

# Basic CAR Model

Alvin Sheng

```
library(here)

## here() starts at /Users/Alvin/Documents/NCSU_Spring_2022/NIH_SIP/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     v forcats  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 /Users/Alvin/Documents/NCSU_Spring_2022/NIH_SIP/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.

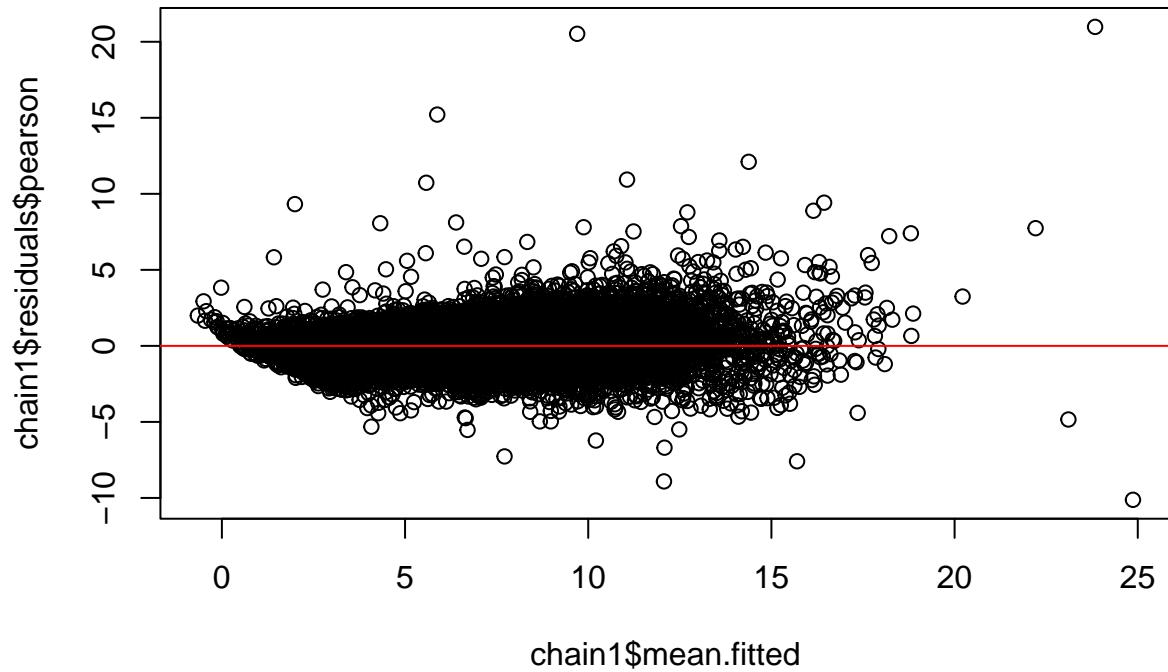
```
load(here("modeling_files/all_census_tract_intrinsic.RData"))
```

## Model Diagnostics

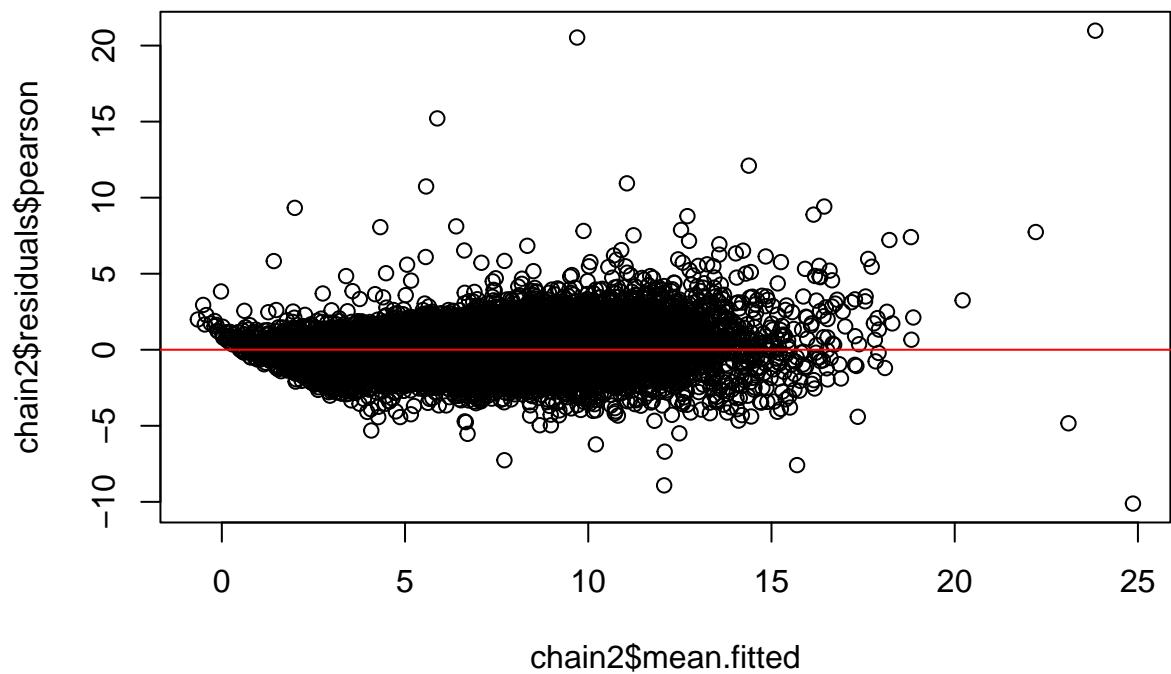
### Residual Plots

For each chain

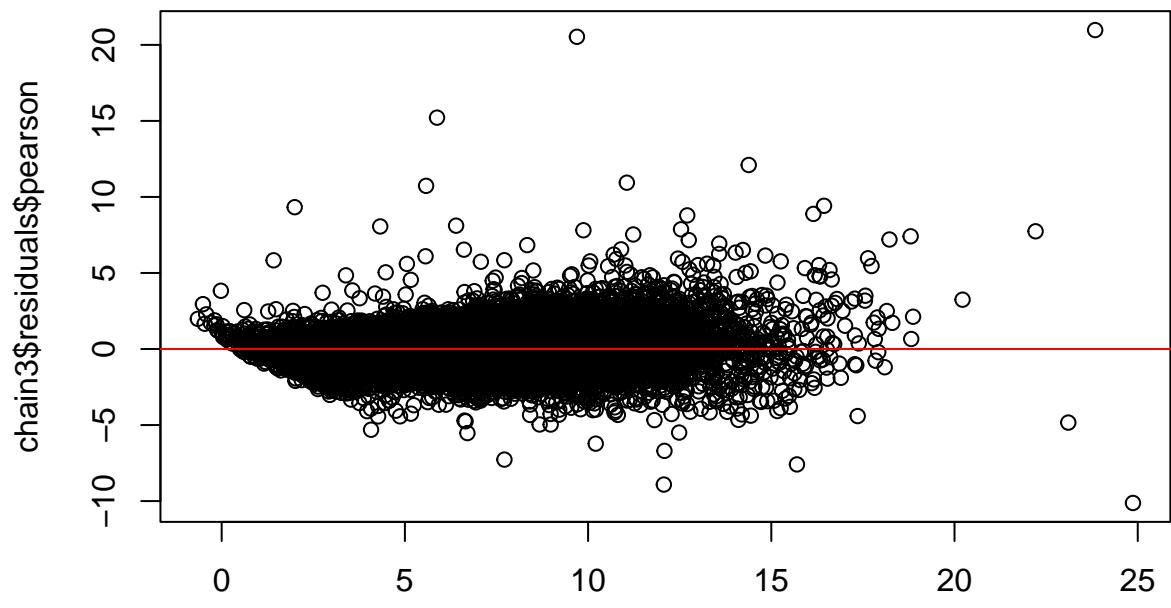
```
plot(chain1$mean.fitted, chain1$residuals$pearson)  
abline(h = 0, col = "red")
```



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

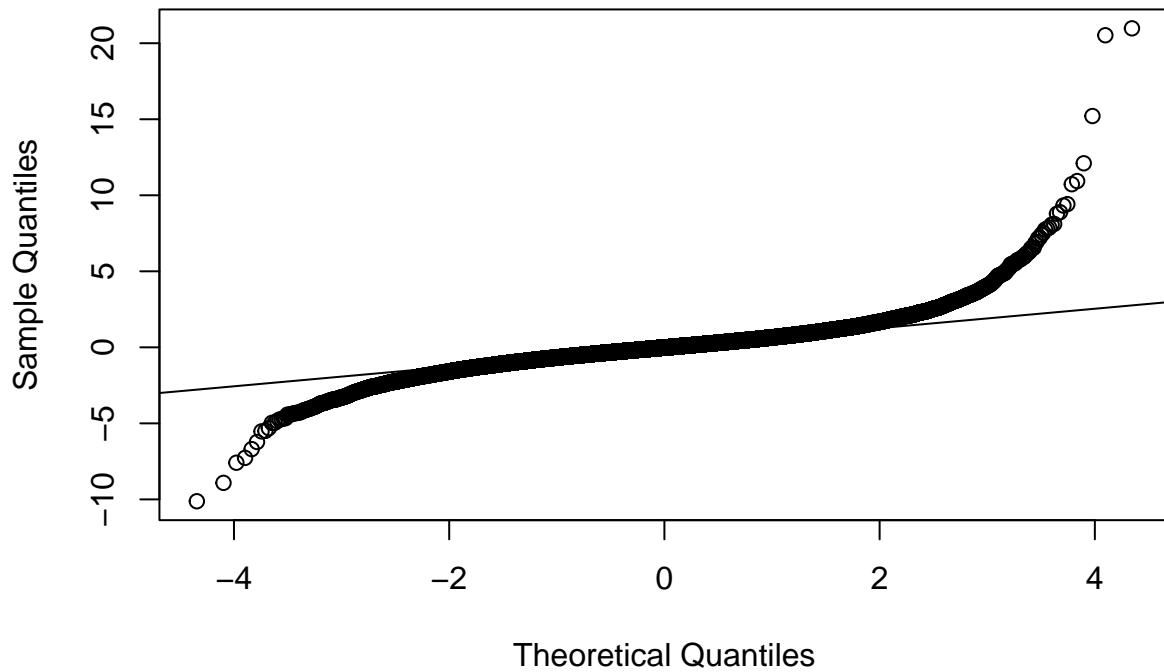


```
plot(chain3$mean.fitted, chain3$residuals$pearson)  
abline(h = 0, col = "red")
```



```
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
## 125587.3887  30497.4799  25914.7758  42380.0138  53869.0285  59894.4919
##          var7          var8          var9          var10         var11         var12
##  44461.2121  73323.0208  30471.0906  41389.2586  35310.2829  49882.2932
##          var13         var14         var15         var16         var17         var18
##  48412.2894  73536.5898  23023.7651  37184.5501  38369.7779  33095.1749
##          var19         var20         var21         var22         var23         var24
##  53797.5897  28734.7493  74018.4362  40469.9423  8444.2043   4793.4227
##          var25         var26         var27         var28         var29         var30
##    324.6699   3734.3679   2020.2759   2651.9490   744.7580   120.8511
##          var31         var32         var33
##    801.7166  1226.6691  20766.9821

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    1.01
## [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    1.02
##
## 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[, 14:(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              6.66097  6.65670  6.66526
## flood_risk_pc1         -0.00962 -0.01833 -0.00099
## flood_risk_pc2          0.01185  0.00214  0.02173
## flood_risk_pc3          0.00366 -0.00375  0.01104
## flood_risk_pc4          0.00625 -0.00048  0.01295
## flood_risk_pc5          0.00128 -0.00519  0.00772
## EP_POV                  0.31406  0.30179  0.32622
## EP_UNEMP                 0.03016  0.02218  0.03816
## EP_PCI                  -0.03565 -0.04759 -0.02379
## EP_NOHSDP                0.19414  0.17861  0.20972
## EP_AGE65                 1.38092  1.37091  1.39103
## EP_AGE17                 0.27855  0.26790  0.28917
## EP_DISABL                0.27025  0.26019  0.28033
## EP_SNGPNT                -0.06502 -0.07422 -0.05581
## EP_MINRTY                -0.03944 -0.05479 -0.02409
## EP_LIMENG                -0.06166 -0.07559 -0.04779
## EP_MUNIT                 -0.05733 -0.06636 -0.04838
## EP_MOBILE                 0.07933  0.07115  0.08755
## EP_CROWD                 -0.04648 -0.05696 -0.03598
## EP_NOVEH                  0.12666  0.11325  0.14010
## EP_GROUPQ                -0.09427 -0.10129 -0.08723
## EP_UNINSUR                0.10421  0.09381  0.11456
## co                        -0.14219 -0.17899 -0.10531
## no2                       -0.06714 -0.11525 -0.01999
## o3                         -0.15162 -0.22884 -0.07385
## pm10                      -0.15766 -0.18959 -0.12574
## pm25                      0.46467  0.41629  0.51246
## so2                        0.03557  0.00284  0.06808
## summer_tmmx                0.07436  0.01870  0.12864
## winter_tmmx                -0.26287 -0.41657 -0.10012
## summer_rmax                -0.04555 -0.11163  0.01835
## winter_rmax                 0.04703 -0.00143  0.09724
## Data_Value_CSMOKING      0.78661  0.76650  0.80657
```

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] "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))

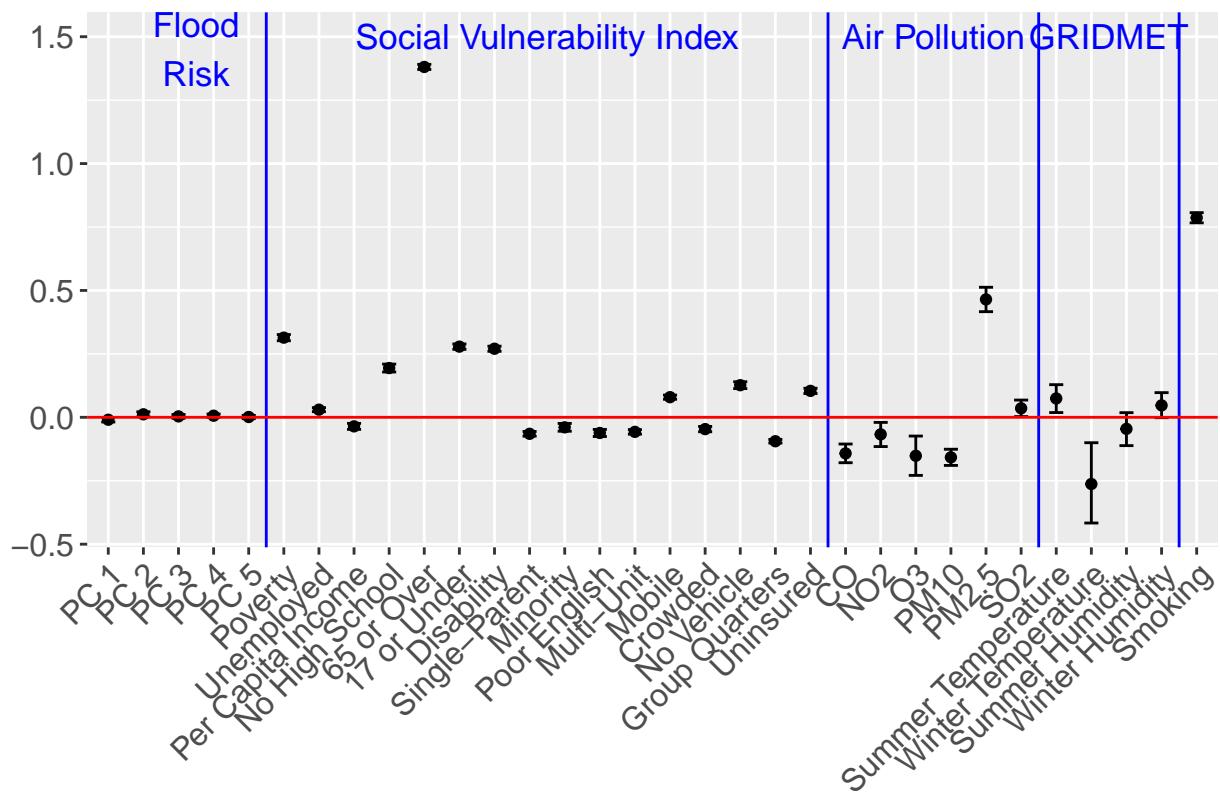
beta_inference_df <- rename(beta_inference_df,
                           post_median = `50%`,
                           post_2.5 = `2.5%`,
                           post_97.5 = `97.5%`)

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.

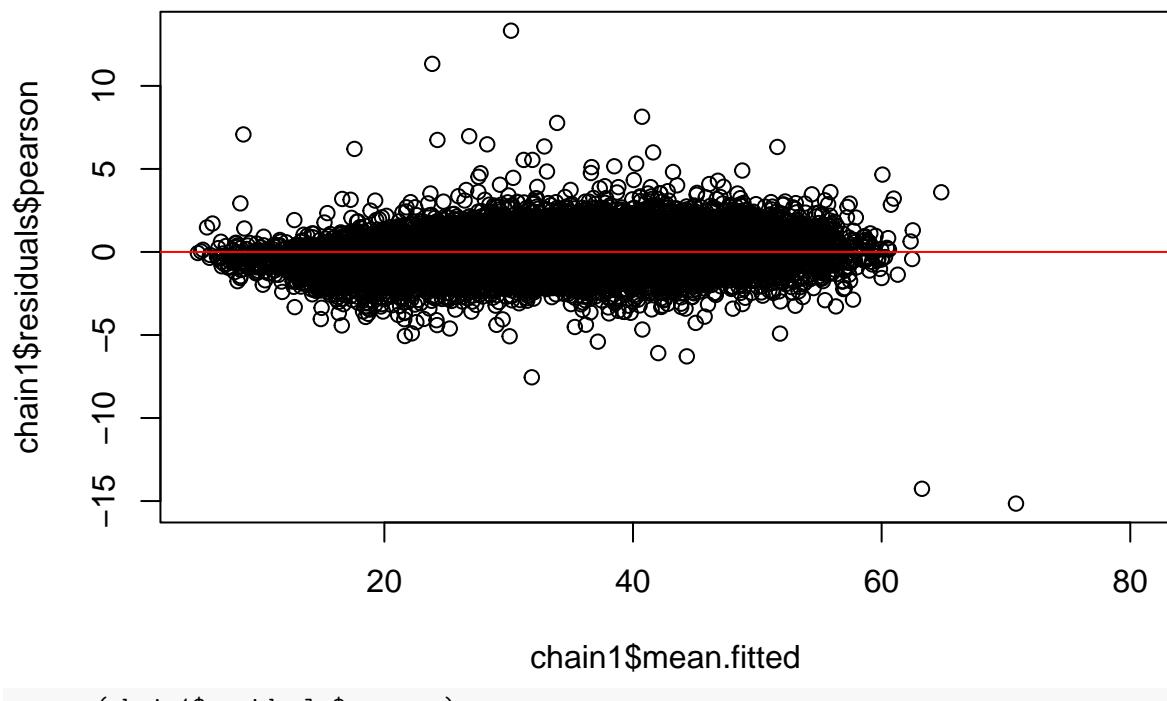
```
load(here("modeling_files/all_census_tract_BPHIGH.RData"))
```

## Model Diagnostics

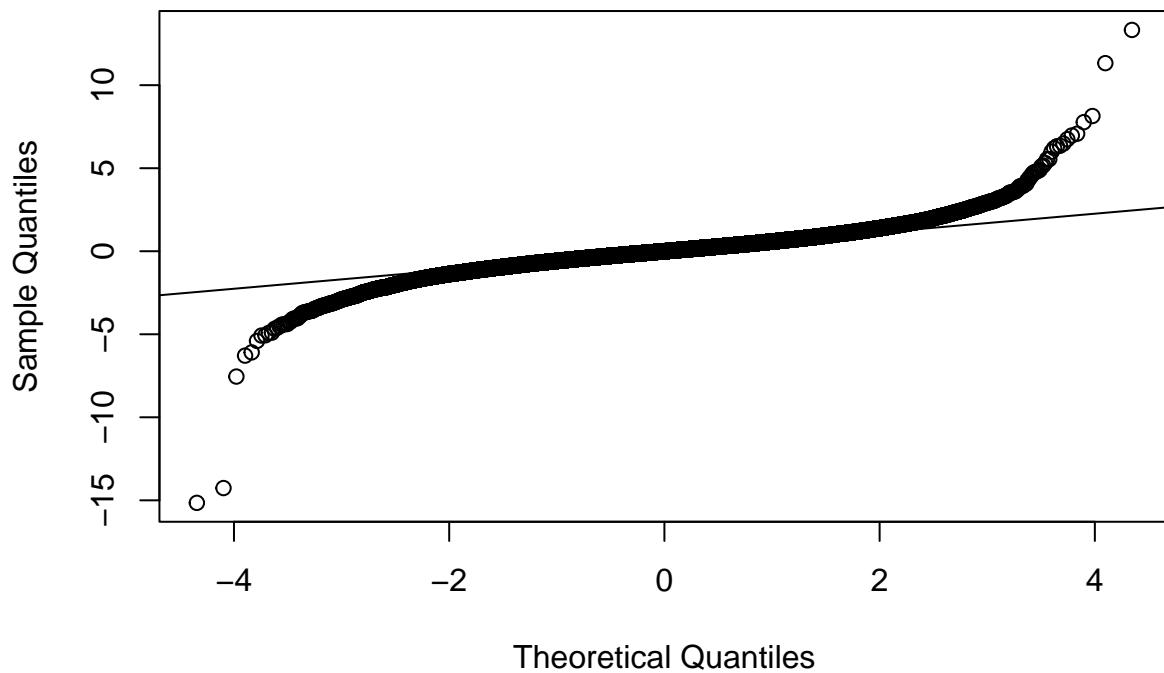
### Residual plot

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

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



**Normal Q-Q Plot**



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
## 81365.75299 24359.87804 21641.78611 32483.81855 41725.04356 46235.27720
##      var7      var8      var9      var10     var11     var12
## 35422.66765 56106.51663 24439.89557 34106.60306 28371.66660 39416.25913
##      var13     var14     var15     var16     var17     var18
## 42721.20650 59400.58202 18159.33426 30310.38112 30146.72542 28710.65889
##      var19     var20     var21     var22     var23     var24
## 43079.07847 24931.20227 55467.73517 34400.84071 7349.14887 3964.49710
##      var25     var26     var27     var28     var29     var30
## 283.20455 3135.47246 1759.83497 2180.69489 616.31626 95.43094
##      var31     var32     var33
## 667.91443 1001.99311 17671.86123

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[, 14:(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            32.32392 32.31256 32.33533
## flood_risk_pc1      -0.01103 -0.03940  0.01716
## flood_risk_pc2       0.08580  0.05447  0.11774
## flood_risk_pc3       0.05210  0.02818  0.07594
## flood_risk_pc4       0.04513  0.02348  0.06673
## flood_risk_pc5       0.01049 -0.01022  0.03116
## EP_POV                0.03662 -0.00261  0.07578
## EP_UNEMP               0.12897  0.10390  0.15423
## EP_PCI                 0.25434  0.21583  0.29269
## EP_NOHSDP              -0.00415 -0.05369  0.04537
## EP_AGE65                4.06727  4.03511  4.09978
## EP_AGE17                0.60886  0.57499  0.64253
## EP_DISABL               0.74145  0.70960  0.77326
## EP_SNGPNT              -0.04888 -0.07787 -0.01989
## EP_MINRTY                2.64375  2.59289  2.69433
## EP_LIMENG                -0.88653 -0.93139 -0.84191
## EP_MUNIT                -0.60918 -0.63801 -0.58065
## EP_MOBILE                  0.11822  0.09225  0.14449
## EP_CROWD                 -0.12525 -0.15847 -0.09194
## EP_NOVEH                  0.61201  0.56898  0.65505
## EP_GROUPQ                 -0.58980 -0.61208 -0.56745
## EP_UNINSUR                  0.23343  0.20043  0.26629
## co                      -0.69335 -0.81656 -0.57007
## no2                     -0.67248 -0.83643 -0.51117
## o3                      -0.48660 -0.76332 -0.19809
## pm10                    -0.49384 -0.60294 -0.38447
## pm25                      1.19286  1.02647  1.35664
## so2                      0.03881 -0.07442  0.15178
## summer_tmmx                0.08632 -0.11593  0.28372
## winter_tmmx                -0.68150 -1.24552 -0.08008

```

```

## summer_rmax      -0.27418 -0.50909 -0.05106
## winter_rmax      0.17489  0.00219  0.35586
## Data_Value_CSMOKING 2.52689  2.46176  2.59163

```

List of significant beta coefficients:

```

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

## [1] "Intercept"           "flood_risk_pc2"       "flood_risk_pc3"
## [4] "flood_risk_pc4"       "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")

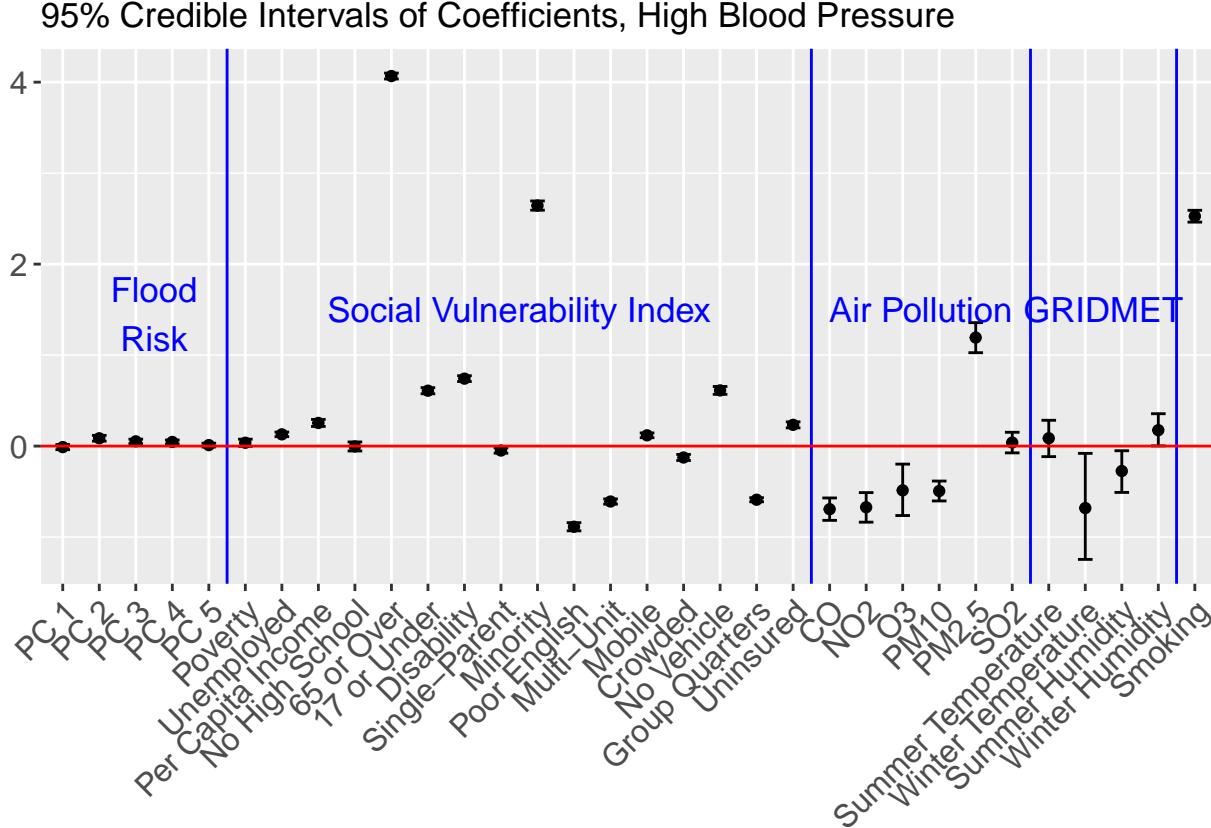
```

```

"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

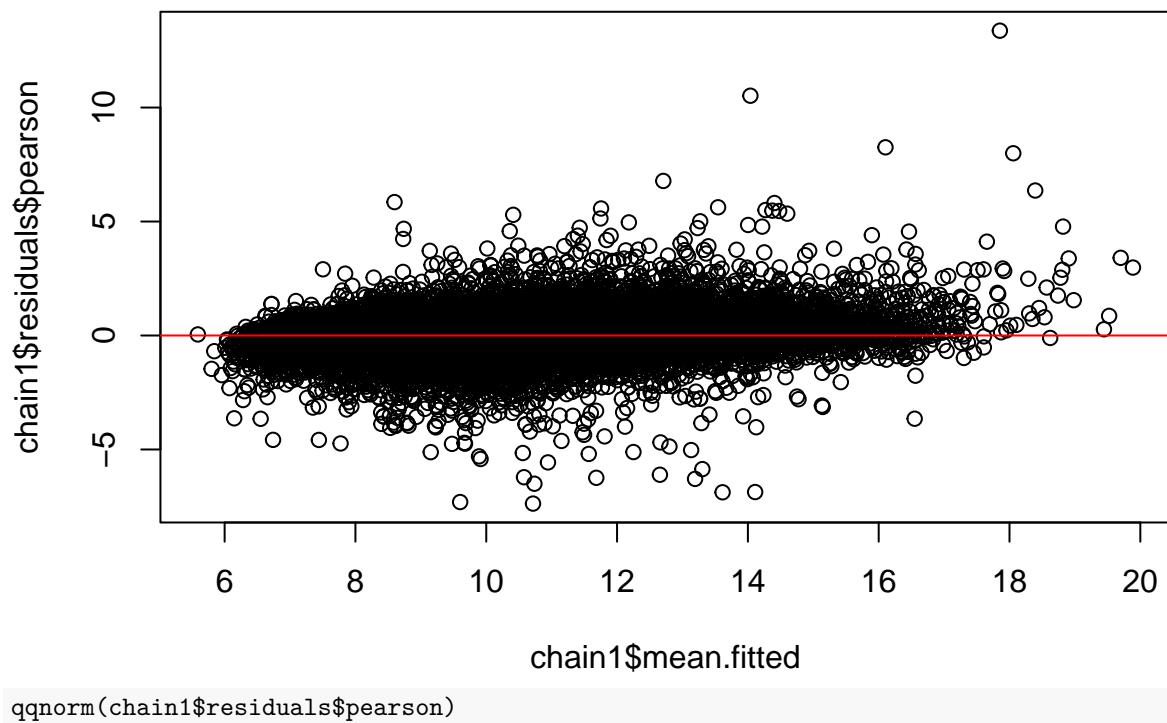
### Residual plot

```

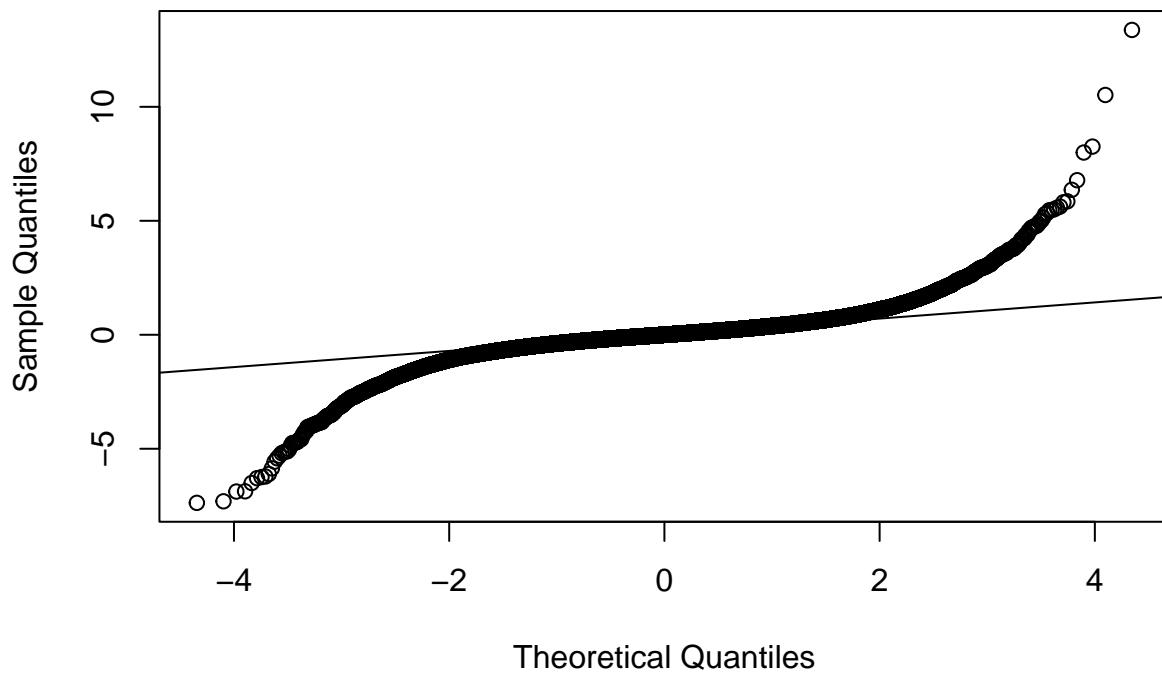
plot(chain1$mean.fitted, chain1$residuals$pearson)

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

```



**Normal Q-Q Plot**



## Beta samples

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

effectiveSize(beta_samples)

##      var1      var2      var3      var4      var5      var6
## 51943.43137 16523.54519 15591.76403 21447.04765 27394.36842 29701.50985
##      var7      var8      var9      var10     var11     var12
## 23952.31896 36225.15120 17787.75495 23477.59681 22799.70159 25975.55548
##      var13     var14     var15     var16     var17     var18
## 29883.75279 38625.71462 12859.33267 20074.06441 21296.50122 22681.76182
##      var19     var20     var21     var22     var23     var24
## 28301.87476 18152.75286 35584.42075 23691.85246 5200.11078 2900.93004
##      var25     var26     var27     var28     var29     var30
## 208.18189 2264.30301 1277.82153 1505.23299 449.78664 65.71628
##      var31     var32     var33
## 489.65505 736.16666 12777.34988

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[, 14:(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                9.89828  9.89673  9.89982
## flood_risk_pc1            0.00248 -0.00241  0.00733
## flood_risk_pc2            0.00548  0.00009  0.01098
## flood_risk_pc3            -0.00206 -0.00614  0.00201
## flood_risk_pc4            -0.00624 -0.00993 -0.00256
## flood_risk_pc5            0.00102 -0.00250  0.00453
## EP_POV                    0.28285  0.27620  0.28950
## EP_UNEMP                  0.07061  0.06641  0.07486
## EP_PCI                    -0.06382 -0.07037 -0.05723
## EP_NOHSDP                 0.07436  0.06595  0.08272
## EP_AGE65                  0.09780  0.09241  0.10327
## EP_AGE17                  0.00979  0.00406  0.01548
## EP_DISABL                 -0.06092 -0.06630 -0.05558
## EP_SNGPNT                 0.03876  0.03389  0.04365
## EP_MINRTY                 0.32439  0.31561  0.33301
## EP_LIMENG                 -0.25507 -0.26270 -0.24742
## EP_MUNIT                  -0.02228 -0.02717 -0.01744
## EP_MOBILE                 -0.01547 -0.01987 -0.01101
## EP_CROWD                  -0.02053 -0.02615 -0.01488
## EP_NOVEH                  0.15001  0.14272  0.15729
## EP_GROUPQ                 -0.04568 -0.04942 -0.04189
## EP_UNINSUR                -0.02823 -0.03381 -0.02268
## co                        -0.06309 -0.08454 -0.04155
## no2                       -0.16953 -0.19855 -0.14091
## o3                         0.01180 -0.03791  0.06602
## pm10                      -0.16118 -0.18043 -0.14178
## pm25                      0.29706  0.26737  0.32599
## so2                        -0.02466 -0.04486 -0.00424
## summer_tmmx                0.03332 -0.00439  0.07000
## winter_tmmx               -0.08875 -0.19221  0.02389

```

```

## summer_rmax      0.00684 -0.03557  0.04684
## winter_rmax     -0.05814 -0.08943 -0.02523
## Data_Value_CSMOKING  0.85522  0.84410  0.86619

```

List of significant beta coefficients:

```

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

## [1] "Intercept"           "flood_risk_pc2"       "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")

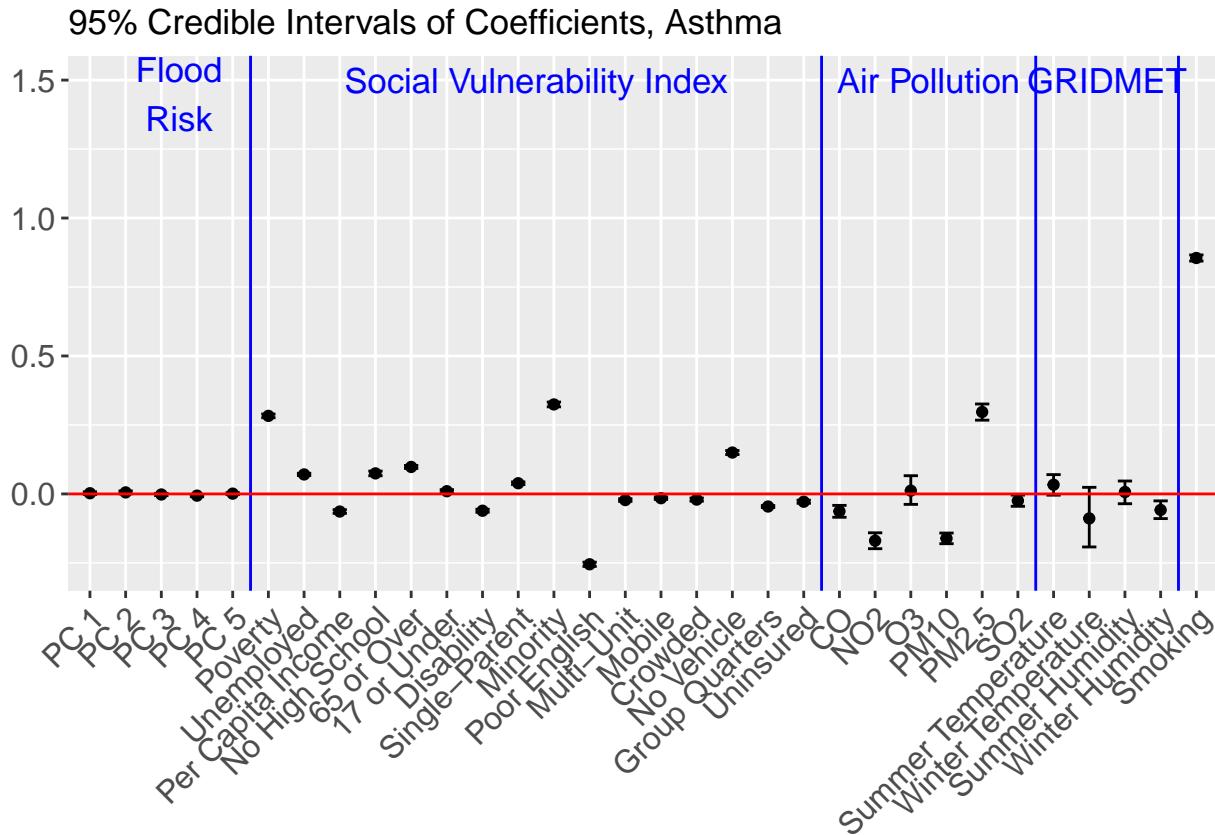
```

```

"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

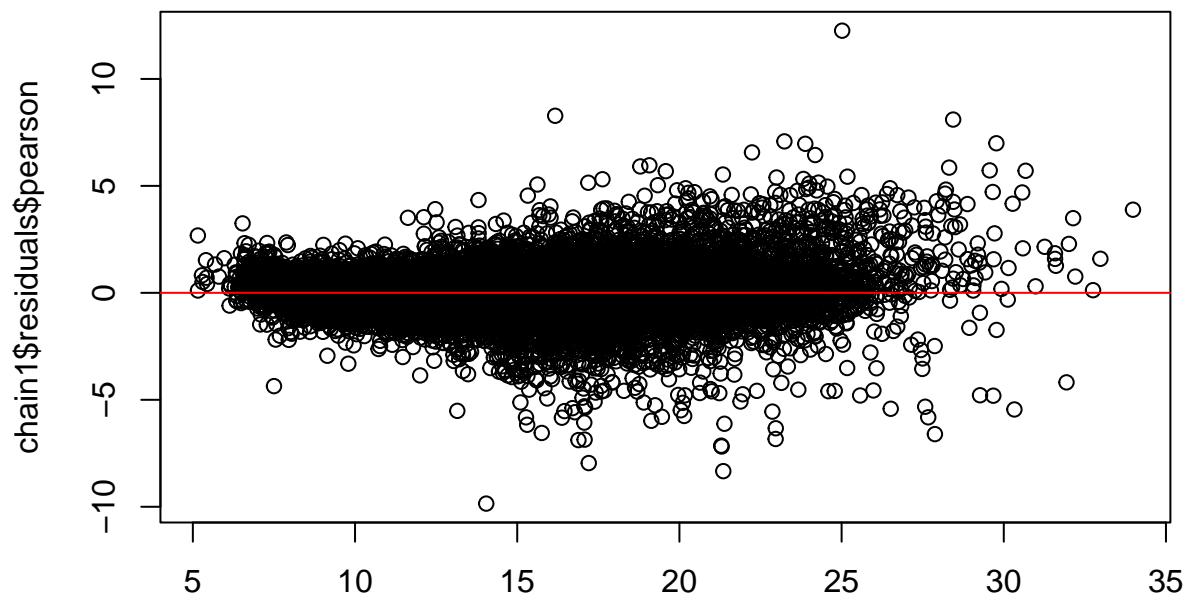
### Residual plot

```

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

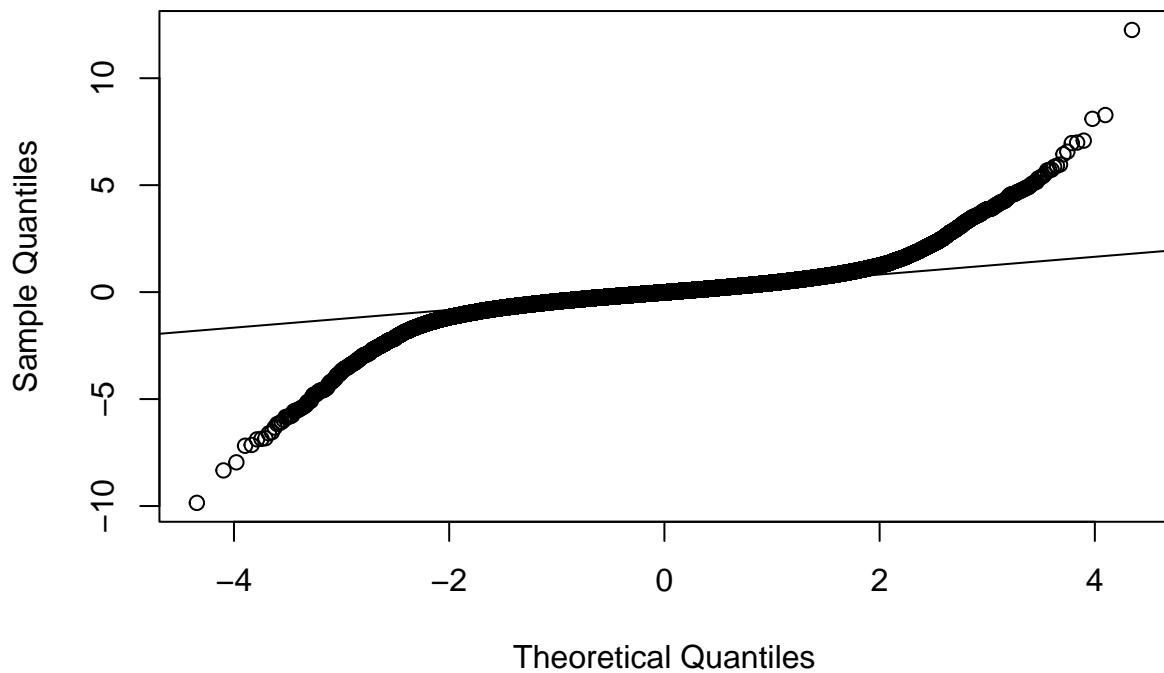
abline(h = 0, col = "red")

```



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

**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
## 67980.11325 21231.73835 19117.24945 28043.13408 35791.10914 39227.90126
##      var7      var8      var9      var10     var11     var12
## 25900.75078 47910.13901 21371.56956 29516.37994 28660.19485 33688.20805
##      var13     var14     var15     var16     var17     var18
## 38687.11232 50674.31921 16382.39450 26481.50121 27280.17904 25572.89472
##      var19     var20     var21     var22     var23     var24
## 37296.64258 22269.16005 47077.31218 30195.33882 6510.56977 3536.60772
##      var25     var26     var27     var28     var29     var30
## 253.26017 2799.12468 1575.93980 1903.32660 547.67411 83.38725
##      var31     var32     var33
## 609.01429 916.58620 15205.46878

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[, 14:(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                14.26585 14.26294 14.26877
## flood_risk_pc1            0.00136 -0.00660  0.00926
## flood_risk_pc2            0.00314 -0.00562  0.01213
## flood_risk_pc3            -0.00527 -0.01195  0.00140
## flood_risk_pc4            -0.00839 -0.01441 -0.00236
## flood_risk_pc5            0.00038 -0.00539  0.00615
## EP_POV                    0.81856  0.80744  0.82965
## EP_UNEMP                  0.07105  0.06410  0.07806
## EP_PCI                    -0.32980 -0.34058 -0.31904
## EP_NOHSDP                 0.25407  0.24026  0.26787
## EP_AGE65                  -0.33632 -0.34519 -0.32735
## EP_AGE17                  -0.10562 -0.11505 -0.09625
## EP_DISABL                 -0.17483 -0.18370 -0.16601
## EP_SNGPNT                  0.07612  0.06805  0.08418
## EP_MINRTY                 -0.17895 -0.19319 -0.16485
## EP_LIMENG                  -0.03136 -0.04390 -0.01890
## EP_MUNIT                   0.08007  0.07202  0.08800
## EP_MOBILE                  -0.01432 -0.02155 -0.00699
## EP_CROWD                   0.05844  0.04917  0.06774
## EP_NOVEH                   0.07669  0.06467  0.08867
## EP_GROUPQ                  0.16887  0.16268  0.17509
## EP_UNINSUR                 -0.02150 -0.03070 -0.01237
## co                         0.08173  0.04693  0.11660
## no2                        -0.05098 -0.09764 -0.00495
## o3                          0.03040 -0.04903  0.11500
## pm10                       -0.14081 -0.17179 -0.10966
## pm25                       0.30814  0.26063  0.35474
## so2                         0.00424 -0.02811  0.03660
## summer_tmmx                 0.08201  0.02315  0.13930
## winter_tmmx                 0.01848 -0.14471  0.19509

```

```

## summer_rmax          0.05025 -0.01718  0.11404
## winter_rmax         -0.03164 -0.08131  0.02047
## Data_Value_CSMOKING 2.14218  2.12385  2.16032

```

List of significant beta coefficients:

```

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

## [1] "Intercept"           "flood_risk_pc4"      "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",
                             "No Vehicle", "Group Quarters", "Uninsured"))

```

```

"CO", "NO2", "O3", "PM10", "PM2.5", "S02",
"Summer Temperature", "Winter Temperature", "Summer Humidity", "Winter Humidity",
"Smoking")) + ggtitle("95% Credible Intervals of Coefficients, Poor Mental Health")

```

p

95% Credible Intervals of Coefficients, Poor Mental Health

