

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  3.1.8      v dplyr    1.0.10
## v tidyr   1.2.1      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_fr_and_pollute_pc.rds")
```

Function for post-processing the inference

```
pc_idx <- 2:6
pc2flip <- c(-1, 1, -1, -1, -1)

post_flip <- function(beta_inf_subset, pc2flip) {
  names_temp <- colnames(beta_inf_subset)
  beta_inf_subset[pc2flip == -1, ] <- beta_inf_subset[pc2flip == -1, c(1, 3, 2)]
  colnames(beta_inf_subset) <- names_temp
```

```

    return(sweep(beta_inf_subset, 1, pc2flip, FUN = "*"))
}

```

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"))
```

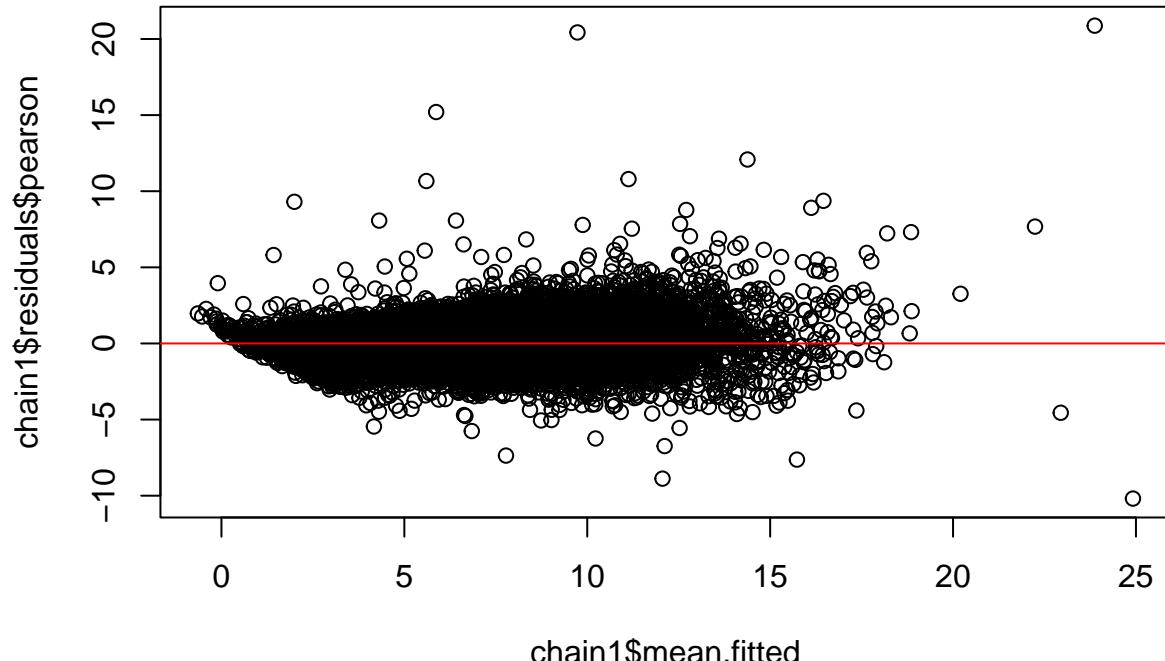
Model Diagnostics

Residual Plots

For each chain

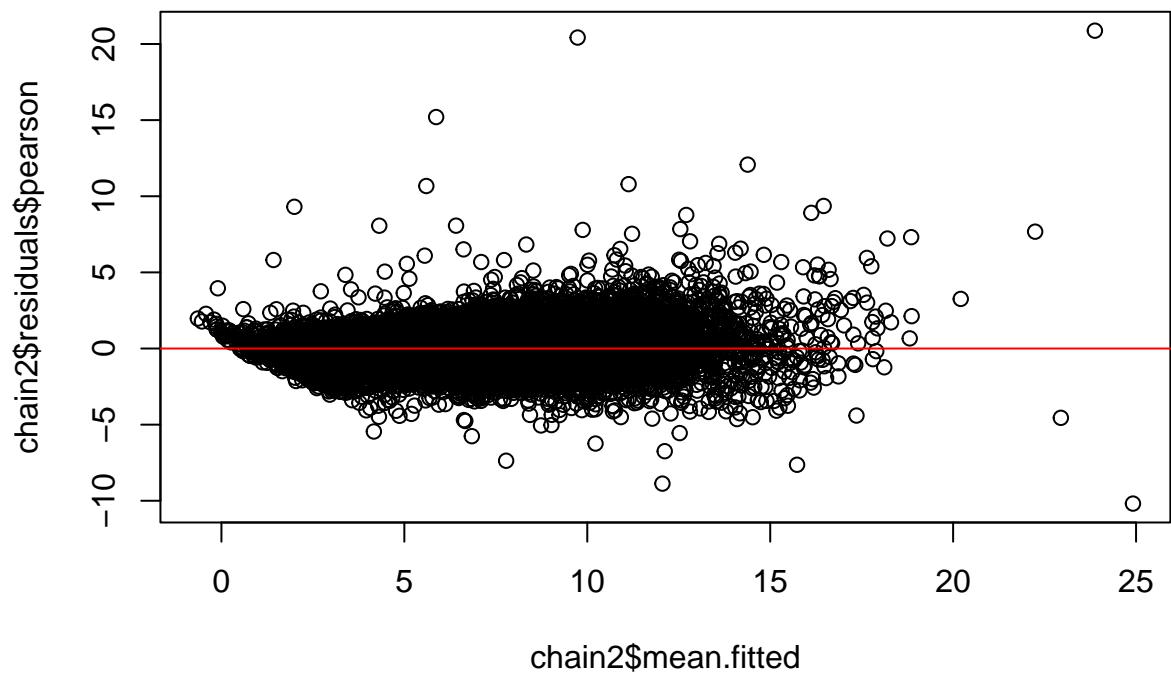
```
plot(chain1$mean.fitted, chain1$residuals$pearson)

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

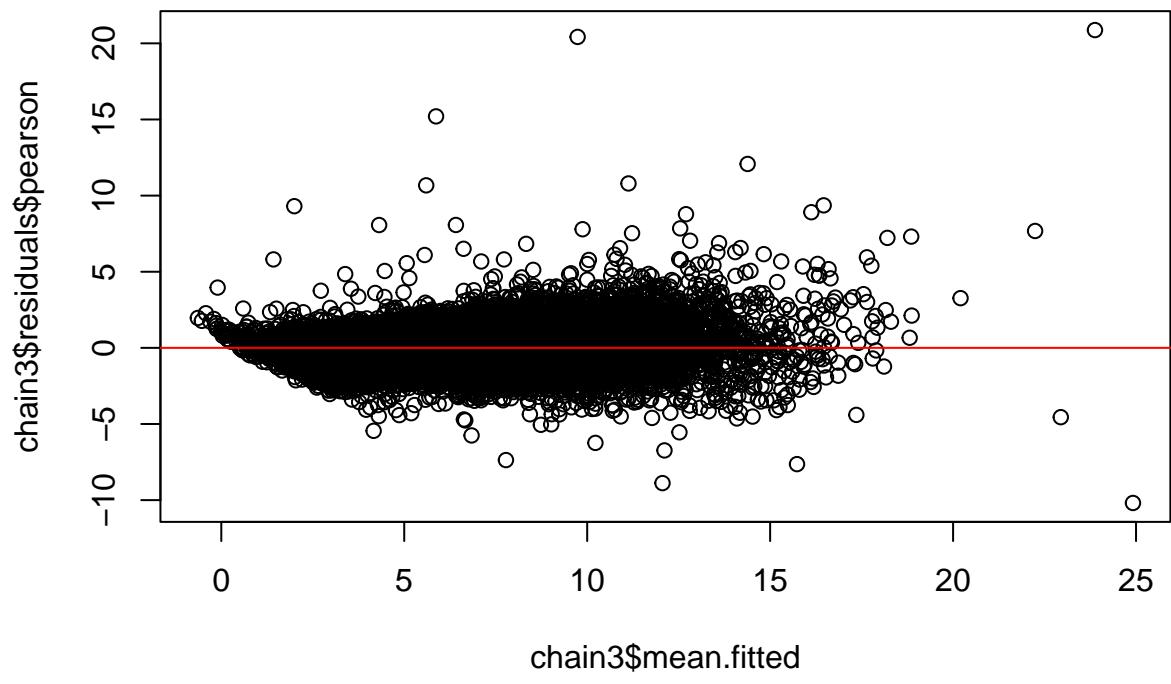


```
plot(chain2$mean.fitted, chain2$residuals$pearson)

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

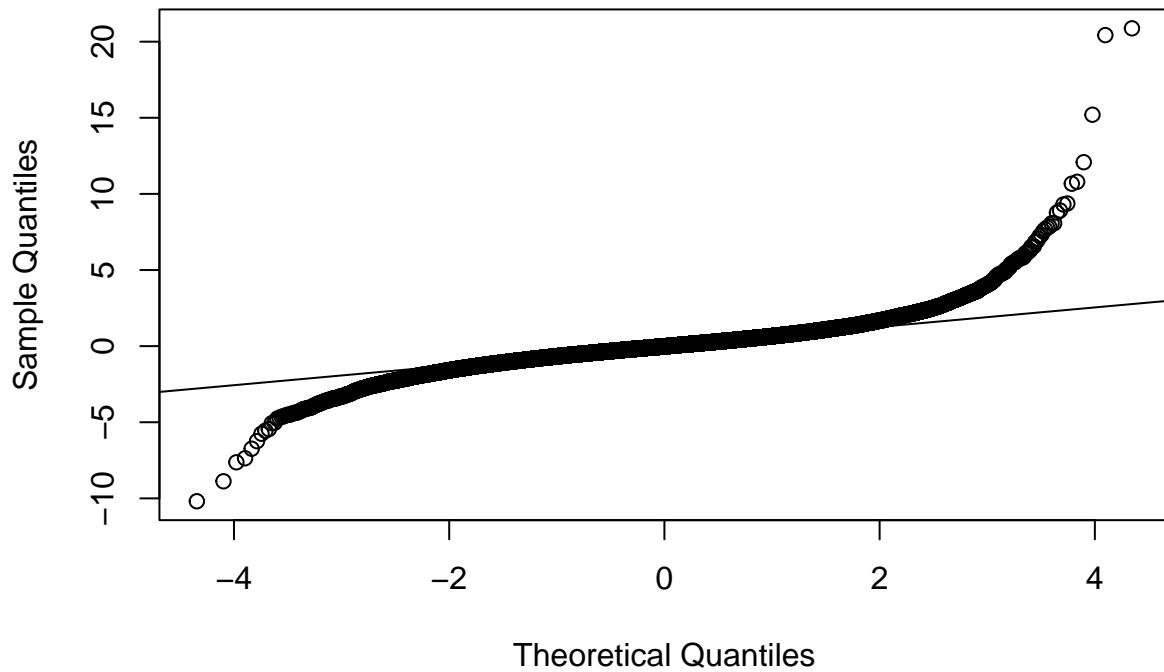


```
plot(chain3$mean.fitted, chain3$residuals$pearson)  
abline(h = 0, col = "red")
```



```
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
## 146297.7390 29590.2832 25027.3838 41921.4098 55016.1576 61121.3882
##      var7      var8      var9      var10     var11     var12
## 40483.1168 70687.5043 25903.2220 40033.9866 33382.9594 44628.3443
##      var13     var14     var15     var16     var17     var18
## 49717.3101 74445.8266 19520.2494 36758.9121 42458.5083 30677.5037
##      var19     var20     var21     var22     var23     var24
## 48409.8141 27735.3490 72615.6565 39053.9669 1579.3121   728.7633
##      var25     var26     var27     var28
## 1079.7187   433.1962   952.5353 16759.8878

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,]     1.01    1.03
## [24,]     1.01    1.04
## [25,]     1.01    1.05
## [26,]     1.01    1.01
## [27,]     1.02    1.05
## [28,]     1.00    1.00
##
## Multivariate psrf
##
## 1.02

```

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          1
## [2,]          1          1
## [3,]          1          1
## [4,]          1          1
## [5,]          1          1
## [6,]          1          1
## [7,]          1          1
## [8,]          1          1
## [9,]          1          1
## [10,]         1          1
##
## Multivariate psrf
##
## 1

```

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)

# flipping the inference results according to the flipped PCs
beta_inference[pc_idx, ] <- post_flip(beta_inference[pc_idx, ], pc2flip)

beta_inference

##           50%      2.5%     97.5%

```

```

## Intercept          6.66173  6.65747  6.66601
## flood_risk_pc1   0.00695 -0.00173  0.01554
## flood_risk_pc2   0.00719 -0.00262  0.01696
## flood_risk_pc3  -0.00603 -0.01344  0.00139
## flood_risk_pc4  -0.00532 -0.01196  0.00140
## flood_risk_pc5  -0.00039 -0.00689  0.00609
## EP_POV            0.31894  0.30673  0.33118
## EP_UNEMP           0.03061  0.02268  0.03856
## EP_PCI             -0.03785 -0.04970 -0.02593
## EP_NOHSDP          0.19923  0.18368  0.21473
## EP_AGE65           1.38157  1.37144  1.39164
## EP_AGE17           0.27886  0.26818  0.28951
## EP_DISABL          0.27340  0.26339  0.28349
## EP_SNGPNT          -0.06258 -0.07183 -0.05336
## EP_MINRTY          -0.04258 -0.05807 -0.02719
## EP_LIMENG          -0.06630 -0.08028 -0.05247
## EP_MUNIT            -0.06388 -0.07285 -0.05487
## EP_MOBILE           0.07389  0.06561  0.08208
## EP_CROWD            -0.04691 -0.05739 -0.03648
## EP_NOVEH            0.12381  0.11032  0.13732
## EP_GROUPQ           -0.09687 -0.10392 -0.08986
## EP_UNINSUR          0.10631  0.09584  0.11672
## pollute_conc_pc1   0.14217  0.11055  0.17307
## pollute_conc_pc2   -0.24975 -0.29232 -0.20764
## pollute_conc_pc3   0.01060 -0.02865  0.04926
## tmmx                0.07007  0.01106  0.13347
## rmax                0.03981 -0.00309  0.08105
## Data_Value_CSMOKING 0.78071  0.76061  0.80067

```

List of significant beta coefficients:

```

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

## [1] "Intercept"          "EP_POV"                  "EP_UNEMP"
## [4] "EP_PCI"               "EP_NOHSDP"                "EP_AGE65"
## [7] "EP_AGE17"              "EP_DISABL"                "EP_SNGPNT"
## [10] "EP_MINRTY"             "EP_LIMENG"                "EP_MUNIT"
## [13] "EP_MOBILE"              "EP_CROWD"                 "EP_NOVEH"
## [16] "EP_GROUPQ"              "EP_UNINSUR"               "pollute_conc_pc1"
## [19] "pollute_conc_pc2"       "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%`)

```

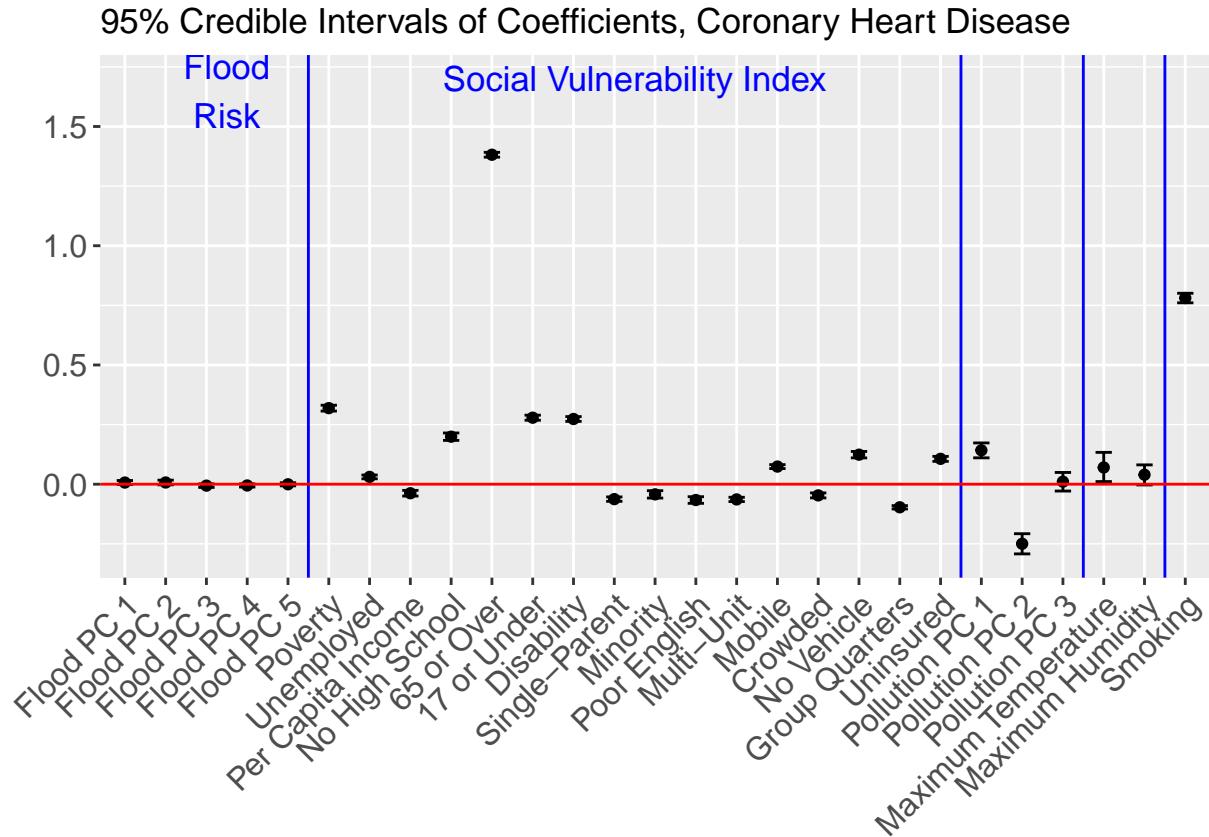
```

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, 25)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
  geom_vline(xintercept = c(5.5, 21.5, 24.5, 26.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3.5, y = 1.65, label = "Flood\nRisk",
           col = "blue", size = 4.5) +
  annotate(geom = "text", x = 13.5, y = 1.7, label = "Social Vulnerability Index",
           col = "blue", size = 4.5) +
  scale_x_discrete(labels = c("Flood PC 1", "Flood PC 2", "Flood PC 3", "Flood PC 4", "Flood 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",
                             "Pollution PC 1", "Pollution PC 2", "Pollution PC 3",
                             "Maximum Temperature", "Maximum Humidity",
                             "Smoking")) + ggttitle("95% Credible Intervals of Coefficients, Coronary Heart Disease")

```

p



```

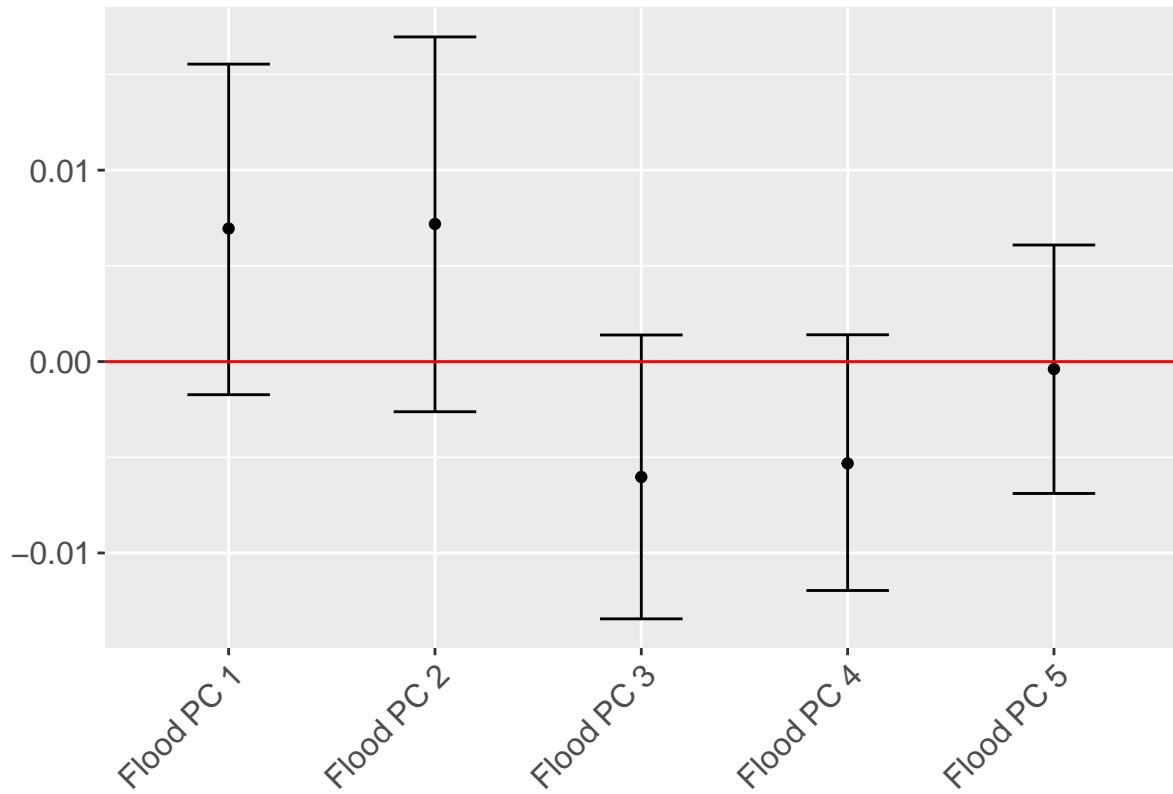
ggsave(here("figures/final_figures/CHD_cred_intervals.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")

p <- ggplot(beta_inference_df[2:6, ], 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, 25)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
  geom_hline(yintercept = 0, col = "red") +
  scale_x_discrete(labels = c("Flood PC 1", "Flood PC 2", "Flood PC 3", "Flood PC 4", "Flood PC 5")) +
  scale_y_continuous(trans = "log")

```

p

95% Credible Intervals for Flood Risk PCs, Coronary Heart Disease



```

ggsave(here("figures/final_figures/CHD_cred_intervals_fr_only.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")

```

CAR model results, High Blood Pressure

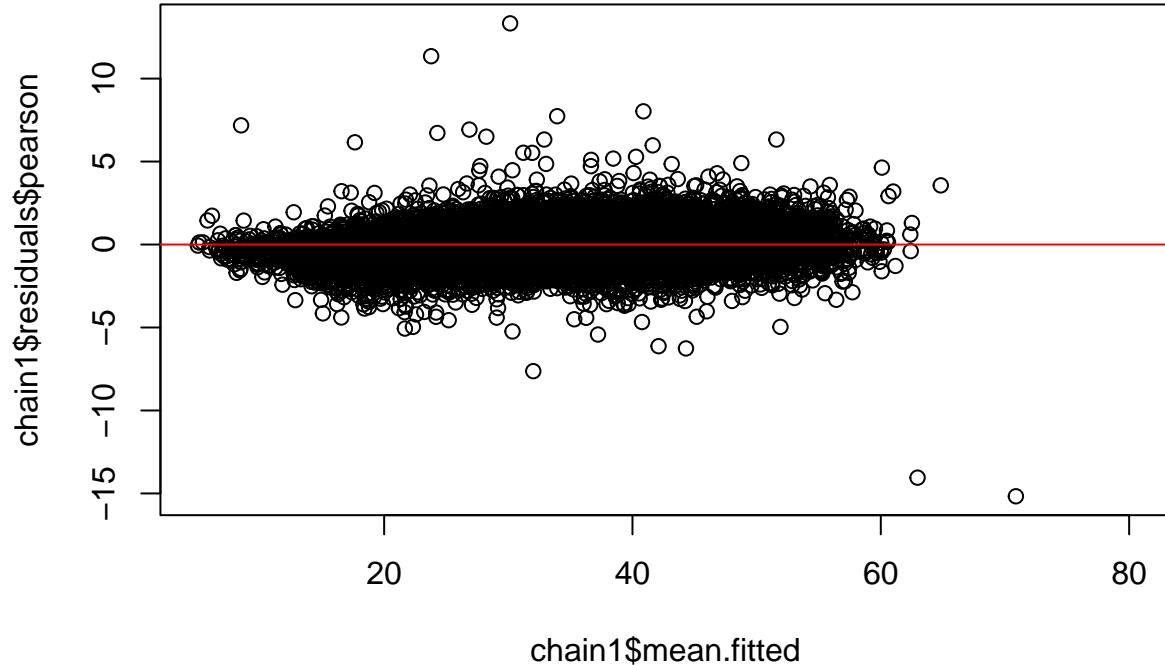
Inference is based on 3 markov chains, each of which has been run for 110000 samples, the first 10000 of which has been removed for burn-in. The remaining 100000 samples are thinned by 2, resulting in 150000 samples for inference across the 3 Markov chains.

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

Model Diagnostics

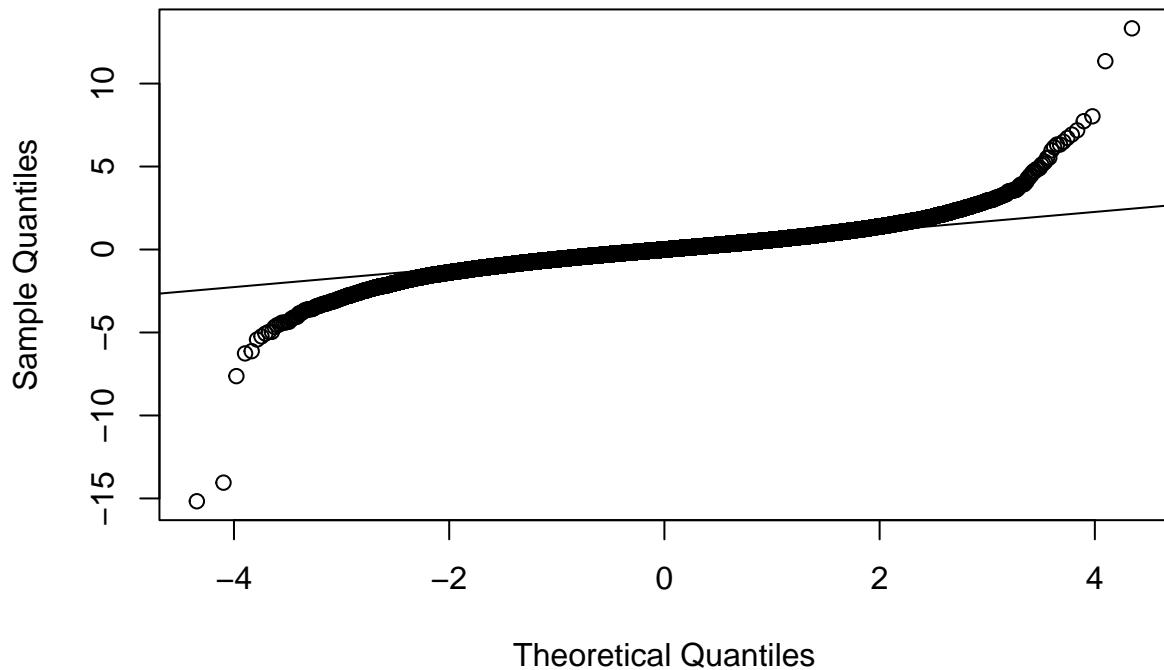
Residual plot

```
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
## 131682.4555 23625.5651 21876.3067 32830.1512 42963.0675 46590.9970
##      var7      var8      var9      var10     var11     var12
## 32656.7694 54687.2647 21570.3482 32879.3938 27404.9677 35932.6781
##      var13     var14     var15     var16     var17     var18
## 42070.6333 58357.9910 15661.2055 29866.3868 33984.4081 27186.1247
##      var19     var20     var21     var22     var23     var24
## 40870.7824 24296.6472 53990.4908 33562.8400 1428.6211   609.7696
##      var25     var26     var27     var28
## 918.9889   363.4227    758.8239 14335.4311

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,]   1.01    1.04
## [24,]   1.01    1.04
## [25,]   1.02    1.06
## [26,]   1.01    1.03
## [27,]   1.02    1.05
## [28,]   1.00    1.00
##
## Multivariate psrf
##
## 1.02

```

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      1
## [2,]      1      1
## [3,]      1      1
## [4,]      1      1
## [5,]      1      1
## [6,]      1      1
## [7,]      1      1
## [8,]      1      1
## [9,]      1      1
## [10,]     1      1
##
## Multivariate psrf
##
## 1
```

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)

# flipping the inference results according to the flipped PCs
beta_inference[pc_idx, ] <- post_flip(beta_inference[pc_idx, ], pc2flip)

beta_inference

##                  50%    2.5%   97.5%
## Intercept      32.32705 32.31572 32.33841
## flood_risk_pc1  0.00893 -0.01940  0.03692
## flood_risk_pc2  0.07384  0.04220  0.10534
## flood_risk_pc3 -0.05762 -0.08158 -0.03357
## flood_risk_pc4 -0.04355 -0.06483 -0.02186
## flood_risk_pc5 -0.00813 -0.02892  0.01265
## EP_POV         0.04623  0.00703  0.08544
```

```

## EP_UNEMP          0.12998  0.10506  0.15508
## EP_PCI           0.24762  0.20920  0.28625
## EP_NOHSDP        0.01054 -0.03916  0.05999
## EP_AGE65         4.06820  4.03573  4.10052
## EP_AGE17         0.61042  0.57667  0.64415
## EP_DISABL        0.74931  0.71774  0.78106
## EP_SNGPNT        -0.04288 -0.07206 -0.01382
## EP_MINRTY        2.63144  2.58042  2.68216
## EP_LIMENG        -0.89603 -0.94098 -0.85154
## EP_MUNIT         -0.62894 -0.65767 -0.60013
## EP_MOBILE         0.10389  0.07745  0.13002
## EP_CROWD          -0.12619 -0.15951 -0.09318
## EP_NOVEH          0.60932  0.56602  0.65268
## EP_GROUPQ         -0.59703 -0.61933 -0.57491
## EP_UNINSUR        0.23759  0.20438  0.27071
## pollute_conc_pc1 -0.30945 -0.41670 -0.20431
## pollute_conc_pc2 -0.97454 -1.12246 -0.82868
## pollute_conc_pc3  0.31276  0.17734  0.44649
## tmmx              0.09056 -0.11418  0.31381
## rmax              0.06842 -0.08521  0.21777
## Data_Value_CSMOKING 2.51471  2.44966  2.57948

```

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_POV"               "EP_UNEMP"
## [7] "EP_PCI"                    "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"               "pollute_conc_pc1"    "pollute_conc_pc2"
## [22] "pollute_conc_pc3"          "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),

```

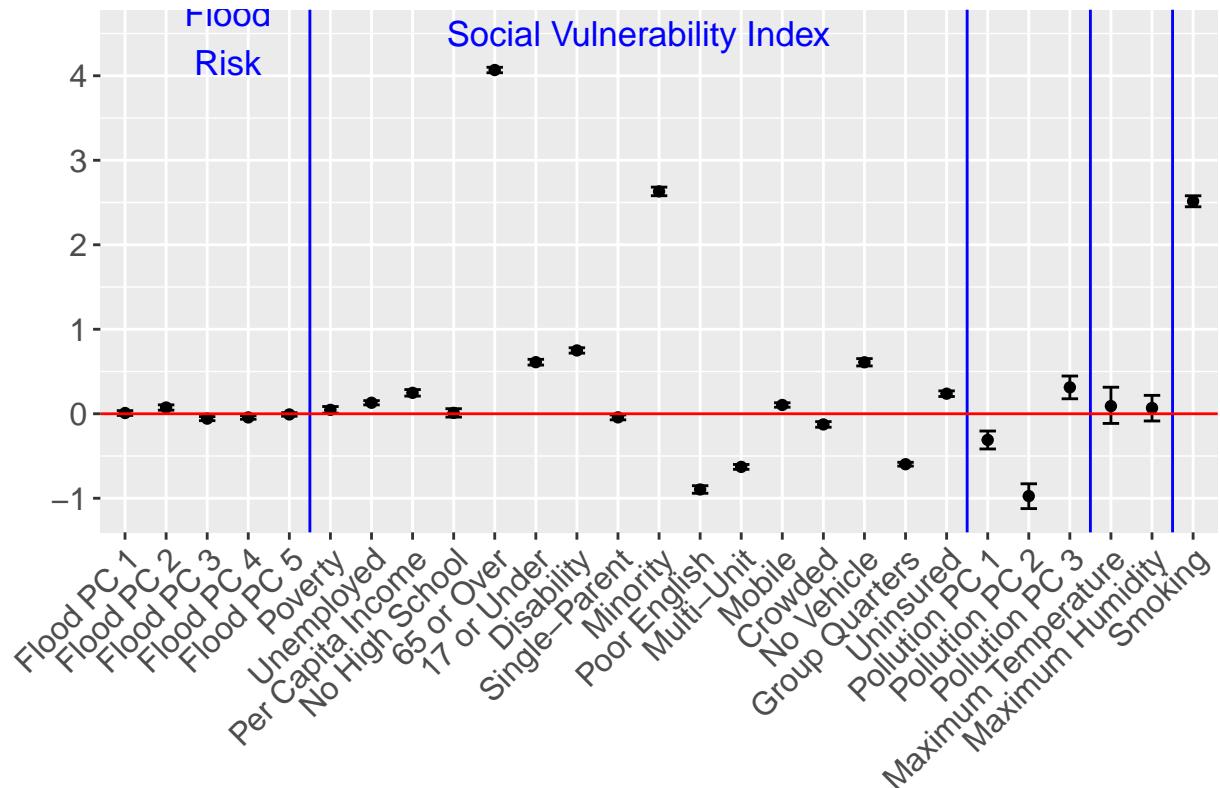
```

    plot.margin = margin(5.5, 5.5, 5.5, 25)) +
geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
geom_vline(xintercept = c(5.5, 21.5, 24.5, 26.5), col = "blue") +
geom_hline(yintercept = 0, col = "red") +
annotate(geom = "text", x = 3.5, y = 4.45, label = "Flood\nRisk",
        col = "blue", size = 4.5) +
annotate(geom = "text", x = 13.5, y = 4.5, label = "Social Vulnerability Index",
        col = "blue", size = 4.5) +
scale_x_discrete(labels = c("Flood PC 1", "Flood PC 2", "Flood PC 3", "Flood PC 4", "Flood 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",
                           "Pollution PC 1", "Pollution PC 2", "Pollution PC 3",
                           "Maximum Temperature", "Maximum Humidity",
                           "Smoking")) + ggtitle("95% Credible Intervals of Coefficients, High Blood Pressure")

```

p

95% Credible Intervals of Coefficients, High Blood Pressure



```

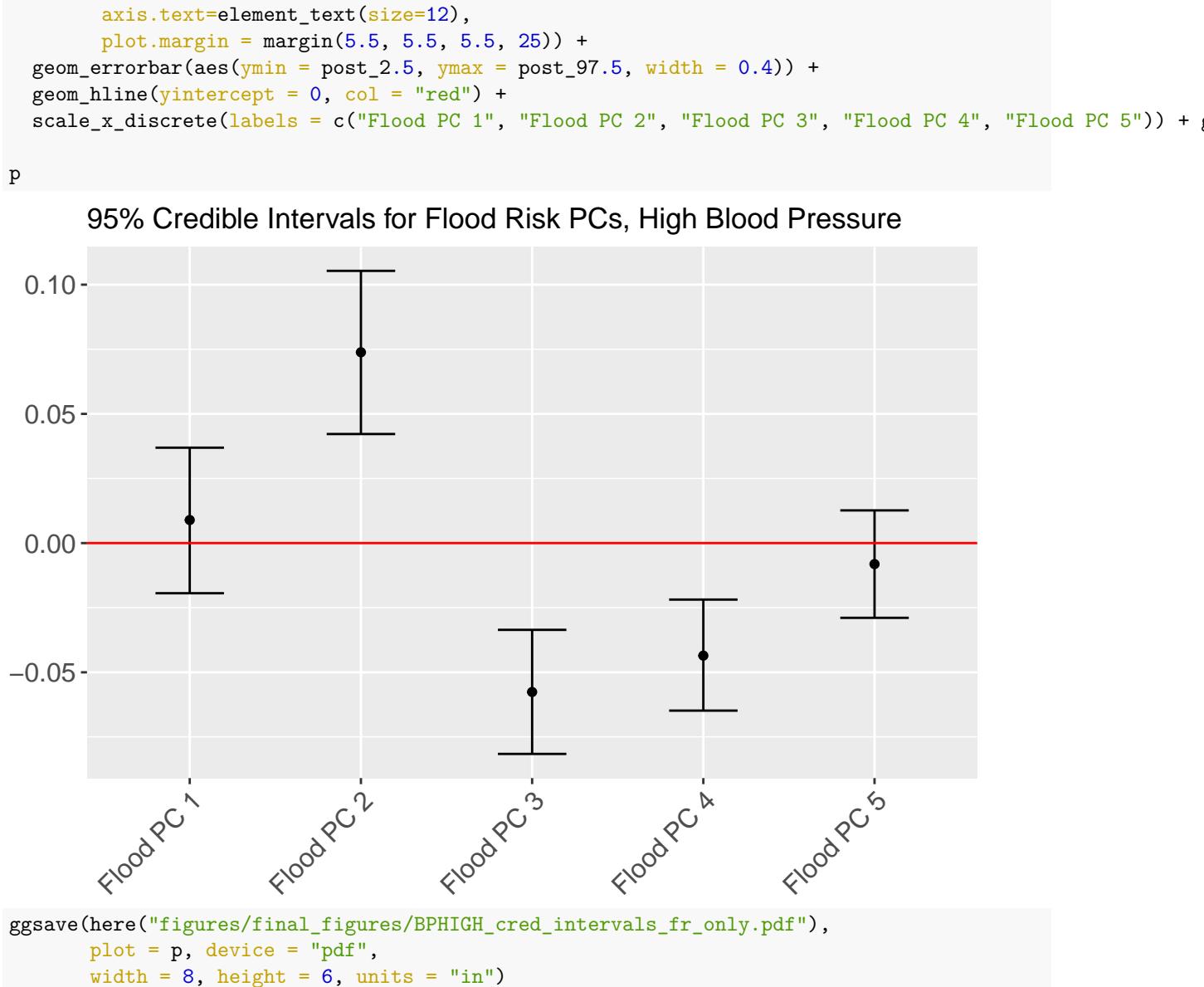
ggsave(here("figures/final_figures/BPHIGH_cred_intervals.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")

```

```

p <- ggplot(beta_inference_df[2:6, ], 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

```



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.

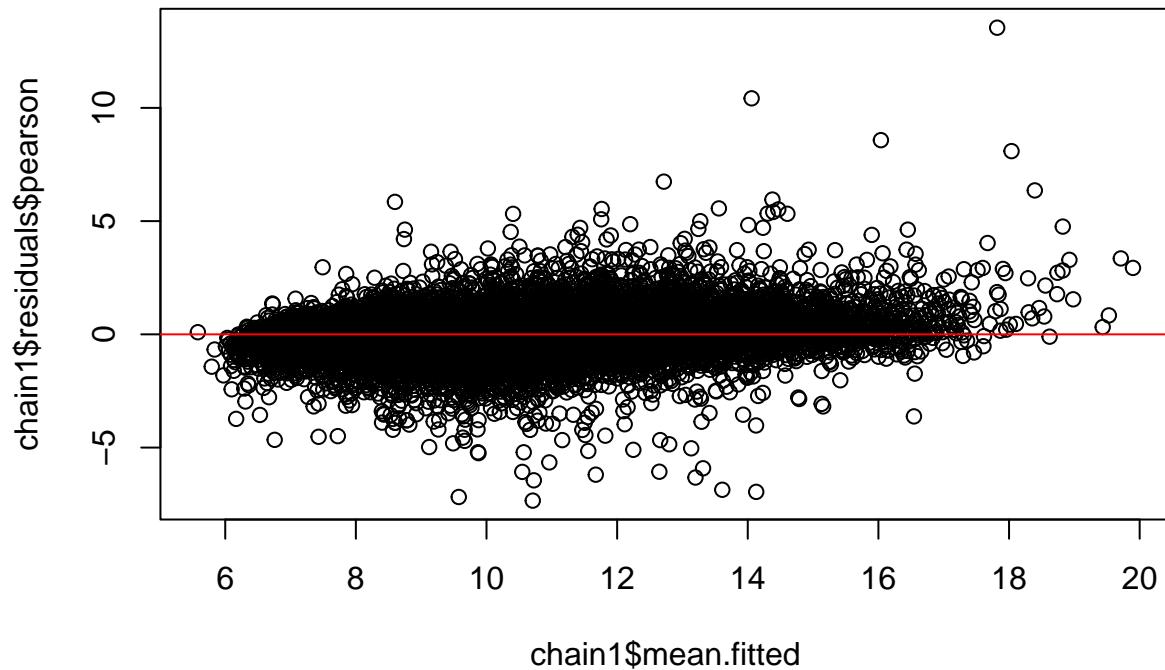
```
load(here("modeling_files/all_census_tract_CASTHMA.RData"))
```

Model Diagnostics

Residual plot

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

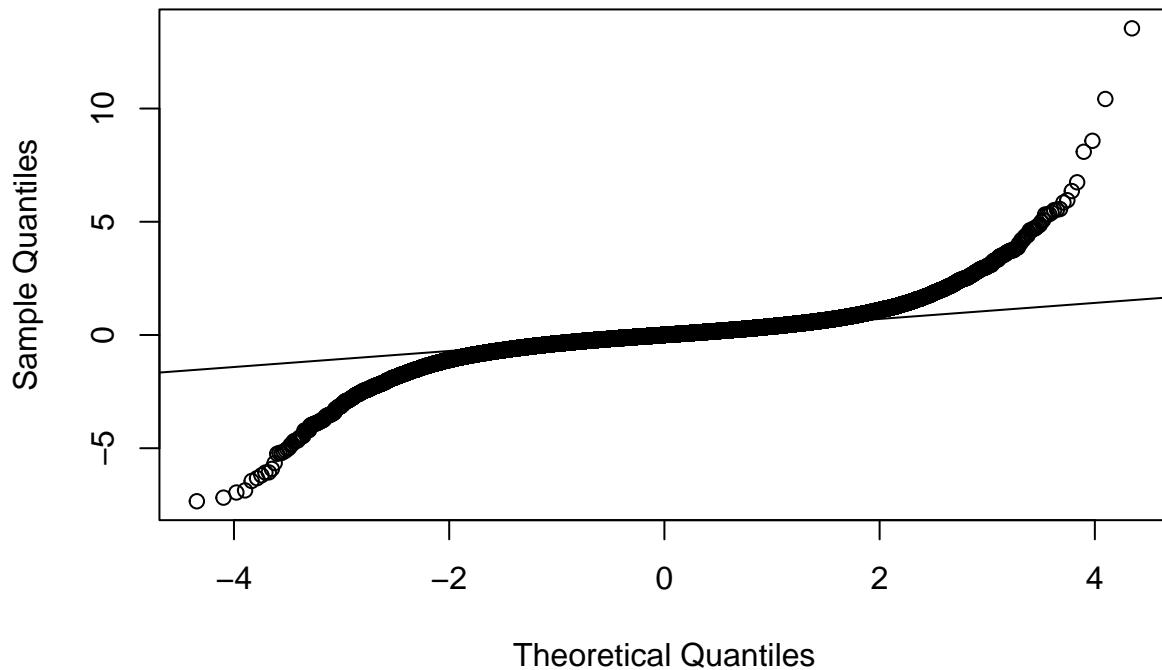
abline(h = 0, col = "red")
```



```
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
## 96098.1946 15753.4790 16051.3258 21971.1037 27340.0888 29846.5309 21947.6590
##      var8      var9      var10     var11     var12     var13     var14
## 35403.4803 15380.6533 21534.9102 21524.2082 25317.3947 30321.2558 37448.8137
##      var15     var16     var17     var18     var19     var20     var21
## 10950.5932 19754.2701 23819.3349 20594.1987 27363.5893 17224.3310 34700.2902
##      var22     var23     var24     var25     var26     var27     var28
## 23106.0295 1090.6248   434.0849   642.5501   243.0511   508.1845 10332.3017

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,] 1.02 1.06
## [24,] 1.01 1.05
## [25,] 1.02 1.08
## [26,] 1.02 1.05
## [27,] 1.02 1.06
## [28,] 1.00 1.00
##
## Multivariate psrf
##
## 1.03

```

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 1.00
## [2,] 1 1.00
## [3,] 1 1.01
## [4,] 1 1.00
## [5,] 1 1.00
## [6,] 1 1.00
## [7,] 1 1.00
## [8,] 1 1.00
## [9,] 1 1.00
## [10,] 1 1.00

## Multivariate psrf
## 1
```

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)

# flipping the inference results according to the flipped PCs
beta_inference[pc_idx, ] <- post_flip(beta_inference[pc_idx, ], pc2flip)

beta_inference

## 50% 2.5% 97.5%
## Intercept 9.89848 9.89695 9.90002
## flood_risk_pc1 -0.00492 -0.00980 -0.00008
## flood_risk_pc2 0.00325 -0.00217 0.00861
## flood_risk_pc3 0.00101 -0.00307 0.00511
## flood_risk_pc4 0.00716 0.00353 0.01086
## flood_risk_pc5 -0.00062 -0.00416 0.00292
## EP_POV 0.28522 0.27858 0.29189
```

```

## EP_UNEMP          0.07078  0.06659  0.07501
## EP_PCI           -0.06512 -0.07170 -0.05851
## EP_NOHSDP         0.07605  0.06760  0.08449
## EP_AGE65          0.09865  0.09319  0.10409
## EP_AGE17          0.00991  0.00423  0.01560
## EP_DISABL         -0.05944 -0.06476 -0.05409
## EP_SNGPNT          0.04026  0.03533  0.04517
## EP_MINRTY         0.32194  0.31315  0.33072
## EP_LIMENG         -0.25647 -0.26413 -0.24887
## EP_MUNIT          -0.02704 -0.03192 -0.02216
## EP_MOBILE          -0.01866 -0.02313 -0.01422
## EP_CROWD           -0.02037 -0.02598 -0.01476
## EP_NOVEH            0.14927  0.14191  0.15663
## EP_GROUPQ          -0.04690 -0.05067 -0.04318
## EP_UNINSUR         -0.02733 -0.03293 -0.02171
## pollute_conc_pc1   -0.06169 -0.08052 -0.04307
## pollute_conc_pc2   -0.18172 -0.20834 -0.15566
## pollute_conc_pc3   0.04999  0.02613  0.07384
## tmmx                0.03688  0.00040  0.07778
## rmax                -0.04627 -0.07446 -0.01857
## Data_Value_CSMOKING 0.85196  0.84086  0.86299

```

List of significant beta coefficients:

```

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

## [1] "Intercept"                  "flood_risk_pc1"      "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"                 "pollute_conc_pc1"    "pollute_conc_pc2"
## [22] "pollute_conc_pc3"          "tmmx"                  "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

```

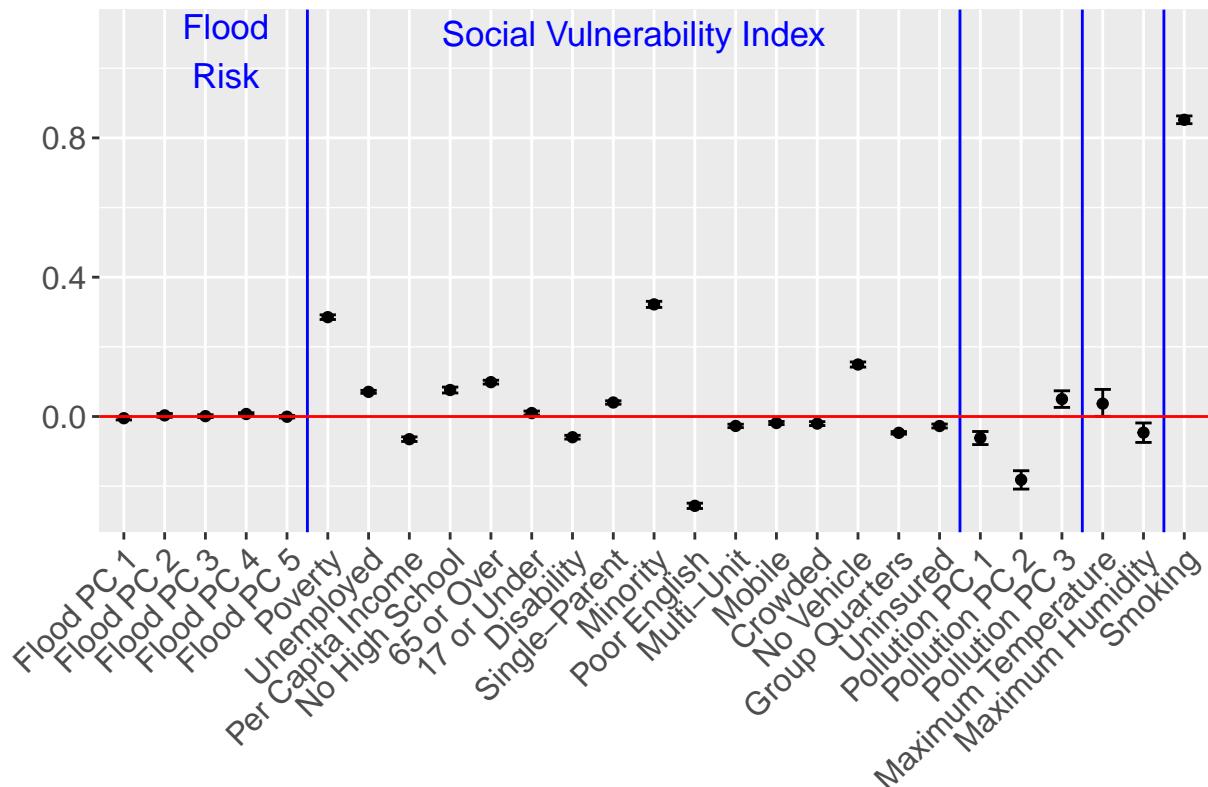
```

axis.text=element_text(size=12),
plot.margin = margin(5.5, 5.5, 5.5, 25)) +
geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
geom_vline(xintercept = c(5.5, 21.5, 24.5, 26.5), col = "blue") +
geom_hline(yintercept = 0, col = "red") +
annotate(geom = "text", x = 3.5, y = 1.05, label = "Flood\nRisk",
        col = "blue", size = 4.5) +
annotate(geom = "text", x = 13.5, y = 1.1, label = "Social Vulnerability Index",
        col = "blue", size = 4.5) +
scale_x_discrete(labels = c("Flood PC 1", "Flood PC 2", "Flood PC 3", "Flood PC 4", "Flood 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",
                            "Pollution PC 1", "Pollution PC 2", "Pollution PC 3",
                            "Maximum Temperature", "Maximum Humidity",
                            "Smoking")) + ggtitle("95% Credible Intervals of Coefficients, Asthma")

```

p

95% Credible Intervals of Coefficients, Asthma



```

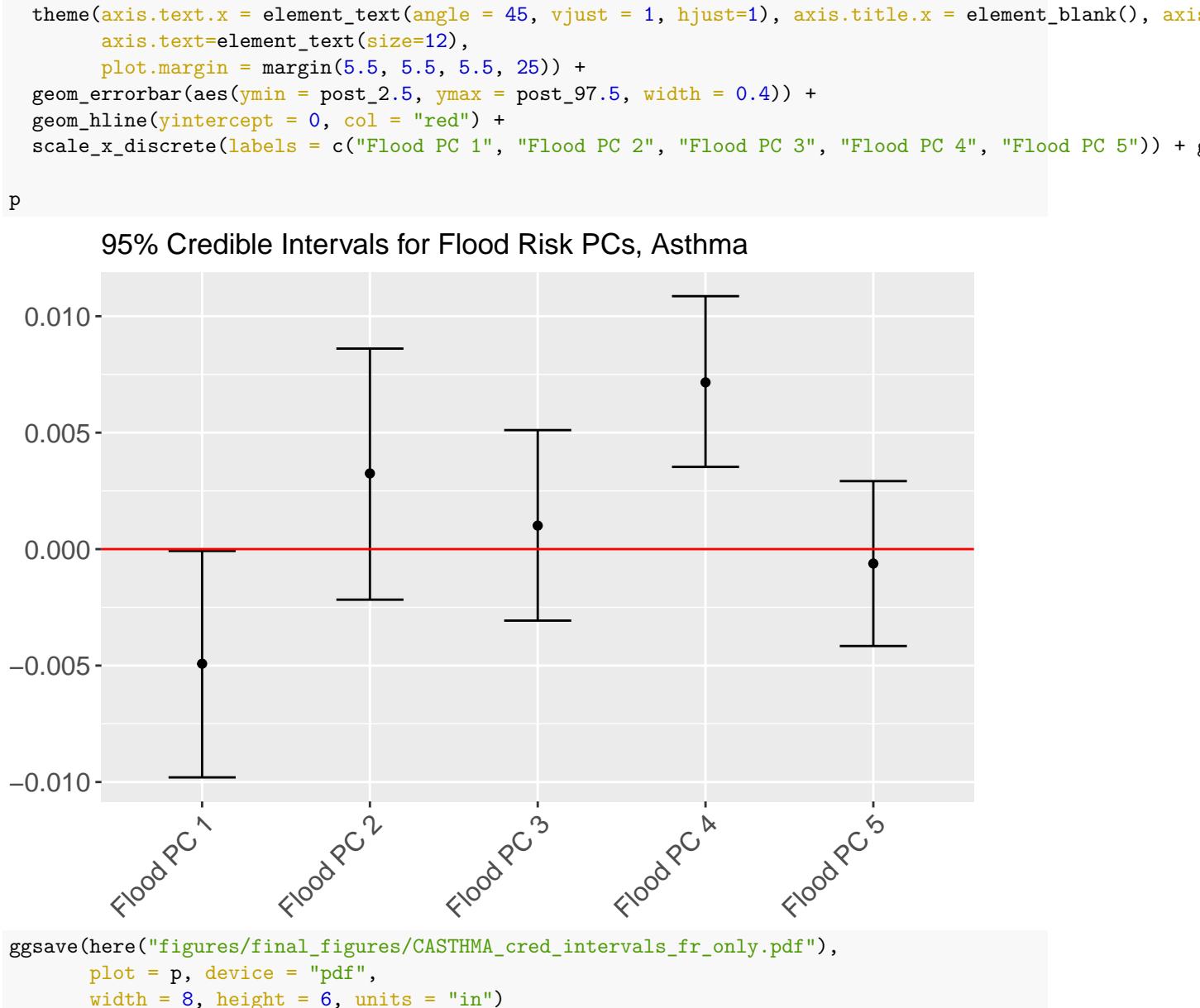
ggsave(here("figures/final_figures/CASTHMA_cred_intervals.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")

```

```

p <- ggplot(beta_inference_df[2:6, ], aes(x = var_name, y = post_median)) +
geom_point() +

```



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.

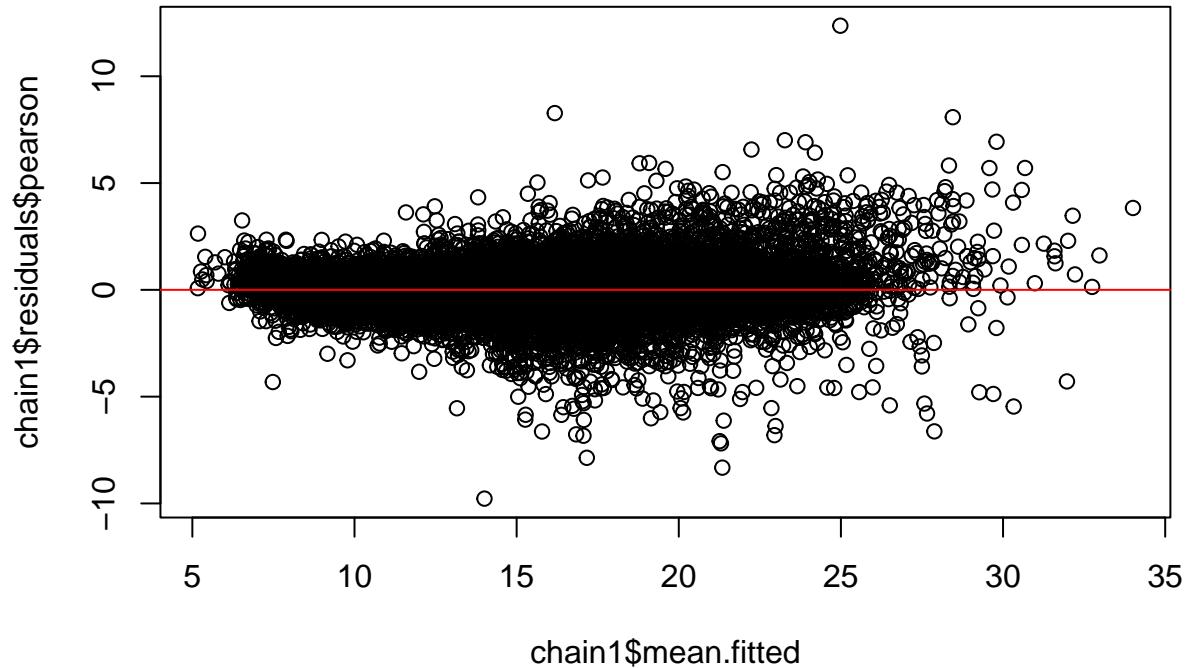
```
load(here("modeling_files/all_census_tract_MHLTH.RData"))
```

Model Diagnostics

Residual plot

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

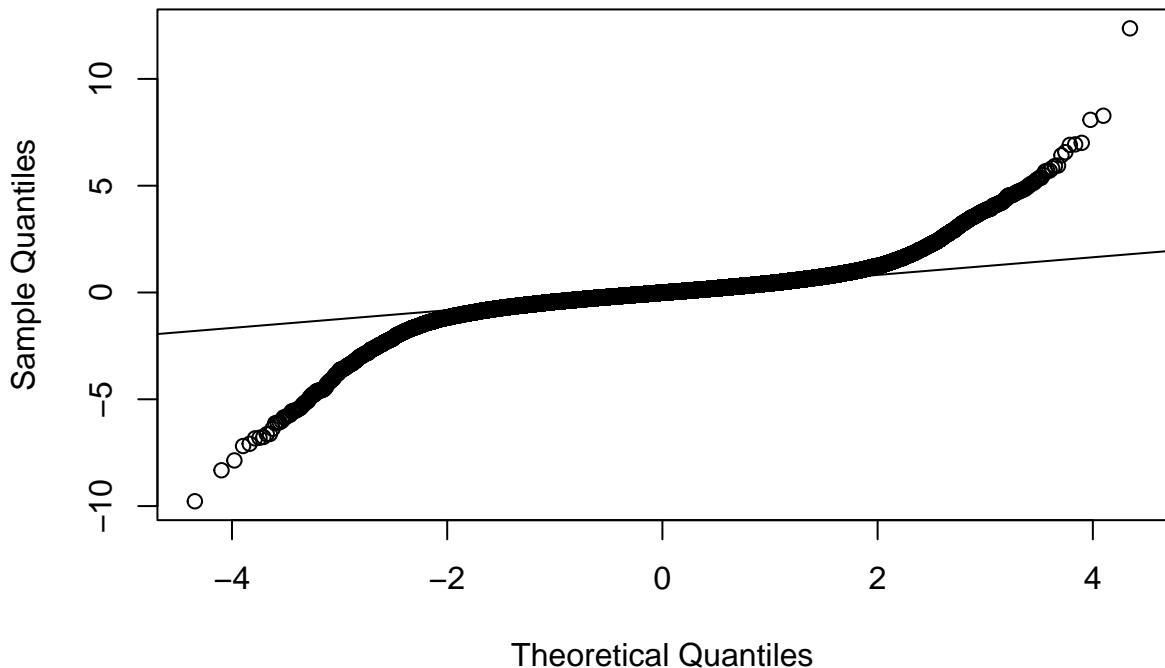
abline(h = 0, col = "red")
```



```
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
## 112229.3353 20403.7410 20234.1838 28217.9258 36122.9457 39377.0637
##      var7      var8      var9      var10     var11     var12
## 23164.4821 46771.1390 18837.6378 28306.2613 26487.8198 30788.1262
##      var13     var14     var15     var16     var17     var18
## 36544.9012 49359.3384 13812.7604 25770.0724 29762.3955 25121.5826
##      var19     var20     var21     var22     var23     var24
## 35130.8909 21517.2856 46144.6269 29394.4770 1297.6800   539.7202
##      var25     var26     var27     var28
## 818.6920   318.7206   652.4327 12394.4874

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,]   1.01    1.05
## [24,]   1.01    1.04
## [25,]   1.02    1.07
## [26,]   1.01    1.04
## [27,]   1.02    1.05
## [28,]   1.00    1.00
##
## Multivariate psrf
##
## 1.02

```

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 1.00
## [2,] 1 1.00
## [3,] 1 1.01
## [4,] 1 1.00
## [5,] 1 1.00
## [6,] 1 1.00
## [7,] 1 1.00
## [8,] 1 1.00
## [9,] 1 1.00
## [10,] 1 1.00

## Multivariate psrf
## 1
```

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)

# flipping the inference results according to the flipped PCs
beta_inference[pc_idx, ] <- post_flip(beta_inference[pc_idx, ], pc2flip)

beta_inference

## 50% 2.5% 97.5%
## Intercept 14.26569 14.26281 14.26859
## flood_risk_pc1 -0.00487 -0.01282 0.00299
## flood_risk_pc2 0.00150 -0.00733 0.01031
## flood_risk_pc3 0.00446 -0.00223 0.01119
## flood_risk_pc4 0.00954 0.00361 0.01560
## flood_risk_pc5 -0.00013 -0.00592 0.00568
## EP_POV 0.82099 0.80988 0.83210
```

```

## EP_UNEMP          0.07124  0.06433  0.07821
## EP_PCI           -0.33084 -0.34158 -0.32000
## EP_NOHSDP         0.25468  0.24082  0.26850
## EP_AGE65          -0.33504 -0.34405 -0.32613
## EP_AGE17          -0.10572 -0.11511 -0.09634
## EP_DISABL         -0.17357 -0.18231 -0.16475
## EP_SNGPNT          0.07765  0.06954  0.08571
## EP_MINRTY         -0.18062 -0.19490 -0.16640
## EP_LIMENG          -0.03261 -0.04512 -0.02017
## EP_MUNIT           0.07576  0.06776  0.08377
## EP_MOBILE          -0.01721 -0.02457 -0.00993
## EP_CROWD            0.05877  0.04952  0.06798
## EP_NOVEH            0.07548  0.06340  0.08756
## EP_GROUPQ           0.16802  0.16182  0.17415
## EP_UNINSUR          -0.02046 -0.02971 -0.01124
## pollute_conc_pc1    0.15796  0.12761  0.18781
## pollute_conc_pc2   -0.07776 -0.12007 -0.03611
## pollute_conc_pc3   -0.05869 -0.09715 -0.02045
## tmmx                 0.15213  0.09367  0.21650
## rmax                  0.01260 -0.03160  0.05606
## Data_Value_CSMOKING 2.13894  2.12064  2.15714

```

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] "pollute_conc_pc1"        "pollute_conc_pc2"        "pollute_conc_pc3"
## [22] "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),

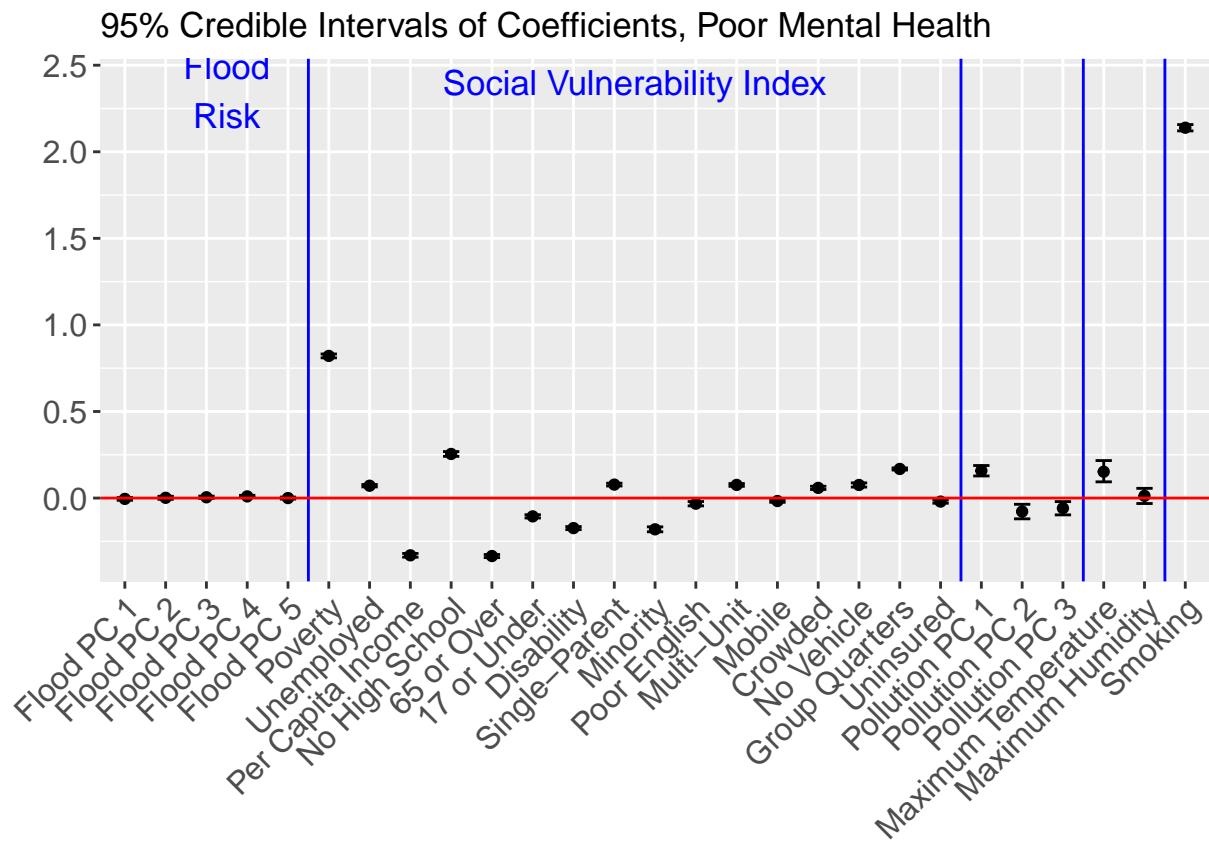
```

```

    plot.margin = margin(5.5, 5.5, 5.5, 25)) +
geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
geom_vline(xintercept = c(5.5, 21.5, 24.5, 26.5), col = "blue") +
geom_hline(yintercept = 0, col = "red") +
annotate(geom = "text", x = 3.5, y = 2.35, label = "Flood\nRisk",
        col = "blue", size = 4.5) +
annotate(geom = "text", x = 13.5, y = 2.4, label = "Social Vulnerability Index",
        col = "blue", size = 4.5) +
scale_x_discrete(labels = c("Flood PC 1", "Flood PC 2", "Flood PC 3", "Flood PC 4", "Flood 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",
                           "Pollution PC 1", "Pollution PC 2", "Pollution PC 3",
                           "Maximum Temperature", "Maximum Humidity",
                           "Smoking")) + ggtitle("95% Credible Intervals of Coefficients, Poor Mental Health")

```

p



```

ggsave(here("figures/final_figures/MHLTH_cred_intervals.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")

```

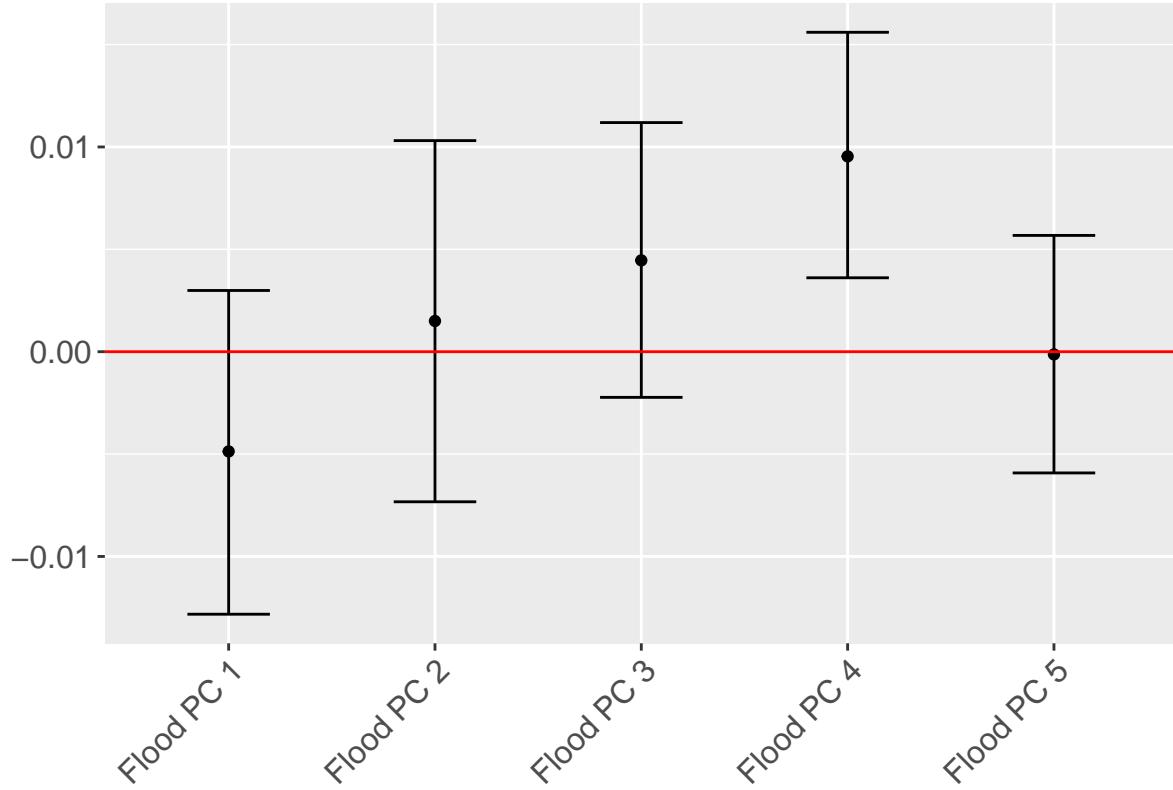
```

p <- ggplot(beta_inference_df[2:6, ], 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

```

```
    axis.text=element_text(size=12),
    plot.margin = margin(5.5, 5.5, 5.5, 25)) +
geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4)) +
geom_hline(yintercept = 0, col = "red") +
scale_x_discrete(labels = c("Flood PC 1", "Flood PC 2", "Flood PC 3", "Flood PC 4", "Flood PC 5")) +
p
```

95% Credible Intervals for Flood Risk PCs, Poor Mental Health



```
ggsave(here("figures/final_figures/MHLTH_cred_intervals_fr_only.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")
```