           "EP_AGE65"             "EP_AGE17"
## [10] "EP_DISABL"           "EP_SNGPNT"            "EP_MINRTY"
## [13] "EP_LIMENG"           "EP_MUNIT"             "EP_MOBILE"
## [16] "EP_CROWD"             "EP_NOVEH"             "EP_GROUPQ"
## [19] "EP_UNINSUR"          "co"                  "no2"
## [22] "pm10"                "pm25"                "so2"
## [25] "winter_rmax"         "Data_Value_CSMOKING"

```

Credible Interval plots for the coefficients, in ggplot

```

# first, process the beta_inference matrix in a form ggplot can understand

beta_inference_df <- as.data.frame(beta_inference)

beta_inference_df <- mutate(beta_inference_df, var_name = row.names(beta_inference_df))

beta_inference_df <- rename(beta_inference_df,
                             post_median = `50%`,
                             post_2.5 = `2.5%`,
                             post_97.5 = `97.5`)

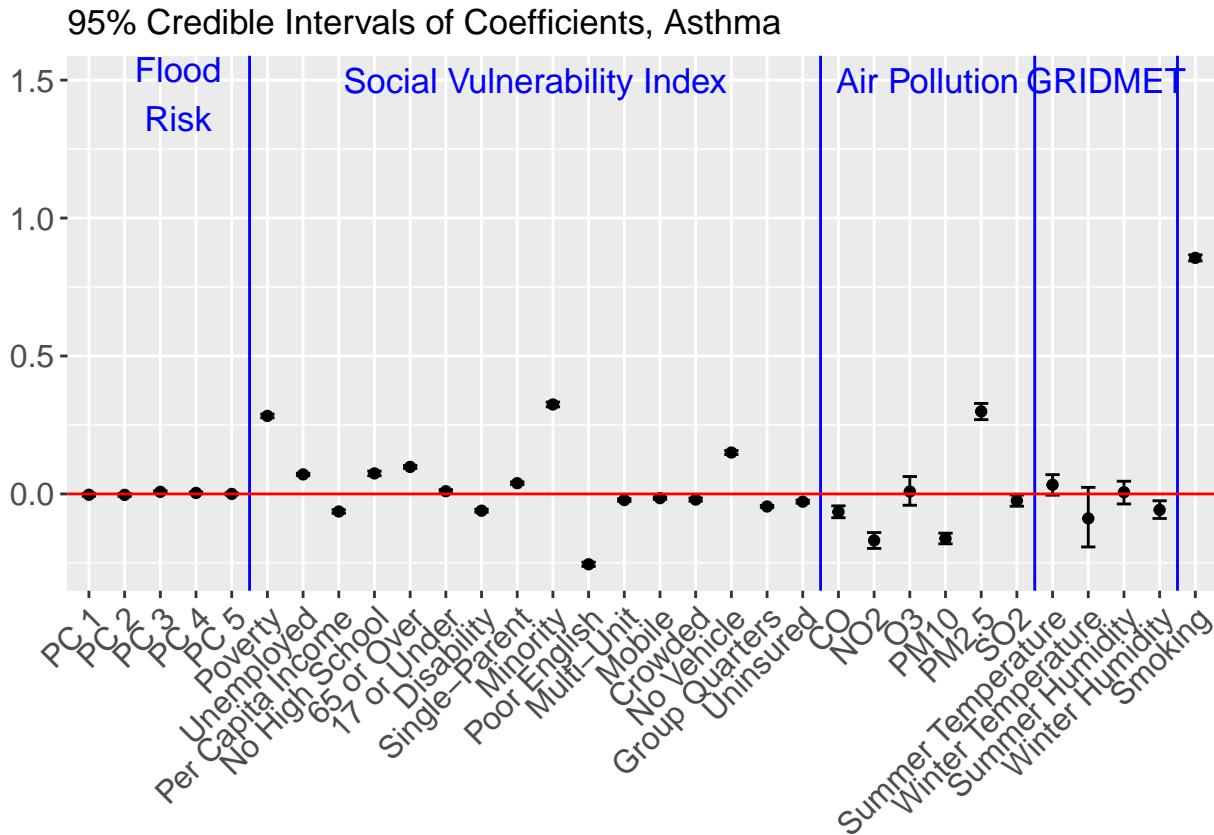
beta_inference_df$var_name <- factor(beta_inference_df$var_name, levels = beta_inference_df$var_name)

p <- ggplot(beta_inference_df[-1, ], aes(x = var_name, y = post_median)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(),
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 10)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
  geom_vline(xintercept = c(5.5, 21.5, 27.5, 31.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3.5, y = 1.45, label = "Flood\nRisk",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 13.5, y = 1.5, label = "Social Vulnerability Index",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 24.5, y = 1.5, label = "Air Pollution",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 29.5, y = 1.5, label = "GRIDMET",
           col = "blue", size = 4.5) +
  scale_x_discrete(labels = c("PC 1", "PC 2", "PC 3", "PC 4", "PC 5",
                             "Poverty", "Unemployed", "Per Capita Income", "No High School",
                             "65 or Over", "17 or Under", "Disability",
                             "Single-Parent", "Minority", "Poor English",
                             "Multi-Unit", "Mobile", "Crowded"),
                    guide = "none")

```

```
"No Vehicle", "Group Quarters", "Uninsured",
"CO", "NO2", "O3", "PM10", "PM2.5", "SO2",
"Summer Temperature", "Winter Temperature", "Summer Humidity", "Winter Humidity",
"Smoking")) + ggtitle("95% Credible Intervals of Coefficients, Asthma")
```

p



CAR model results, Poor Mental Health

Inference is based on 3 markov chains, each of which has been run for 110000 samples, the first 10000 of which has been removed for burn-in. The remaining 100000 samples are thinned by 2, resulting in 150000 samples for inference across the 3 Markov chains.

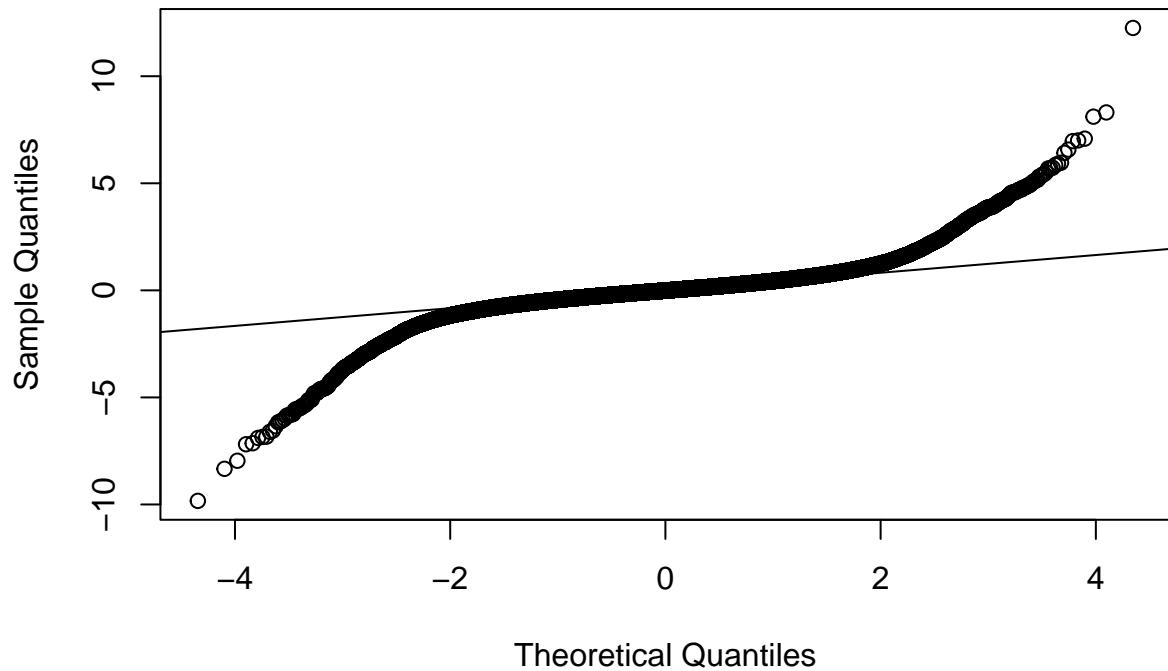
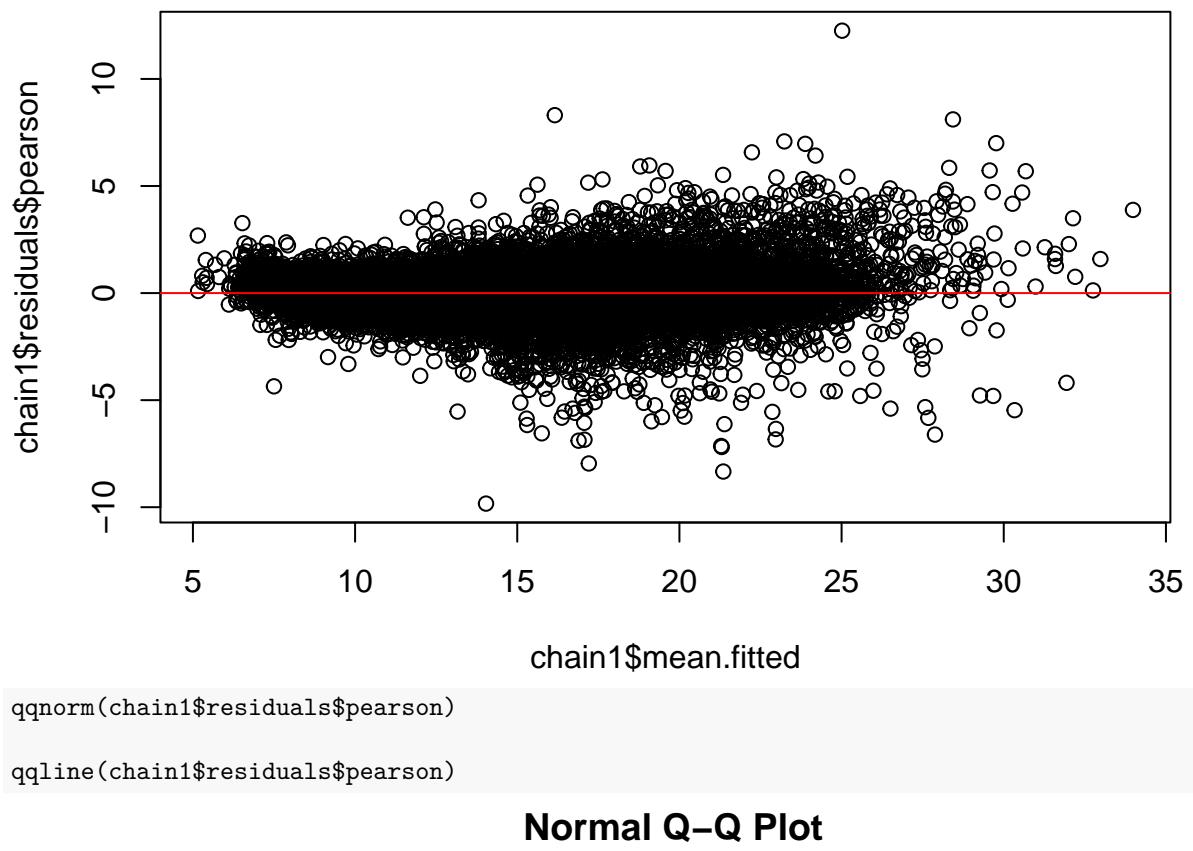
```
load(here("modeling_files/all_census_tract_MHLTH.RData"))
```

Model Diagnostics

Residual plot

```
plot(chain1$mean.fitted, chain1$residuals$pearson)

abline(h = 0, col = "red")
```



Beta samples

```
beta_samples <- mcmc.list(chain1$samples$beta, chain2$samples$beta,
                           chain3$samples$beta)

effectiveSize(beta_samples)

##      var1      var2      var3      var4      var5      var6      var7
## 67809.7977 22505.1526 19871.8448 27670.6742 43604.7155 38737.7764 25814.7409
##      var8      var9      var10     var11     var12     var13     var14
## 47906.8466 22193.2195 29779.3115 28629.0535 33691.8967 38836.3425 50753.3877
##      var15     var16     var17     var18     var19     var20     var21
## 16381.9570 26165.7625 27313.1036 25524.9983 37305.4156 22191.7899 47263.3209
##      var22     var23     var24     var25     var26     var27     var28
## 30183.4846 6490.1032 3406.3266  257.1841 2803.4303 1574.7801 1908.4972
##      var29     var30     var31     var32     var33
## 547.9531   82.8496  610.3417  892.8869 15164.8394

plot(beta_samples)

gelman.diag(beta_samples)

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]    1.00    1.00
## [2,]    1.00    1.00
## [3,]    1.00    1.00
## [4,]    1.00    1.00
## [5,]    1.00    1.00
## [6,]    1.00    1.00
## [7,]    1.00    1.00
## [8,]    1.00    1.00
## [9,]    1.00    1.00
## [10,]   1.00    1.00
## [11,]   1.00    1.00
## [12,]   1.00    1.00
## [13,]   1.00    1.00
## [14,]   1.00    1.00
## [15,]   1.00    1.00
## [16,]   1.00    1.00
## [17,]   1.00    1.00
## [18,]   1.00    1.00
## [19,]   1.00    1.00
## [20,]   1.00    1.00
## [21,]   1.00    1.00
## [22,]   1.00    1.00
## [23,]   1.00    1.01
## [24,]   1.00    1.01
## [25,]   1.04    1.12
## [26,]   1.01    1.02
## [27,]   1.02    1.06
## [28,]   1.00    1.02
## [29,]   1.03    1.11
## [30,]   1.11    1.33
```

```

## [31,]      1.00      1.00
## [32,]      1.01      1.01
## [33,]      1.00      1.00
##
## Multivariate psrf
##
## 1.11

```

Examining sigma2, nu2, rho

```

sigma2_samples <- mcmc.list(chain1$samples$sigma2, chain2$samples$sigma2,
                             chain3$samples$sigma2)

nu2_samples <- mcmc.list(chain1$samples$nu2, chain2$samples$nu2,
                         chain3$samples$nu2)

plot(sigma2_samples)

plot(nu2_samples)

gelman.diag(sigma2_samples)

## Potential scale reduction factors:
## 
##      Point est. Upper C.I.
## [1,]      1      1
gelman.diag(nu2_samples)

## Potential scale reduction factors:
## 
##      Point est. Upper C.I.
## [1,]      1      1

```

Examining a sample of the 3108 phi parameters

```

phi_samples <- mcmc.list(chain1$samples$phi, chain2$samples$phi, chain3$samples$phi)

set.seed(1157, kind = "Mersenne-Twister", normal.kind = "Inversion", sample.kind = "Rejection")

phi_subset_idx <- sample(1:ncol(phi_samples[[1]]), size = 10)

phi_samples_subset <- phi_samples[, phi_subset_idx]

plot(phi_samples_subset)

gelman.diag(phi_samples_subset)

## Potential scale reduction factors:
## 
##      Point est. Upper C.I.
## [1,]      1.01      1.02
## [2,]      1.01      1.03
## [3,]      1.02      1.08

```

```

## [4,]    1.00    1.01
## [5,]    1.00    1.00
## [6,]    1.00    1.00
## [7,]    1.00    1.00
## [8,]    1.00    1.00
## [9,]    1.00    1.00
## [10,]   1.01    1.03
##
## Multivariate psrf
##
## 1.03

```

Inference

```

beta_samples_matrix <- rbind(chain1$samples$beta, chain2$samples$beta, chain3$samples$beta)

colnames(beta_samples_matrix) <- c("Intercept", names(fhs_model_df[, 19:(ncol(fhs_model_df) - 4)]))

(beta_inference <- round(t(apply(beta_samples_matrix, 2, quantile, c(0.5, 0.025, 0.975))),5))

##                                     50%     2.5%    97.5%
## Intercept                  14.26584 14.26293 14.26876
## flood_risk_pc1            -0.00081 -0.00937  0.00771
## flood_risk_pc2            -0.00171 -0.01014  0.00674
## flood_risk_pc3             0.00960  0.00292  0.01626
## flood_risk_pc4             0.00266 -0.00283  0.00820
## flood_risk_pc5            -0.00129 -0.00719  0.00462
## EP_POV                      0.81842  0.80729  0.82953
## EP_UNEMP                     0.07109  0.06415  0.07810
## EP_PCI                       -0.32973 -0.34049 -0.31896
## EP_NOHSDP                     0.25405  0.24022  0.26786
## EP_AGE65                      -0.33624 -0.34510 -0.32728
## EP_AGE17                      -0.10573 -0.11515 -0.09637
## EP_DISABL                     -0.17481 -0.18367 -0.16598
## EP_SNGPNT                      0.07617  0.06810  0.08421
## EP_MINRTY                     -0.17907 -0.19332 -0.16496
## EP_LIMENG                      -0.03143 -0.04397 -0.01897
## EP_MUNIT                      0.08008  0.07205  0.08802
## EP_MOBILE                     -0.01421 -0.02144 -0.00693
## EP_CROWD                      0.05836  0.04910  0.06765
## EP_NOVEH                      0.07668  0.06467  0.08866
## EP_GROUPQ                      0.16894  0.16275  0.17516
## EP_UNINSUR                    -0.02150 -0.03069 -0.01236
## co                            0.08029  0.04551  0.11511
## no2                           -0.05016 -0.09677 -0.00412
## o3                            0.02717 -0.05227  0.11183
## pm10                          -0.14101 -0.17210 -0.10980
## pm25                          0.30982  0.26228  0.35635
## so2                            0.00478 -0.02757  0.03719
## summer_tmmx                   0.08150  0.02258  0.13884
## winter_tmmx                   0.01861 -0.14472  0.19531
## summer_rmax                   0.04905 -0.01838  0.11281
## winter_rmax                  -0.03111 -0.08074  0.02100

```

```
## Data_Value_CSMOKING 2.14270 2.12431 2.16088
```

List of significant beta coefficients:

```
colnames(beta_samples_matrix)[sign(beta_inference[, 2]) == sign(beta_inference[, 3])]
```

```
## [1] "Intercept"           "flood_risk_pc3"      "EP_POV"
## [4] "EP_UNEMP"            "EP_PCI"              "EP_NOHSDP"
## [7] "EP_AGE65"             "EP_AGE17"            "EP_DISABL"
## [10] "EP_SNGPNT"            "EP_MINRTY"           "EP_LIMENG"
## [13] "EP_MUNIT"             "EP_MOBILE"            "EP_CROWD"
## [16] "EP_NOVEH"             "EP_GROUPQ"           "EP_UNINSUR"
## [19] "co"                   "no2"                 "pm10"
## [22] "pm25"                "summer_tmmx"          "Data_Value_CSMOKING"
```

Credible Interval plots for the coefficients, in ggplot

```
# first, process the beta_inference matrix in a form ggplot can understand
beta_inference_df <- as.data.frame(beta_inference)

beta_inference_df <- mutate(beta_inference_df, var_name = row.names(beta_inference_df))

beta_inference_df <- rename(beta_inference_df,
                             post_median = `50%`,
                             post_2.5 = `2.5%`,
                             post_97.5 = `97.5%`)

beta_inference_df$var_name <- factor(beta_inference_df$var_name, levels = beta_inference_df$var_name)

p <- ggplot(beta_inference_df[-1, ], aes(x = var_name, y = post_median)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(),
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 10)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
  geom_vline(xintercept = c(5.5, 21.5, 27.5, 31.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3.5, y = 1.45, label = "Flood\nRisk",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 13.5, y = 1.5, label = "Social Vulnerability Index",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 24.5, y = 1.5, label = "Air Pollution",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 29.5, y = 1.5, label = "GRIDMET",
           col = "blue", size = 4.5) +
  scale_x_discrete(labels = c("PC 1", "PC 2", "PC 3", "PC 4", "PC 5",
                             "Poverty", "Unemployed", "Per Capita Income", "No High School",
                             "65 or Over", "17 or Under", "Disability",
                             "Single-Parent", "Minority", "Poor English",
                             "Multi-Unit", "Mobile", "Crowded",
                             "No Vehicle", "Group Quarters", "Uninsured",
                             "CO", "NO2", "O3", "PM10", "PM2.5", "SO2",
                             "Summer Temperature", "Winter Temperature", "Summer Humidity", "Winter Hu"))
```

