

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_no_collinear.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.

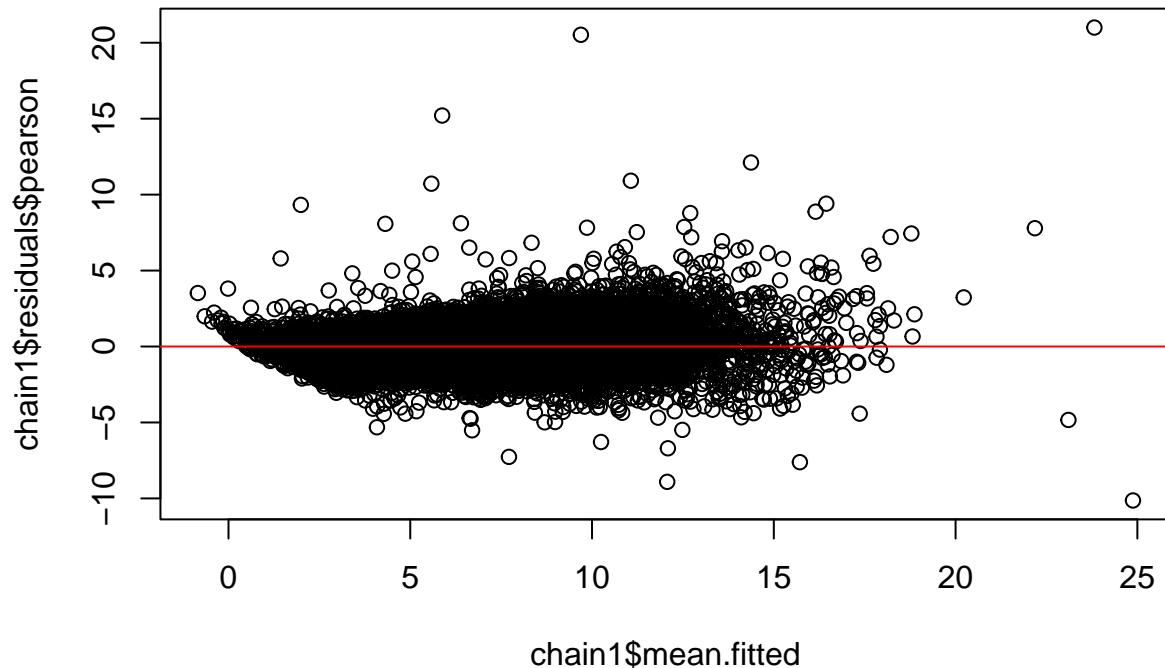
```
load(here("modeling_files/all_census_tract_intrinsic.RData"))
```

Model Diagnostics

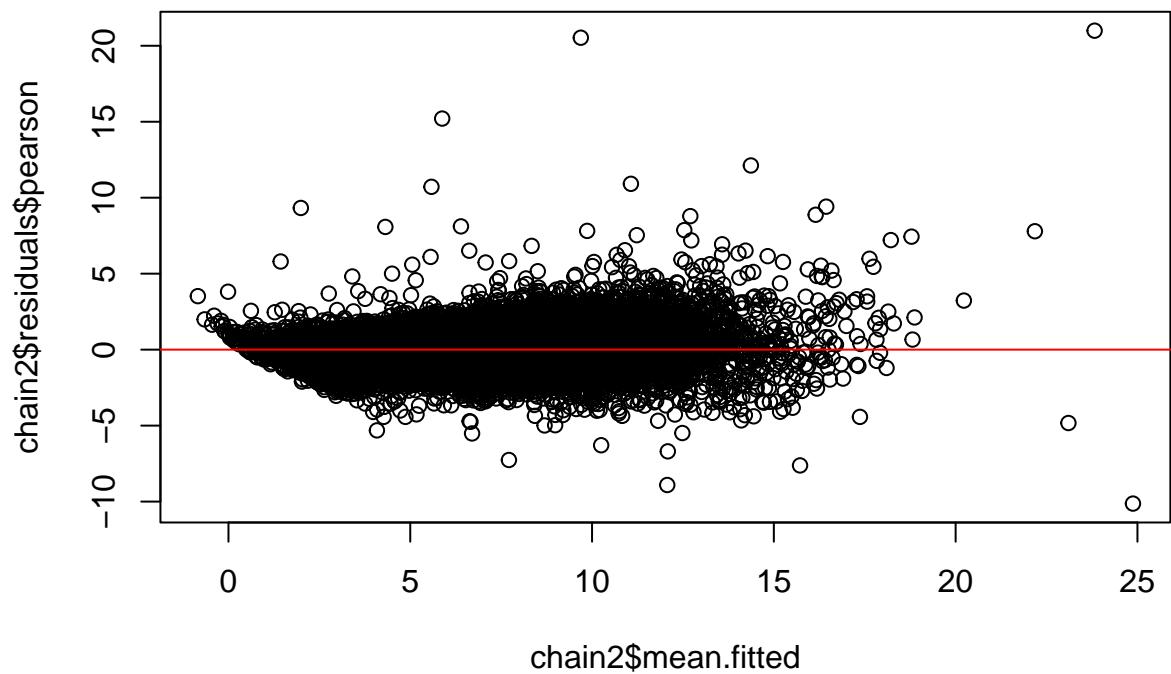
Residual Plots

For each chain

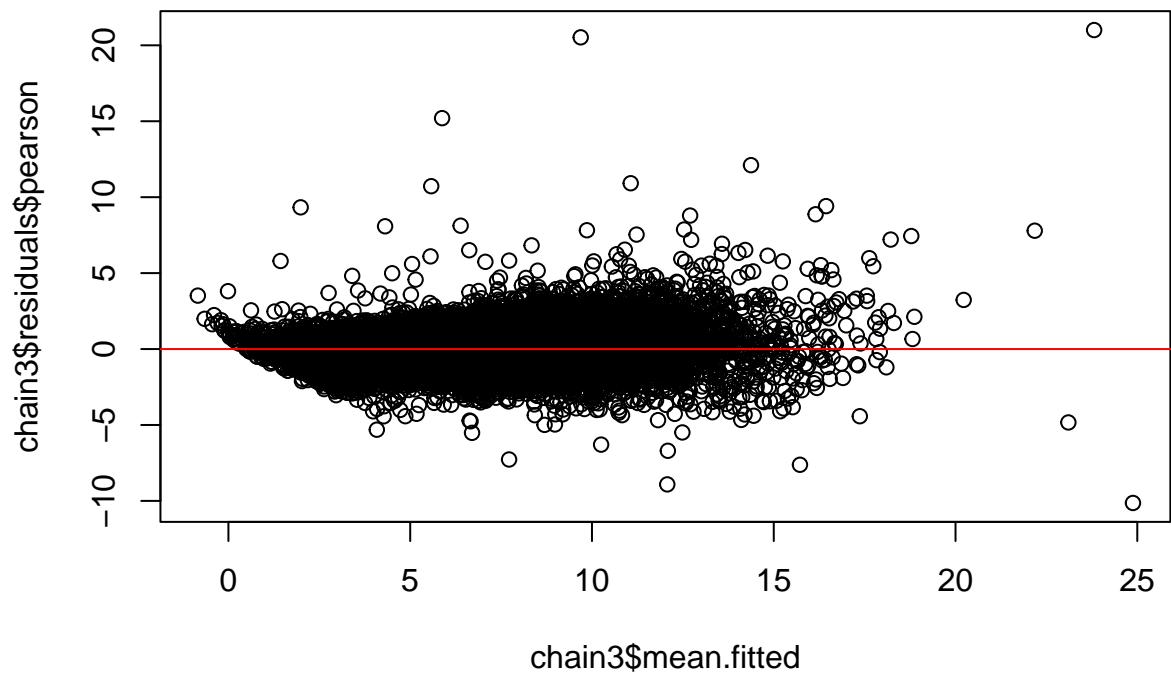
```
plot(chain1$mean.fitted, chain1$residuals$pearson)  
abline(h = 0, col = "red")
```



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

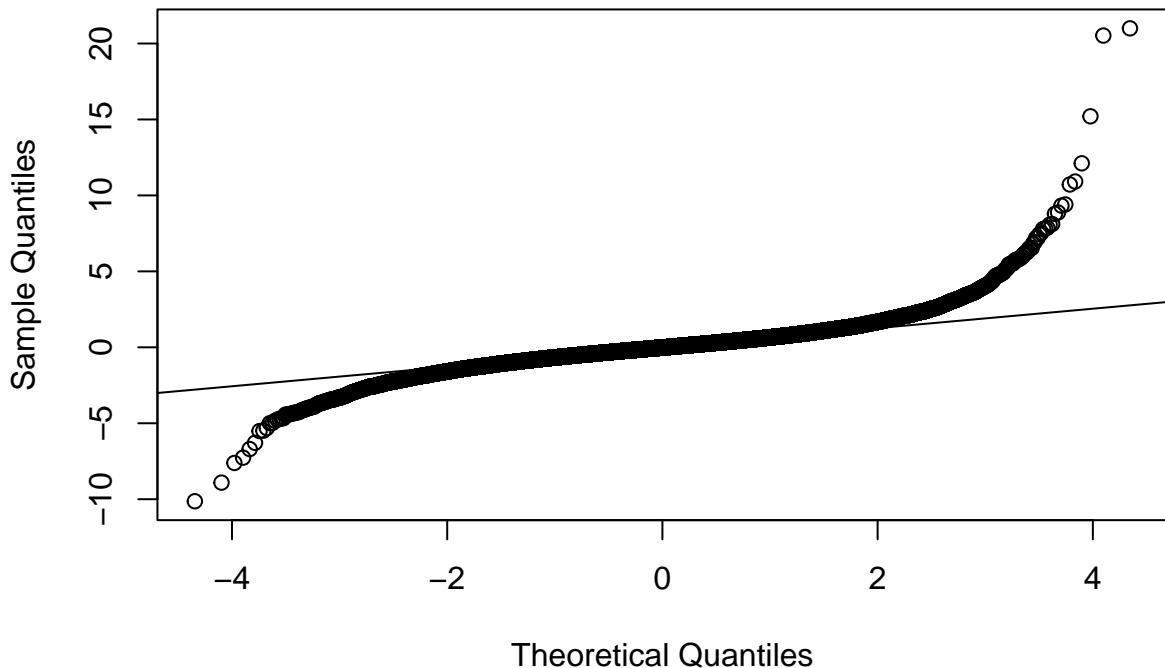


```
plot(chain3$mean.fitted, chain3$residuals$pearson)  
abline(h = 0, col = "red")
```



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

Normal Q-Q Plot



Theoretical Quantiles

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
## 122756.2924  23957.7010  26952.3897  34969.1160  67500.8773  54889.6529
##          var7          var8          var9          var10         var11         var12
##  39252.7768  71142.4053  29907.5737  40588.7259  34875.3078  49558.5771
##          var13         var14         var15         var16         var17         var18
##  47248.2948  74057.6859  20763.3342  37256.2303  47955.6812  31163.0541
##          var19         var20         var21         var22         var23         var24
##  54434.5715  27697.3134  74480.0752  39401.6901  4082.6737   316.9174
##          var25         var26         var27         var28         var29         var30
##  3080.1394   1888.0073  2693.0664   923.1480   179.6829   702.6014
##          var31         var32
##  1081.5755  19155.6374

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    1.00
## [24,]   1.02    1.06
## [25,]   1.01    1.02
## [26,]   1.00    1.01
## [27,]   1.00    1.01
## [28,]   1.06    1.17
## [29,]   1.14    1.42
## [30,]   1.01    1.03
## [31,]   1.00    1.01
## [32,]   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    1.00
## [2,]    1.01    1.04
## [3,]    1.00    1.01
## [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    1.05
##
## 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              6.66108  6.65680  6.66536
## flood_risk_pc1         -0.01236 -0.02187 -0.00298
## flood_risk_pc2          0.00521 -0.00406  0.01451
## flood_risk_pc3         -0.00899 -0.01646 -0.00157
## flood_risk_pc4         -0.00499 -0.01121  0.00120
## flood_risk_pc5          0.00182 -0.00488  0.00853
## EP_POV                  0.31350  0.30126  0.32570
## EP_UNEMP                 0.03009  0.02212  0.03802
## EP_PCI                  -0.03529 -0.04712 -0.02346
## EP_NOHSDP                0.19433  0.17874  0.20990
## EP_AGE65                 1.38161  1.37151  1.39170
## EP_AGE17                 0.27966  0.26913  0.29032
## EP_DISABLE                0.26992  0.25984  0.28004
## EP_SNGPNT                 -0.06540 -0.07459 -0.05617
## EP_MINRTY                 -0.04126 -0.05671 -0.02583
## EP_LIMENG                 -0.06146 -0.07527 -0.04746
## EP_MUNIT                  -0.05873 -0.06761 -0.04982
## EP_MOBILE                  0.07926  0.07098  0.08753
## EP_CROWD                  -0.04643 -0.05680 -0.03596
## EP_NOVEH                  0.12658  0.11316  0.14025
## EP_GROUPQ                 -0.09391 -0.10092 -0.08683
## EP_UNINSUR                 0.10401  0.09363  0.11444
## co                         -0.16865 -0.19959 -0.13731
## o3                          -0.11945 -0.19409 -0.04418
## pm10                      -0.16287 -0.19504 -0.13067
## pm25                      0.43561  0.38965  0.48068
## so2                        0.03389  0.00214  0.06590
## summer_tmmx                 0.06789  0.01497  0.12082
## winter_tmmx                 -0.23071 -0.37150 -0.09610
## summer_rmax                 -0.04613 -0.11481  0.02203
## winter_rmax                  0.05540  0.00642  0.10728
## Data_Value_CSMOKING       0.78721  0.76697  0.80733
```

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_DISABLE"           "EP_SNGPNT"             "EP_MINRTY"
## [13] "EP_LIMENG"             "EP_MUNIT"              "EP_MOBILE"
## [16] "EP_CROWD"              "EP_NOVEH"              "EP_GROUPQ"
## [19] "EP_UNINSUR"            "co"                   "o3"
## [22] "pm10"                  "pm25"                  "so2"
## [25] "summer_tmmx"           "winter_tmmx"            "winter_rmax"
## [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))

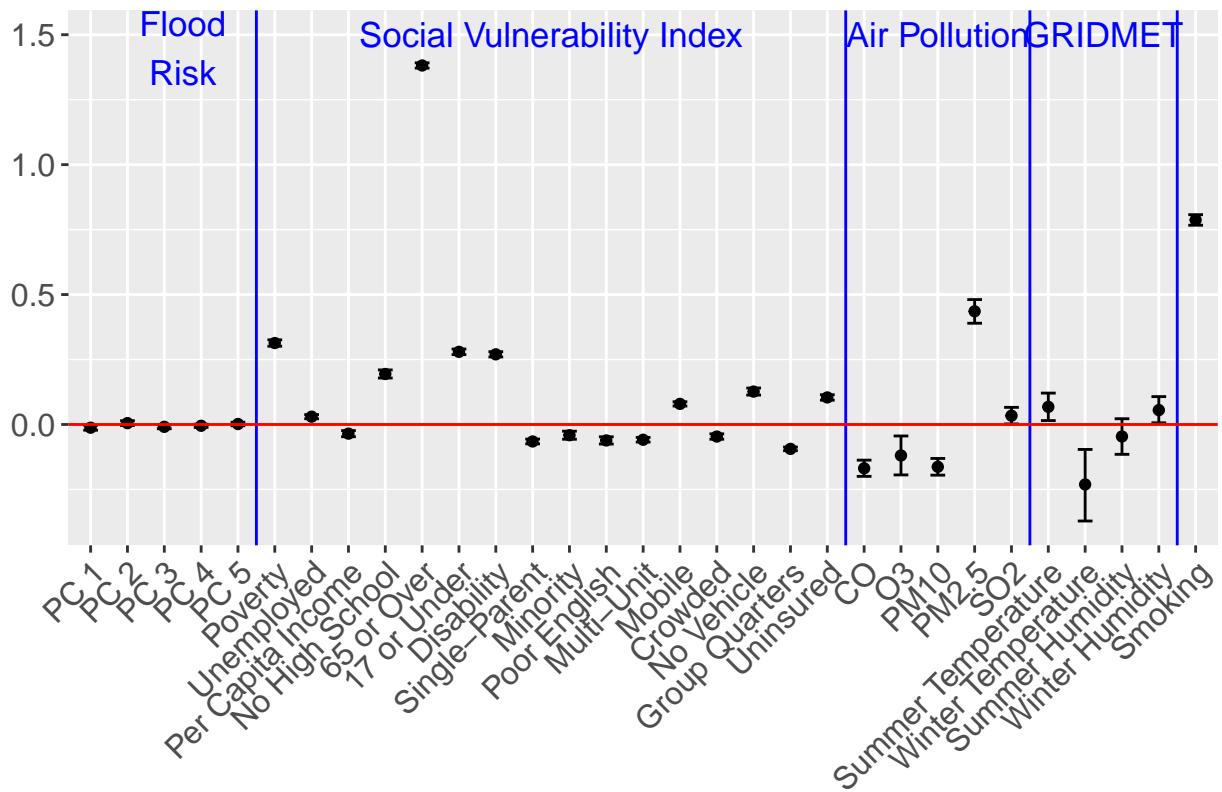
beta_inference_df <- rename(beta_inference_df,
                           post_median = `50%`,
                           post_2.5 = `2.5%`,
                           post_97.5 = `97.5%`)

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, 26.5, 30.5), 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, y = 1.5, label = "Air Pollution",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 28.5, 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", "O3", "PM10", "PM2.5", "SO2",
                             "Summer Temperature", "Winter Temperature", "Summer Humidity", "Winter Humidity",
                             "Smoking")) + ggtitle("95% Credible Intervals of Coefficients, Coronary Heart Disease Risk Model")
```

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.

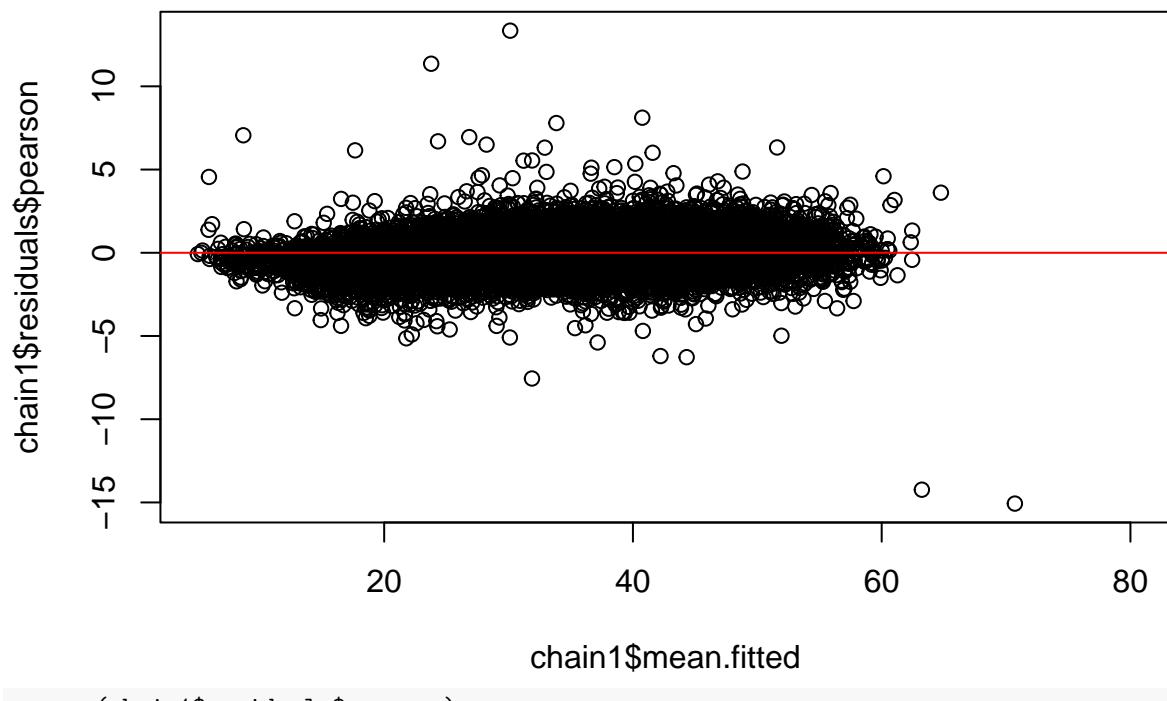
```
load(here("modeling_files/all_census_tract_BPHIGH.RData"))
```

Model Diagnostics

Residual plot

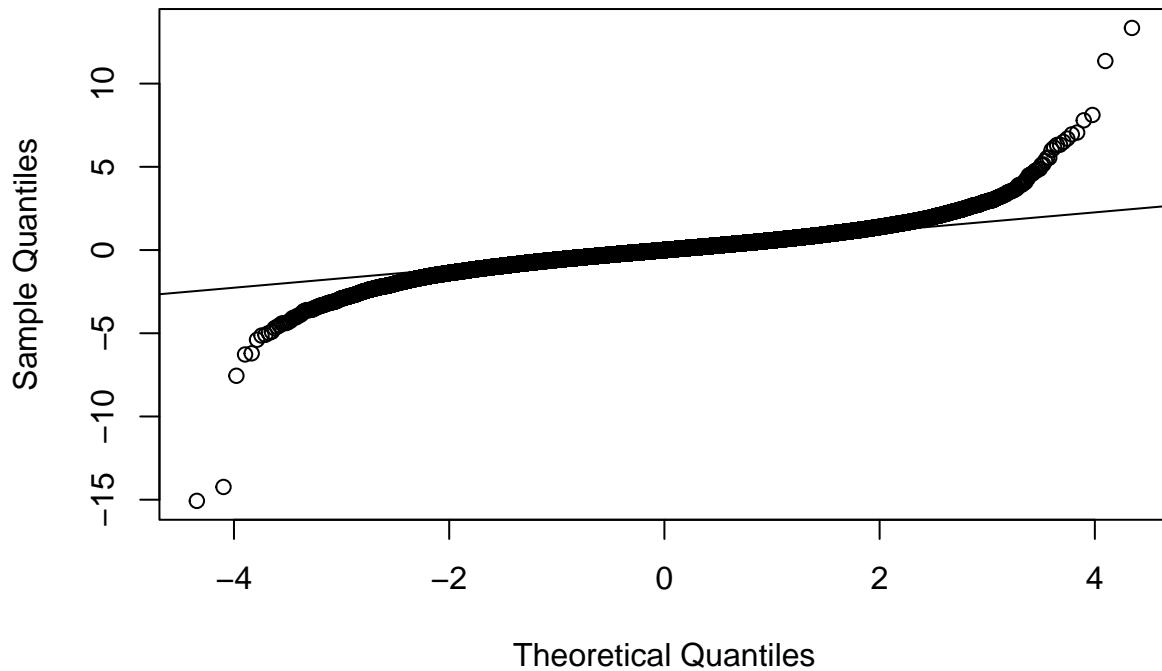
```
plot(chain1$mean.fitted, chain1$residuals$pearson)

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



```
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      var7
## 92155.2555 22015.7275 21956.3006 28443.4919 52398.1148 43784.8534 31557.7791
##      var8      var9      var10     var11     var12     var13     var14
## 54935.3187 26436.2865 32903.0524 27836.2836 39295.6259 40332.5946 59066.7927
##      var15     var16     var17     var18     var19     var20     var21
## 16548.2713 30279.0237 37215.0422 28528.5453 43671.8235 24185.3100 54820.8439
##      var22     var23     var24     var25     var26     var27     var28
## 35044.0622 3746.2059   255.2251 2701.4078 1662.7064 2221.6522  751.2948
##      var29     var30     var31     var32
## 142.1019 599.4142   880.5075 16301.8873

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    1.00
## [24,]   1.02    1.07
## [25,]   1.01    1.03
## [26,]   1.00    1.01
## [27,]   1.00    1.00
## [28,]   1.07    1.23
## [29,]   1.17    1.52
## [30,]   1.01    1.03
```

```

## [31,]      1.01      1.02
## [32,]      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,]      1.00      1.00
## [2,]      1.02      1.06
## [3,]      1.00      1.01
## [4,]      1.01      1.02

```

```

## [5,]    1.00    1.00
## [6,]    1.00    1.00
## [7,]    1.00    1.01
## [8,]    1.00    1.00
## [9,]    1.00    1.00
## [10,]   1.02    1.06
##
## 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            32.32500 32.31359 32.33638
## flood_risk_pc1      -0.09032 -0.12098 -0.06004
## flood_risk_pc2      -0.03403 -0.06428 -0.00361
## flood_risk_pc3      -0.04685 -0.07092 -0.02289
## flood_risk_pc4      -0.00342 -0.02315  0.01635
## flood_risk_pc5      0.01428 -0.00702  0.03575
## EP_POV               0.03201 -0.00730  0.07130
## EP_UNEMP              0.12850  0.10342  0.15341
## EP_PCI                0.25805  0.21983  0.29639
## EP_NOHSDP             -0.00339 -0.05294  0.04642
## EP_AGE65              4.07122  4.03877  4.10364
## EP_AGE17              0.61764  0.58414  0.65132
## EP_DISABL             0.73950  0.70764  0.77150
## EP_SNGPNT             -0.05149 -0.08040 -0.02242
## EP_MINRTY              2.62682  2.57586  2.67773
## EP_LIMENG             -0.88539 -0.92978 -0.84014
## EP_MUNIT              -0.62209 -0.65053 -0.59366
## EP_MOBILE              0.11857  0.09215  0.14478
## EP_CROWD              -0.12541 -0.15835 -0.09221
## EP_NOVEH              0.61253  0.56952  0.65623
## EP_GROUPQ              -0.58624 -0.60851 -0.56388
## EP_UNINSUR             0.23189  0.19904  0.26496
## co                     -0.95977 -1.06385 -0.85298
## o3                     -0.21331 -0.48407  0.06061
## pm10                  -0.54799 -0.65826 -0.43770
## pm25                   0.94924  0.79192  1.10466
## so2                     0.02227 -0.08750  0.13381
## summer_tmmx             0.05790 -0.13394  0.24938
## winter_tmmx             -0.49601 -1.01597 -0.01400
## summer_rmax             -0.26632 -0.50983 -0.02422
## winter_rmax              0.24368  0.06861  0.42995
## Data_Value_CSMOKING  2.53408  2.46872  2.59928

```

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"                   "pm10"                 "pm25"
## [22] "winter_tmmx"          "summer_rmax"           "winter_rmax"
## [25] "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))

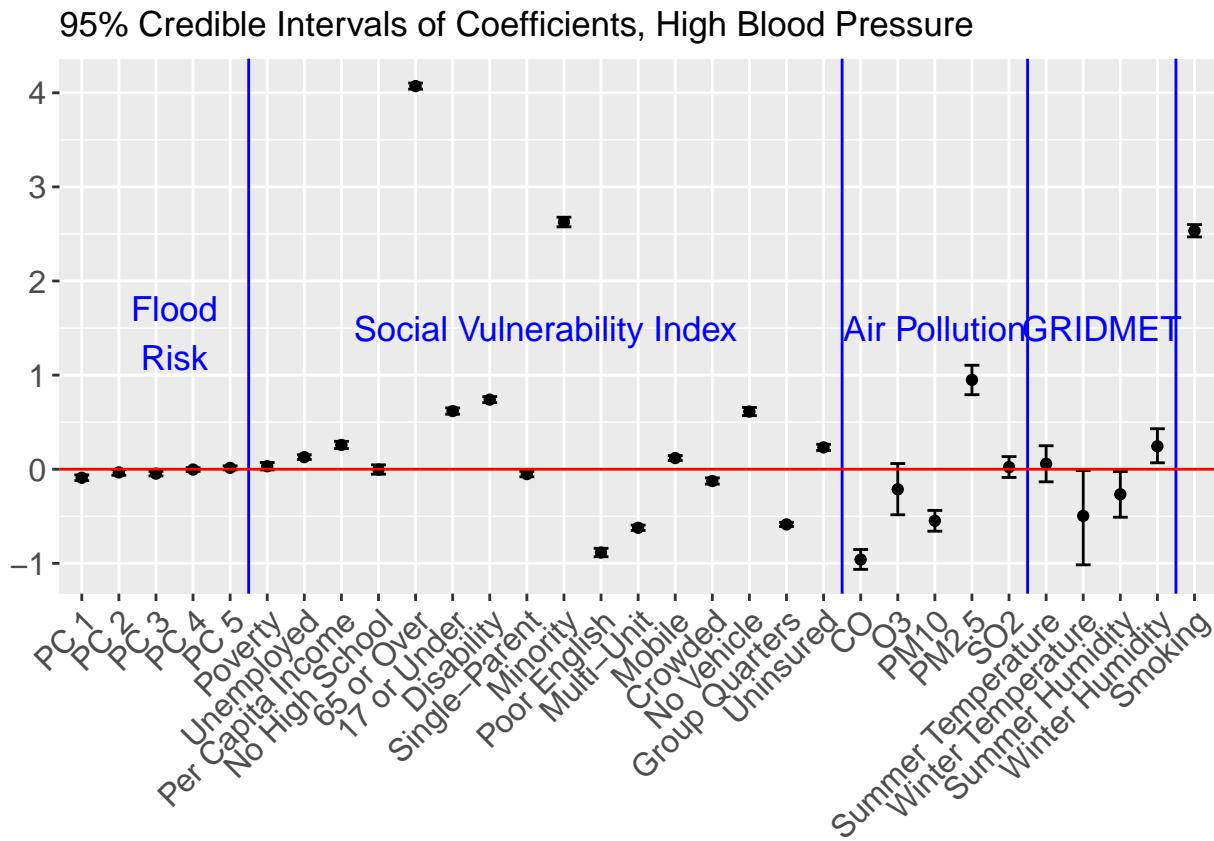
beta_inference_df <- rename(beta_inference_df,
                             post_median = `50%`,
                             post_2.5 = `2.5%`,
                             post_97.5 = `97.5`)

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, 26.5, 30.5), 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, y = 1.5, label = "Air Pollution",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 28.5, 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", "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

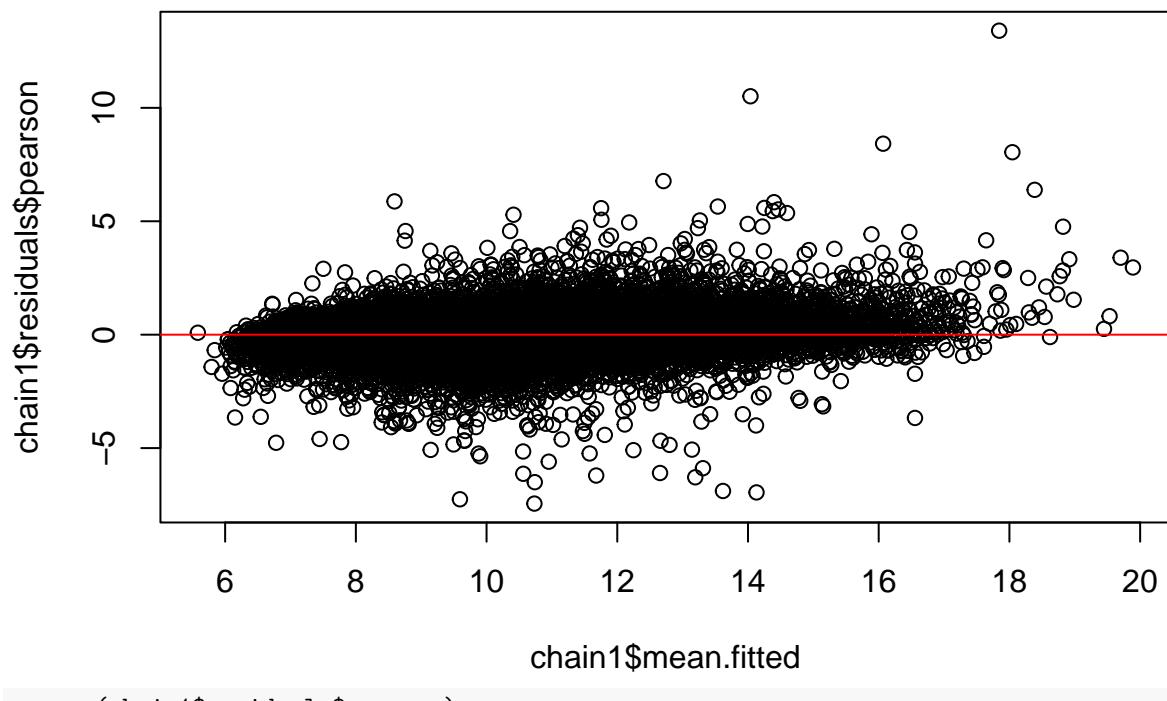
Inference is based on 3 markov chains, each of which has been run for 110000 samples, the first 10000 of which has been removed for burn-in. The remaining 100000 samples are thinned by 2, resulting in 150000 samples for inference across the 3 Markov chains.

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

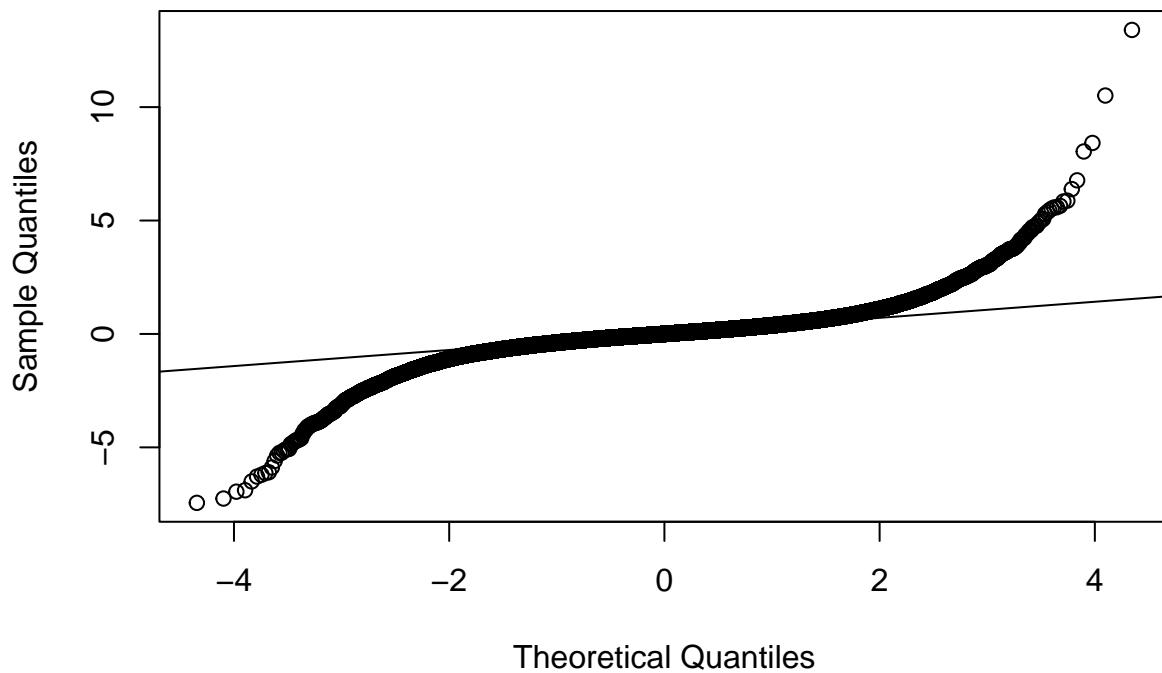
Model Diagnostics

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      var7
## 52429.8651 16289.7907 15169.1783 19775.2647 34243.9669 28956.2837 22970.4977
##      var8      var9      var10     var11     var12     var13     var14
## 35264.7069 18595.3460 22420.7373 20983.4445 25928.0727 28250.5393 38496.5711
##      var15     var16     var17     var18     var19     var20     var21
## 11744.7324 19849.7710 25058.2639 20746.2989 28700.8311 17160.7919 34840.6785
##      var22     var23     var24     var25     var26     var27     var28
## 24004.8506 3085.9677 179.2329 2069.3208 1263.0059 1560.1331 554.5321
##      var29     var30     var31     var32
## 105.0920 450.2868 622.2020 11892.9445

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,]   1.03    1.10
## [25,]   1.01    1.04
## [26,]   1.01    1.02
## [27,]   1.00    1.01
## [28,]   1.12    1.36
## [29,]   1.25    1.79
## [30,]   1.01    1.04
```

```

## [31,]      1.01      1.02
## [32,]      1.00      1.00
##
## Multivariate psrf
##
## 1.17

```

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,]      1.03      1.10
## [3,]      1.01      1.03
## [4,]      1.01      1.05

```

```

## [5,]    1.00    1.00
## [6,]    1.00    1.01
## [7,]    1.00    1.01
## [8,]    1.00    1.01
## [9,]    1.00    1.00
## [10,]   1.03    1.11
##
## Multivariate psrf
##
## 1.06

```

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                  9.89855  9.89700  9.90010
## flood_risk_pc1             -0.00548 -0.01072 -0.00030
## flood_risk_pc2             -0.00377 -0.00899  0.00147
## flood_risk_pc3              0.00790  0.00380  0.01198
## flood_risk_pc4              0.00338  0.00004  0.00673
## flood_risk_pc5             -0.00005 -0.00366  0.00358
## EP_POV                      0.28167  0.27500  0.28832
## EP_UNEMP                     0.07053  0.06630  0.07472
## EP_PCI                      -0.06281 -0.06931 -0.05626
## EP_NOHSDP                    0.07440  0.06597  0.08284
## EP_AGE65                     0.09870  0.09324  0.10417
## EP_AGE17                     0.01175  0.00610  0.01741
## EP_DISABL                   -0.06129 -0.06668 -0.05589
## EP_SNGPNT                    0.03824  0.03336  0.04310
## EP_MINRTY                   0.32047  0.31172  0.32924
## EP_LIMENG                   -0.25483 -0.26241 -0.24711
## EP_MUNIT                     -0.02552 -0.03035 -0.02069
## EP_MOBILE                    -0.01538 -0.01987 -0.01094
## EP_CROWD                     -0.02083 -0.02638 -0.01522
## EP_NOVEH                     0.15024  0.14291  0.15764
## EP_GROUPQ                    -0.04480 -0.04855 -0.04105
## EP_UNINSUR                  -0.02863 -0.03418 -0.02305
## co                           -0.12867 -0.14682 -0.10972
## o3                           0.07925  0.02887  0.13011
## pm10                        -0.17603 -0.19562 -0.15638
## pm25                         0.24079  0.21263  0.26832
## so2                          -0.02799 -0.04743 -0.00790
## summer_tmmx                  0.02541 -0.00985  0.06079
## winter_tmmx                 -0.04140 -0.13766  0.04811
## summer_rmax                  0.00619 -0.03699  0.04981
## winter_rmax                 -0.04119 -0.07269 -0.00716
## Data_Value_CSMOKING        0.85721  0.84606  0.86835

```

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] "flood_risk_pc4"      "EP_POV"              "EP_UNEMP"
## [7] "EP_PCI"               "EP_NOHSDP"            "EP AGE65"
## [10] "EP AGE17"             "EP_DISABL"            "EP_SNGPNT"
## [13] "EP_MINRTY"            "EP_LIMENG"             "EP_MUNIT"
## [16] "EP_MOBILE"             "EP_CROWD"              "EP_NOVEH"
## [19] "EP_GROUPQ"             "EP_UNINSUR"            "co"
## [22] "o3"                   "pm10"                 "pm25"
## [25] "so2"                  "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))

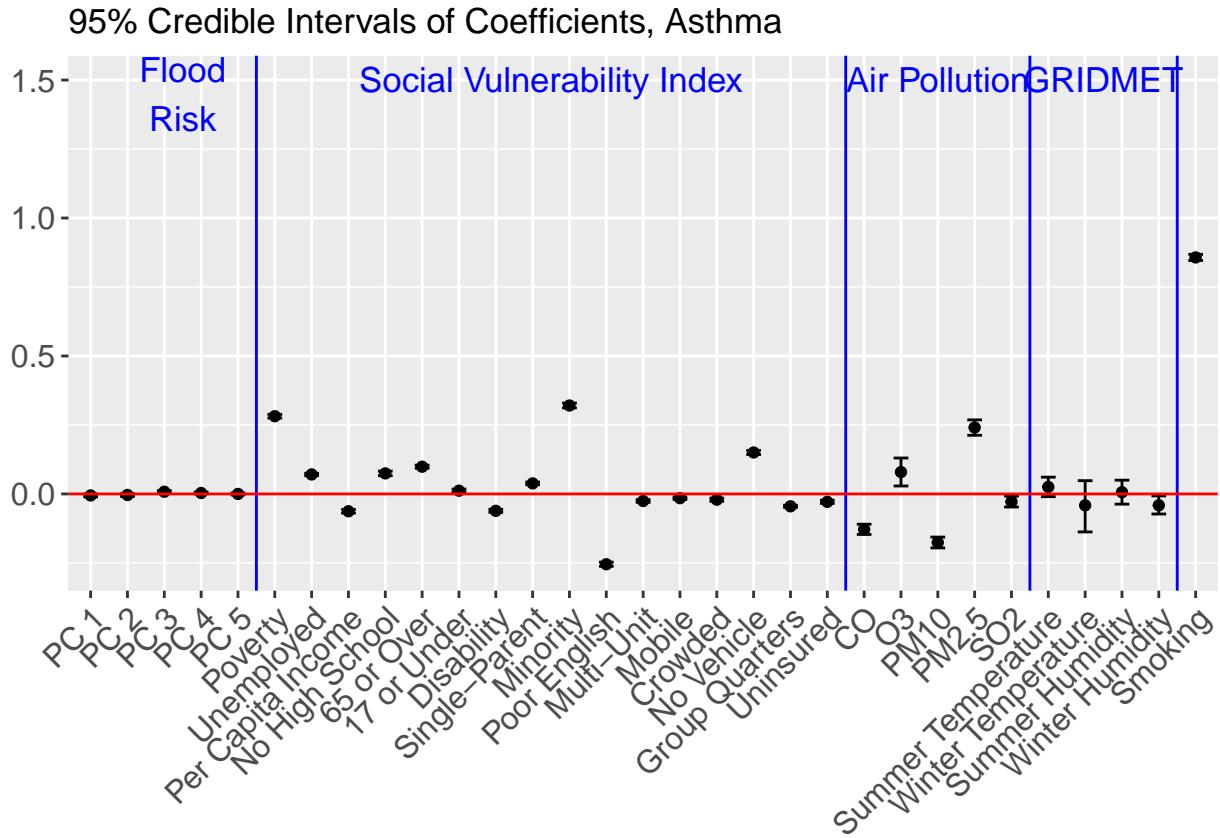
beta_inference_df <- rename(beta_inference_df,
                             post_median = `50%`,
                             post_2.5 = `2.5%`,
                             post_97.5 = `97.5`)

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, 26.5, 30.5), 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, y = 1.5, label = "Air Pollution",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 28.5, 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", "O3", "PM10", "PM2.5", "SO2",
                             "Summer Temperature", "Winter Temperature", "Summer Humidity", "Winter Humidity",
                             "Smoking")) + ggttitle("95% Credible Intervals of Coefficients, Asthma")

```

p



CAR model results, Poor Mental Health

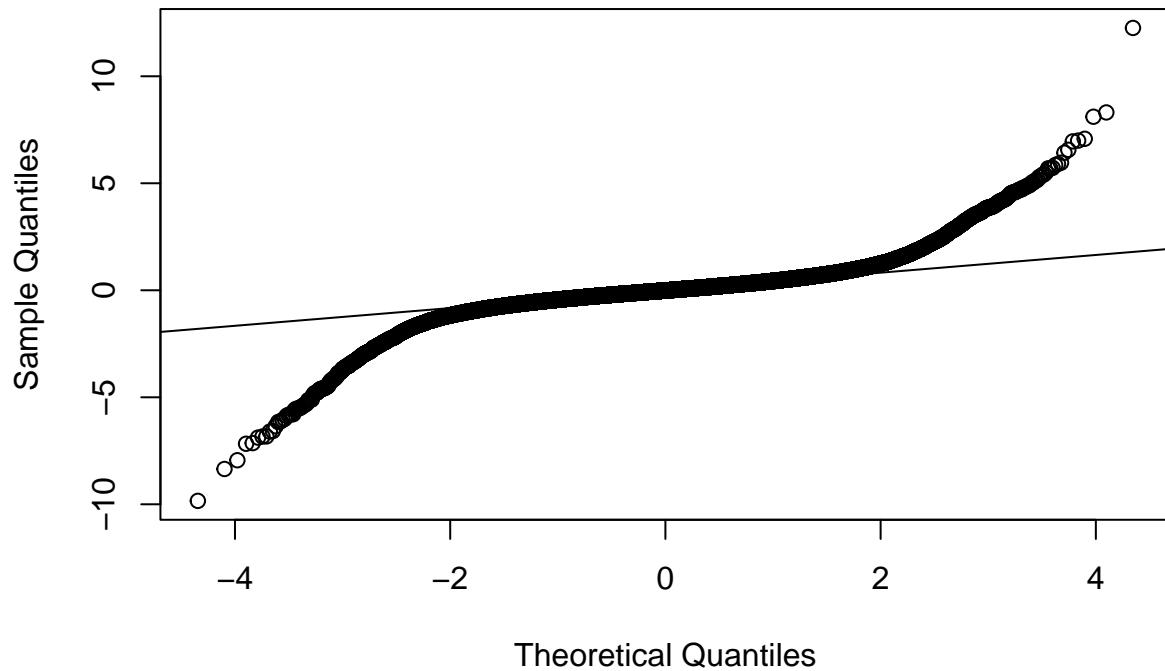
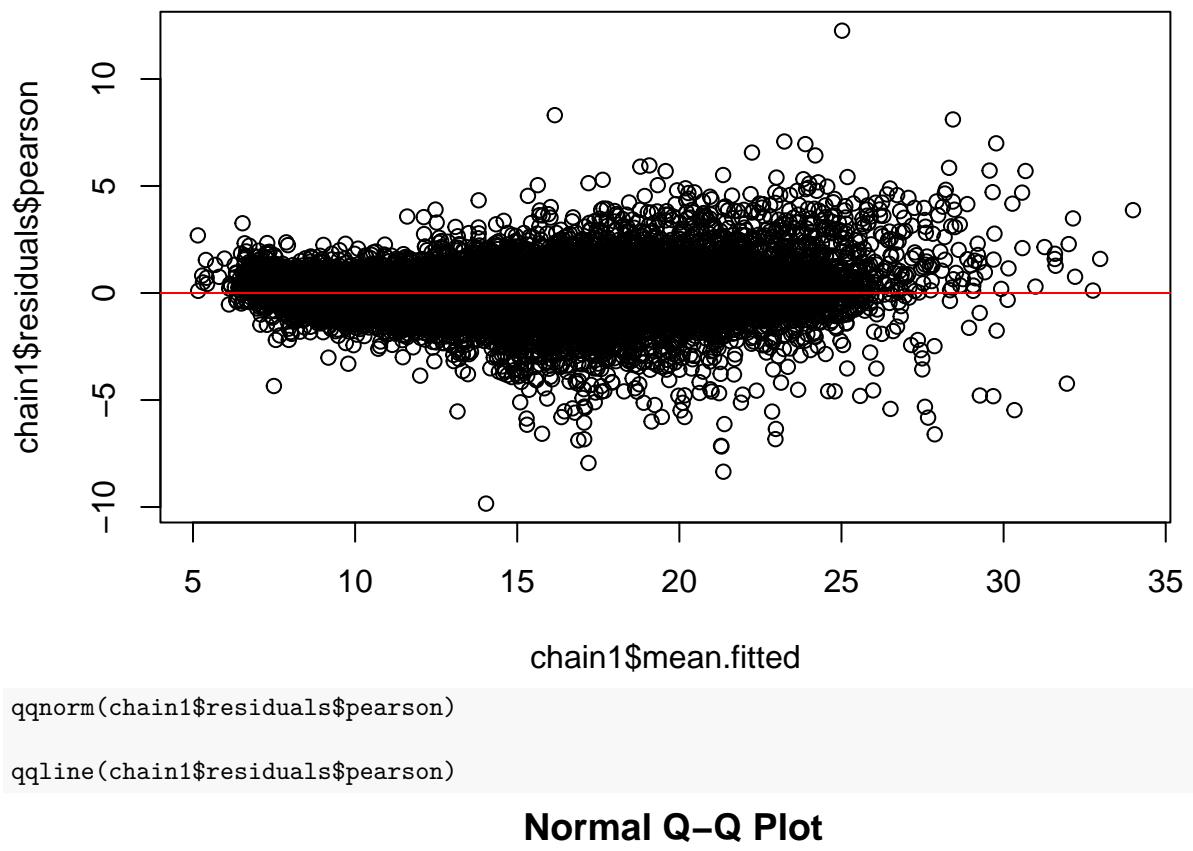
Inference is based on 3 markov chains, each of which has been run for 110000 samples, the first 10000 of which has been removed for burn-in. The remaining 100000 samples are thinned by 2, resulting in 150000 samples for inference across the 3 Markov chains.

```
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
## 71085.2537 19843.5442 19249.0378 24766.8174 44892.4058 37833.0912 24253.8255
##      var8      var9      var10     var11     var12     var13     var14
## 46647.8413 23293.1739 28368.8014 26136.8080 32892.5046 35707.3904 50189.4477
##      var15     var16     var17     var18     var19     var20     var21
## 14877.8633 26313.0934 32372.2044 25516.4302 37351.0010 21447.5568 45684.8259
##      var22     var23     var24     var25     var26     var27     var28
## 30631.9286 3423.2508   222.9628  2422.7028  1576.6693  1952.4391   669.9863
##      var29     var30     var31     var32
## 127.1948 533.4971 773.4855 14106.7597

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,]   1.02    1.08
## [25,]   1.01    1.03
## [26,]   1.01    1.01
## [27,]   1.00    1.01
## [28,]   1.09    1.27
## [29,]   1.19    1.61
## [30,]   1.01    1.03
```

```

## [31,]      1.01      1.02
## [32,]      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,]      1.00      1.01
## [2,]      1.02      1.07
## [3,]      1.00      1.02
## [4,]      1.01      1.03

```

```

## [5,]    1.00    1.00
## [6,]    1.00    1.01
## [7,]    1.00    1.01
## [8,]    1.00    1.01
## [9,]    1.00    1.00
## [10,]   1.02    1.08
##
## 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                  14.26595 14.26304 14.26886
## flood_risk_pc1             -0.00153 -0.01010  0.00695
## flood_risk_pc2             -0.00183 -0.01032  0.00671
## flood_risk_pc3              0.00970  0.00298  0.01638
## flood_risk_pc4              0.00259 -0.00290  0.00808
## flood_risk_pc5             -0.00132 -0.00726  0.00464
## EP_POV                      0.81804  0.80693  0.82915
## EP_UNEMP                     0.07107  0.06410  0.07797
## EP_PCI                      -0.32946 -0.34014 -0.31874
## EP_NOHSDP                    0.25407  0.24026  0.26795
## EP_AGE65                     -0.33590 -0.34486 -0.32690
## EP_AGE17                     -0.10512 -0.11443 -0.09575
## EP_DISABL                    -0.17494 -0.18381 -0.16606
## EP_SNGPNT                     0.07599  0.06797  0.08405
## EP_MINRTY                    -0.18025 -0.19455 -0.16597
## EP_LIMENG                     -0.03130 -0.04370 -0.01866
## EP_MUNIT                      0.07904  0.07112  0.08695
## EP_MOBILE                     -0.01426 -0.02162 -0.00696
## EP_CROWD                      0.05831  0.04916  0.06754
## EP_NOVEH                      0.07675  0.06470  0.08894
## EP_GROUPQ                     0.16915  0.16297  0.17535
## EP_UNINSUR                    -0.02161 -0.03074 -0.01242
## co                            0.06172  0.03232  0.09210
## o3                            0.05348 -0.02564  0.13299
## pm10                          -0.14444 -0.17578 -0.11307
## pm25                          0.29070  0.24581  0.33482
## so2                           0.00316 -0.02811  0.03512
## summer_tmmx                   0.07150  0.01583  0.12678
## winter_tmmx                   0.06279 -0.08721  0.20250
## summer_rmax                   0.04320 -0.02625  0.11262
## winter_rmax                   -0.02414 -0.07435  0.02970
## Data_Value_CSMOKING          2.14325  2.12486  2.16169

```

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"                         "pm10"                "pm25"
## [22] "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, 26.5, 30.5), 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, y = 1.5, label = "Air Pollution",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 28.5, 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", "O3", "PM10", "PM2.5", "SO2",
                             "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

