# Sensitivity Analysis: High Carbon Emissions Scenario

Alvin Sheng

## Contents

```r
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
```

```r
library(coda)
library(CARBayes)
```

```
## Loading required package: MASS

## Loading required package: Rcpp

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
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      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()
```

```r
fhs_model_df <- readRDS("intermediary_data/sensitivity_analysis/fhs_model_df_high_ver.rds")

var_names <- c("Intercept", "flood_risk_pc1", "flood_risk_pc2",
               "flood_risk_pc3", "flood_risk_pc4", "flood_risk_pc5",
               "EP_UNINSUR", "pollute_conc_pc1", "pollute_conc_pc2",
               "pollute_conc_pc3", "tmmx", "rmax", "Data_Value_CSMOKING")
```

```r
names_high_ver_strat <- c(paste("strat0", var_names, sep = ":"),
                          paste("strat1", var_names, sep = ":"))
```

## Helper Functions

Function for post-processing the inference

```r
pc2flip <- c(-1, 1, -1, -1, -1,
             -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 = "*"))
}
```

```
# extract the flood risk PC coefficients
# pc_idx is the vector of indices of the flood risk PC coefficients, after splitting data frame by stra
beta_data_frames_extract <- function(beta_inference_df, pc_idx) {

  beta_inference_df_strat0 <- beta_inference_df[1:(nrow(beta_inference_df)/2),]

  beta_inference_df_strat1 <- beta_inference_df[(nrow(beta_inference_df)/2 + 1):nrow(beta_inference_df)

  beta_pcs_strat0 <- beta_inference_df_strat0[pc_idx, ]

  beta_pcs_strat1 <- beta_inference_df_strat1[pc_idx, ]



  beta_pcs_strat0 <- mutate(beta_pcs_strat0, var_idx = factor(1:nrow(beta_pcs_strat0)))
  beta_pcs_strat1 <- mutate(beta_pcs_strat1, var_idx = factor(1:nrow(beta_pcs_strat1)))

  return(list(beta_pcs_strat0 = beta_pcs_strat0, beta_pcs_strat1 = beta_pcs_strat1))

}
```

# CAR model results, Coronary Heart Disease Stratified on RPL_THEMES

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/sensitivity_analysis/high_ver/all_census_tract_CHD.RData"))
```

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

colnames(beta_samples_matrix) <- names_high_ver_strat
```

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

pc_idx <- c(2:6,
            nrow(beta_inference)/2 + 2:6)

# 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%
## strat0:Intercept            6.23210  6.21246  6.25171
## strat0:flood_risk_pc1       0.08073  0.06057  0.10087
## strat0:flood_risk_pc2       0.02585  0.00309  0.04886
## strat0:flood_risk_pc3       0.02670  0.00956  0.04389
## strat0:flood_risk_pc4      -0.01402 -0.03116  0.00298
## strat0:flood_risk_pc5       0.03319  0.01684  0.04951
## strat0:EP_UNINSUR          -0.01004 -0.03859  0.01831
## strat0:pollute_conc_pc1    -0.44328 -0.50019 -0.38603
## strat0:pollute_conc_pc2    -0.50716 -0.58857 -0.42320
## strat0:pollute_conc_pc3    -0.20803 -0.28421 -0.13150
## strat0:tmmx                 0.00701 -0.11578  0.13401
## strat0:rmax                 0.14027  0.05140  0.22968
## strat0:Data_Value_CSMOKING  0.38040  0.34927  0.41179
## strat1:Intercept            6.86975  6.85294  6.88647
## strat1:flood_risk_pc1       0.06136  0.04195  0.08076
## strat1:flood_risk_pc2       0.01721 -0.00395  0.03831
## strat1:flood_risk_pc3       0.01855  0.00109  0.03596
## strat1:flood_risk_pc4      -0.00817 -0.02329  0.00690
## strat1:flood_risk_pc5       0.00507 -0.00985  0.02022
## strat1:EP_UNINSUR          -0.13925 -0.15807 -0.12059
## strat1:pollute_conc_pc1    -0.28208 -0.34033 -0.22425
## strat1:pollute_conc_pc2    -0.40678 -0.48702 -0.32435
## strat1:pollute_conc_pc3    -0.27466 -0.35137 -0.19772
## strat1:tmmx                 0.09583 -0.02746  0.22423
## strat1:rmax                 0.13802  0.04937  0.22850
## strat1:Data_Value_CSMOKING  0.85698  0.83338  0.88037
```

```
saveRDS(beta_inference, file = here("modeling_files/sensitivity_analysis/high_ver/beta_inference_files/
```

List of significant beta coefficients:

```
row.names(beta_inference)[sign(beta_inference[, 2]) == sign(beta_inference[, 3])]
```

```
##  [1] "strat0:Intercept"           "strat0:flood_risk_pc1"
##  [3] "strat0:flood_risk_pc2"      "strat0:flood_risk_pc3"
##  [5] "strat0:flood_risk_pc5"      "strat0:pollute_conc_pc1"
##  [7] "strat0:pollute_conc_pc2"    "strat0:pollute_conc_pc3"
##  [9] "strat0:rmax"                "strat0:Data_Value_CSMOKING"
## [11] "strat1:Intercept"           "strat1:flood_risk_pc1"
## [13] "strat1:flood_risk_pc3"      "strat1:EP_UNINSUR"
## [15] "strat1:pollute_conc_pc1"    "strat1:pollute_conc_pc2"
## [17] "strat1:pollute_conc_pc3"    "strat1:rmax"
## [19] "strat1: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$var_name <- substring(beta_inference_df$var_name, first = 8)
beta_inference_df$var_name <- factor(beta_inference_df$var_name,
                                     levels = unique(beta_inference_df$var_name))
beta_inference_df$strat <- factor(c(rep("Low SV", (nrow(beta_inference_df)/2)),
                                    rep("High SV", (nrow(beta_inference_df)/2))), levels = c("Low SV",
```

Splitting up the beta coefficients for each strata

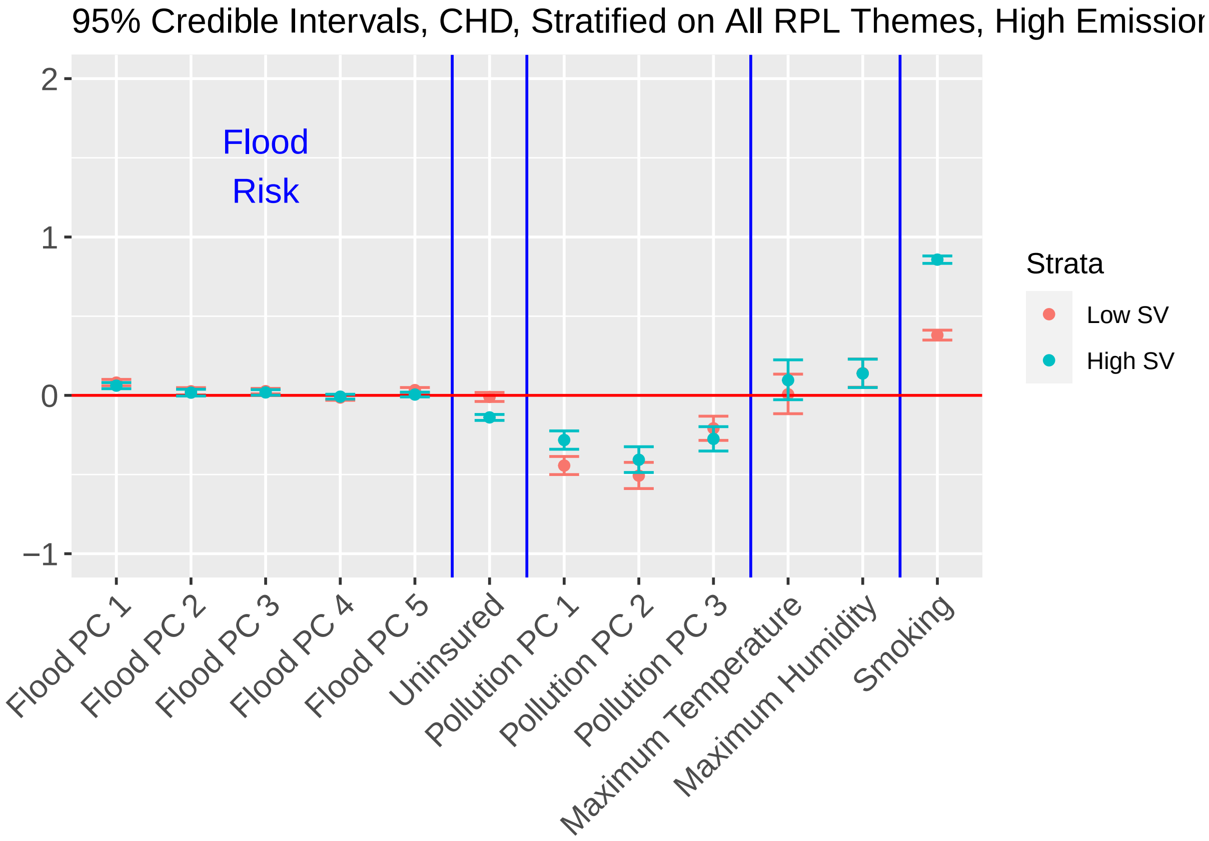
```
beta_inference_df_strat0 <- beta_inference_df[1:(nrow(beta_inference_df)/2),]
```

```
beta_inference_df_strat1 <- beta_inference_df[(nrow(beta_inference_df)/2 + 1):nrow(beta_inference_df),]
```

Note: The intercept for both strata is not included.

```
p <- ggplot(beta_inference_df_strat0[-1, ], aes(x = var_name, y = post_median, color = strat)) +
  geom_point() +
  ylim(c(-1, 2)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(), axi
        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), col = "#F8766D") +
  geom_vline(xintercept = c(5.5, 6.5, 9.5, 11.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3, y = 1.45, label = "Flood\nRisk",
           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",
                              "Uninsured",
                              "Pollution PC 1", "Pollution PC 2", "Pollution PC 3",
                              "Maximum Temperature", "Maximum Humidity",
                              "Smoking")) + ggtitle("95% Credible Intervals, CHD, Stratified on All RPL
  geom_point(data = beta_inference_df_strat1[-1, ], col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_inference_df_strat1[-1, ], aes(ymin = post_2.5, ymax = post_97.5, width = 0
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)

p
```

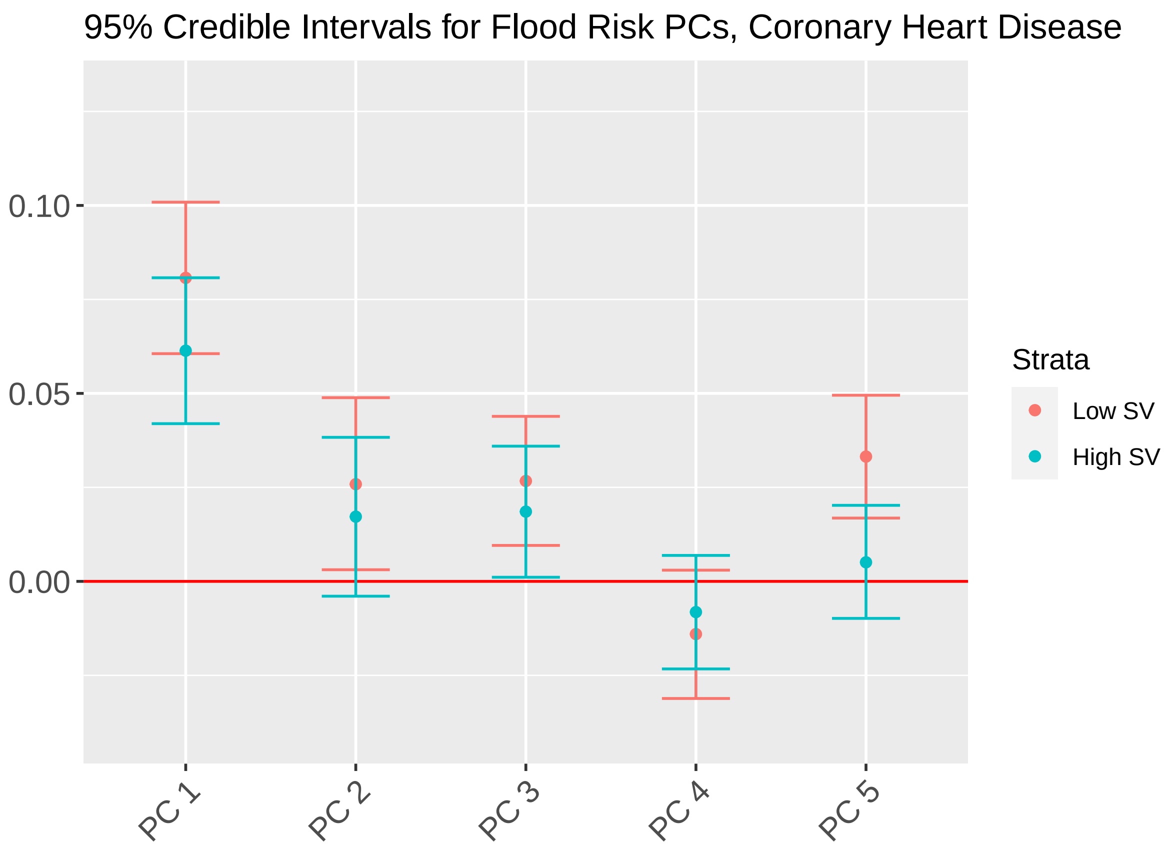


```
ggsave(here("figures/final_figures/sensitivity_analysis/high_ver/CHD_CI_rpls.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")
```

```
pc_extract_idx <- 2:6

beta_CHD_pcs <- beta_data_frames_extract(beta_inference_df, pc_extract_idx)
```

```
p <- ggplot(beta_CHD_pcs$beta_pcs_strat0, aes(x = var_idx, y = post_median, color = strat)) +
  geom_point() +
  ylim(c(-0.04, 0.13)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(), axi
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 25)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4), col = "#F8766D") +
  geom_hline(yintercept = 0, col = "red") +
  scale_x_discrete(labels = c("PC 1", "PC 2", "PC 3", "PC 4", "PC 5"), 6) + ggtitle("95% Credible Inter
  geom_point(data = beta_CHD_pcs$beta_pcs_strat1, col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_CHD_pcs$beta_pcs_strat1, aes(ymin = post_2.5, ymax = post_97.5, width = 0.4
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)
p
```



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

# CAR model results, High Blood Pressure Stratified on RPL_THEMES

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

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

colnames(beta_samples_matrix) <- names_high_ver_strat
```

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

pc_idx <- c(2:6,
            nrow(beta_inference)/2 + 2:6)

# 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%
## strat0:Intercept           31.04664 30.98839 31.10412
## strat0:flood_risk_pc1       0.15649  0.09534  0.21793
## strat0:flood_risk_pc2       0.11713  0.04821  0.18707
## strat0:flood_risk_pc3       0.07333  0.02154  0.12540
## strat0:flood_risk_pc4      -0.05738 -0.10924 -0.00606
## strat0:flood_risk_pc5       0.09633  0.04704  0.14558
## strat0:EP_UNINSUR          -0.09892 -0.18451 -0.01400
## strat0:pollute_conc_pc1    -1.67306 -1.85017 -1.49672
## strat0:pollute_conc_pc2    -2.35475 -2.60651 -2.09416
## strat0:pollute_conc_pc3     0.35826  0.11973  0.60037
## strat0:tmmx                -0.01358 -0.40663  0.39019
## strat0:rmax                 0.19766 -0.08941  0.48507
## strat0:Data_Value_CSMOKING  0.72682  0.63171  0.82232
## strat1:Intercept           32.84348 32.79415 32.89269
## strat1:flood_risk_pc1       0.00222 -0.05696  0.06108
## strat1:flood_risk_pc2      -0.11166 -0.17550 -0.04776
## strat1:flood_risk_pc3       0.03116 -0.02163  0.08360
## strat1:flood_risk_pc4       0.04313 -0.00263  0.08822
## strat1:flood_risk_pc5       0.04644  0.00124  0.09198
## strat1:EP_UNINSUR          -0.47536 -0.53199 -0.41888
## strat1:pollute_conc_pc1    -0.80665 -0.98769 -0.62936
## strat1:pollute_conc_pc2    -2.23399 -2.48297 -1.97638
## strat1:pollute_conc_pc3    -0.19665 -0.43742  0.04640
## strat1:tmmx                 0.50752  0.11302  0.91234
## strat1:rmax                 0.27989 -0.00683  0.56842
## strat1:Data_Value_CSMOKING  2.12791  2.05591  2.19925
```

```
saveRDS(beta_inference, file = here("modeling_files/sensitivity_analysis/high_ver/beta_inference_files/
```

List of significant beta coefficients:

```
row.names(beta_inference)[sign(beta_inference[, 2]) == sign(beta_inference[, 3])]
```

```
##  [1] "strat0:Intercept"           "strat0:flood_risk_pc1"
##  [3] "strat0:flood_risk_pc2"      "strat0:flood_risk_pc3"
##  [5] "strat0:flood_risk_pc4"      "strat0:flood_risk_pc5"
##  [7] "strat0:EP_UNINSUR"          "strat0:pollute_conc_pc1"
##  [9] "strat0:pollute_conc_pc2"    "strat0:pollute_conc_pc3"
## [11] "strat0:Data_Value_CSMOKING" "strat1:Intercept"
## [13] "strat1:flood_risk_pc2"      "strat1:flood_risk_pc5"
## [15] "strat1:EP_UNINSUR"          "strat1:pollute_conc_pc1"
## [17] "strat1:pollute_conc_pc2"    "strat1:tmmx"
## [19] "strat1: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$var_name <- substring(beta_inference_df$var_name, first = 8)
beta_inference_df$var_name <- factor(beta_inference_df$var_name,
                                     levels = unique(beta_inference_df$var_name))
beta_inference_df$strat <- factor(c(rep("Low SV", (nrow(beta_inference_df)/2)),
                                    rep("High SV", (nrow(beta_inference_df)/2))), levels = c("Low SV",
```

Splitting up the beta coefficients for each strata

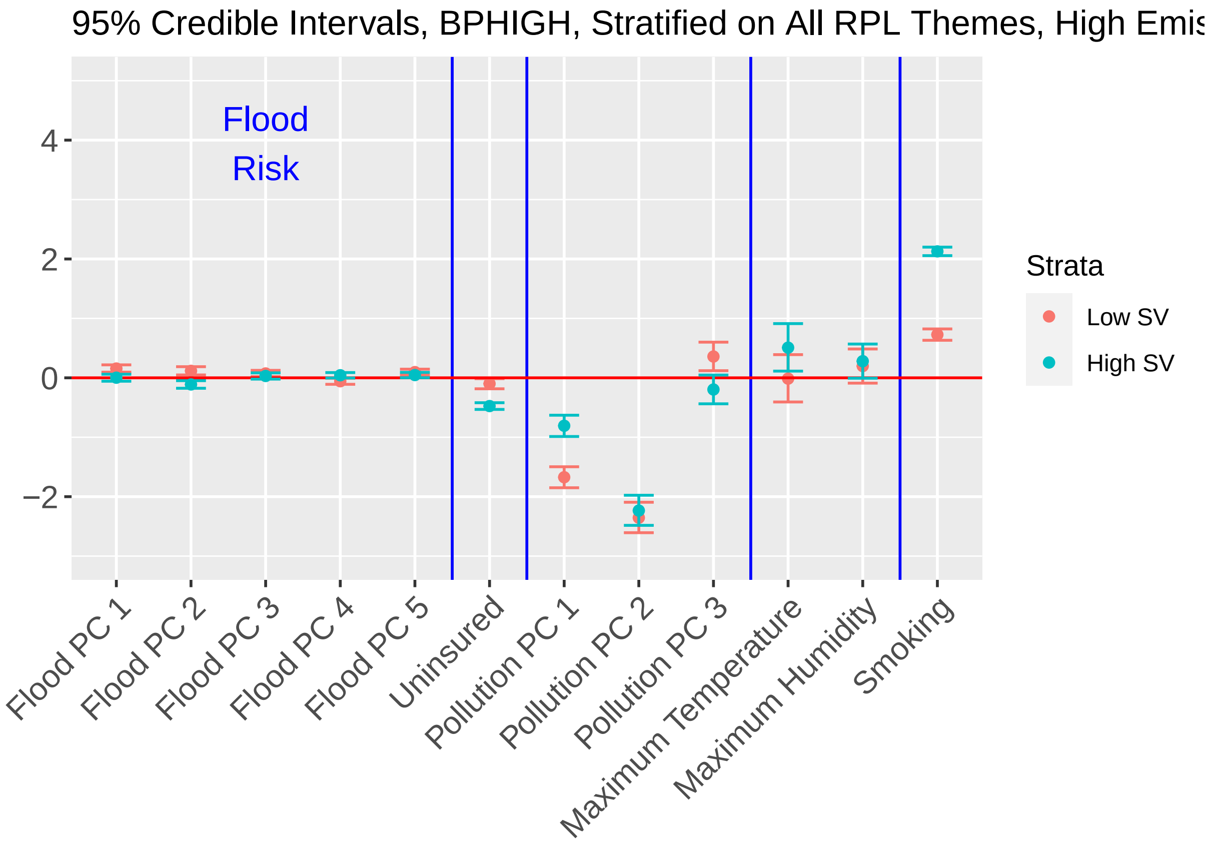
```
beta_inference_df_strat0 <- beta_inference_df[1:(nrow(beta_inference_df)/2),]

beta_inference_df_strat1 <- beta_inference_df[(nrow(beta_inference_df)/2 + 1):nrow(beta_inference_df),]
```

Note: The intercept for both strata is not included.

```
p <- ggplot(beta_inference_df_strat0[-1, ], aes(x = var_name, y = post_median, color = strat)) +
  geom_point() +
  ylim(c(-3, 5)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(), axi
        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), col = "#F8766D") +
  geom_vline(xintercept = c(5.5, 6.5, 9.5, 11.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3, y = 3.95, label = "Flood\nRisk",
           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",
                              "Uninsured",
                              "Pollution PC 1", "Pollution PC 2", "Pollution PC 3",
                              "Maximum Temperature", "Maximum Humidity",
                              "Smoking")) + ggtitle("95% Credible Intervals, BPHIGH, Stratified on All
  geom_point(data = beta_inference_df_strat1[-1, ], col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_inference_df_strat1[-1, ], aes(ymin = post_2.5, ymax = post_97.5, width = 0
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)

p
```

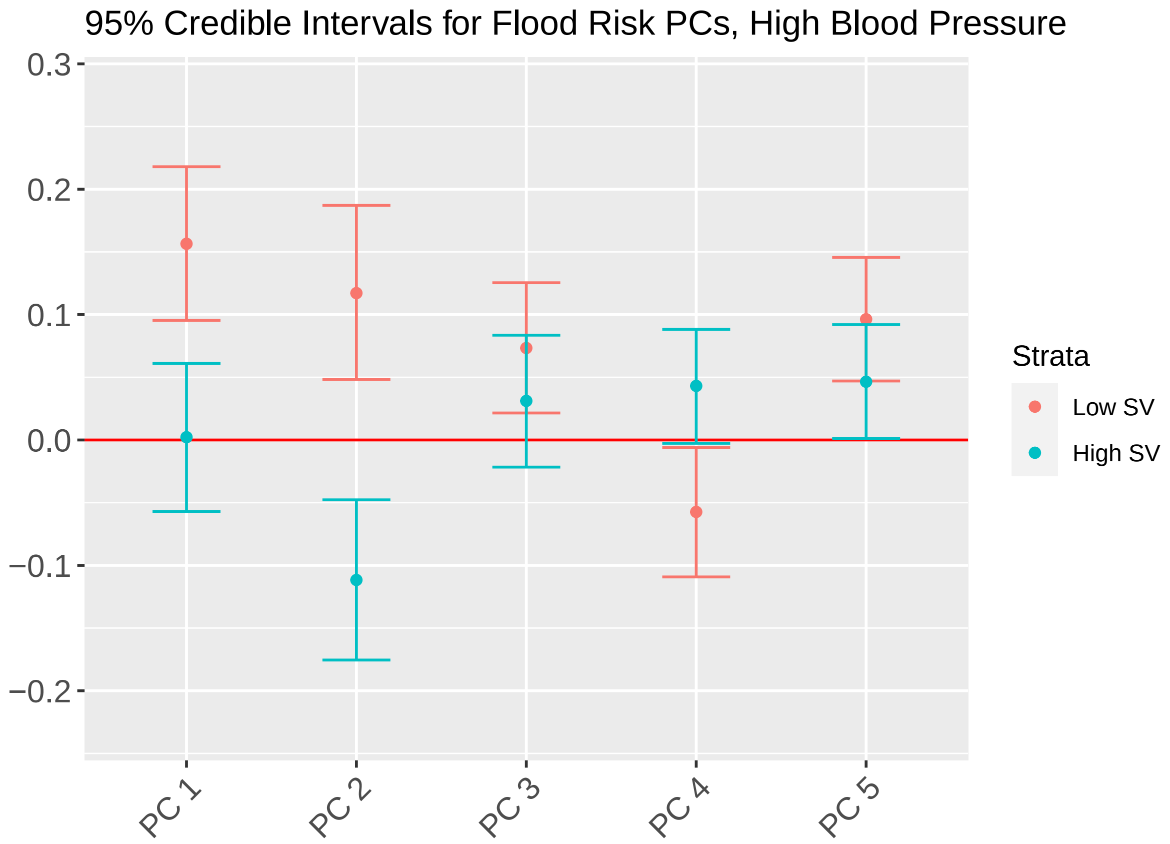


```r
ggsave(here("figures/final_figures/sensitivity_analysis/high_ver/BPHIGH_CI_rpls.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")
```

```r
pc_extract_idx <- 2:6

beta_BPHIGH_pcs <- beta_data_frames_extract(beta_inference_df, pc_extract_idx)
```

```r
p <- ggplot(beta_BPHIGH_pcs$beta_pcs_strat0, aes(x = var_idx, y = post_median, color = strat)) +
  geom_point() +
  ylim(c(-0.23, 0.28)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(), axis
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 25)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4), col = "#F8766D") +
  geom_hline(yintercept = 0, col = "red") +
  scale_x_discrete(labels = c("PC 1", "PC 2", "PC 3", "PC 4", "PC 5"), 6) + ggtitle("95% Credible Inter
  geom_point(data = beta_BPHIGH_pcs$beta_pcs_strat1, col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_BPHIGH_pcs$beta_pcs_strat1, aes(ymin = post_2.5, ymax = post_97.5, width = 
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)
p
```



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

## CAR model results, Asthma Stratified on RPL_THEMES

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

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

colnames(beta_samples_matrix) <- names_high_ver_strat
```

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

pc_idx <- c(2:6,
            nrow(beta_inference)/2 + 2:6)

# 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%
## strat0:Intercept              9.76067  9.75386  9.76740
## strat0:flood_risk_pc1        -0.00513 -0.01235  0.00214
## strat0:flood_risk_pc2         0.01101  0.00283  0.01930
## strat0:flood_risk_pc3       0.00039 -0.00574  0.00658
## strat0:flood_risk_pc4       0.01407  0.00793  0.02016
## strat0:flood_risk_pc5       0.00108 -0.00476  0.00690
## strat0:EP_UNINSUR          -0.02866 -0.03879 -0.01862
## strat0:pollute_conc_pc1     0.11377  0.09266  0.13466
## strat0:pollute_conc_pc2    -0.17270 -0.20273 -0.14183
## strat0:pollute_conc_pc3    -0.05373 -0.08212 -0.02486
## strat0:tmmx                -0.00219 -0.04952  0.04673
## strat0:rmax                -0.08509 -0.12018 -0.05032
## strat0:Data_Value_CSMOKING  1.12441  1.11311  1.13577
## strat1:Intercept            9.94422  9.93845  9.95001
## strat1:flood_risk_pc1      -0.01515 -0.02220 -0.00816
## strat1:flood_risk_pc2      -0.02591 -0.03344 -0.01831
## strat1:flood_risk_pc3       0.00554 -0.00071  0.01178
## strat1:flood_risk_pc4       0.01659  0.01116  0.02193
## strat1:flood_risk_pc5       0.00328 -0.00209  0.00864
## strat1:EP_UNINSUR          -0.08547 -0.09218 -0.07878
## strat1:pollute_conc_pc1     0.17834  0.15668  0.19927
## strat1:pollute_conc_pc2    -0.16687 -0.19643 -0.13622
## strat1:pollute_conc_pc3    -0.10763 -0.13643 -0.07857
## strat1:tmmx                 0.04361 -0.00405  0.09273
## strat1:rmax                -0.12060 -0.15586 -0.08595
## strat1:Data_Value_CSMOKING  1.31082  1.30230  1.31932
```

```
saveRDS(beta_inference, file = here("modeling_files/sensitivity_analysis/high_ver/beta_inference_files/
```

List of significant beta coefficients:

```
row.names(beta_inference)[sign(beta_inference[, 2]) == sign(beta_inference[, 3])]
```

```
##  [1] "strat0:Intercept"           "strat0:flood_risk_pc2"
##  [3] "strat0:flood_risk_pc4"      "strat0:EP_UNINSUR"
##  [5] "strat0:pollute_conc_pc1"    "strat0:pollute_conc_pc2"
##  [7] "strat0:pollute_conc_pc3"    "strat0:rmax"
##  [9] "strat0:Data_Value_CSMOKING" "strat1:Intercept"
## [11] "strat1:flood_risk_pc1"      "strat1:flood_risk_pc2"
## [13] "strat1:flood_risk_pc4"      "strat1:EP_UNINSUR"
## [15] "strat1:pollute_conc_pc1"    "strat1:pollute_conc_pc2"
## [17] "strat1:pollute_conc_pc3"    "strat1:rmax"
## [19] "strat1: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$var_name <- substring(beta_inference_df$var_name, first = 8)
beta_inference_df$var_name <- factor(beta_inference_df$var_name,
                                     levels = unique(beta_inference_df$var_name))
beta_inference_df$strat <- factor(c(rep("Low SV", (nrow(beta_inference_df)/2)),
                                    rep("High SV", (nrow(beta_inference_df)/2))), levels = c("Low SV",
```

Splitting up the beta coefficients for each strata

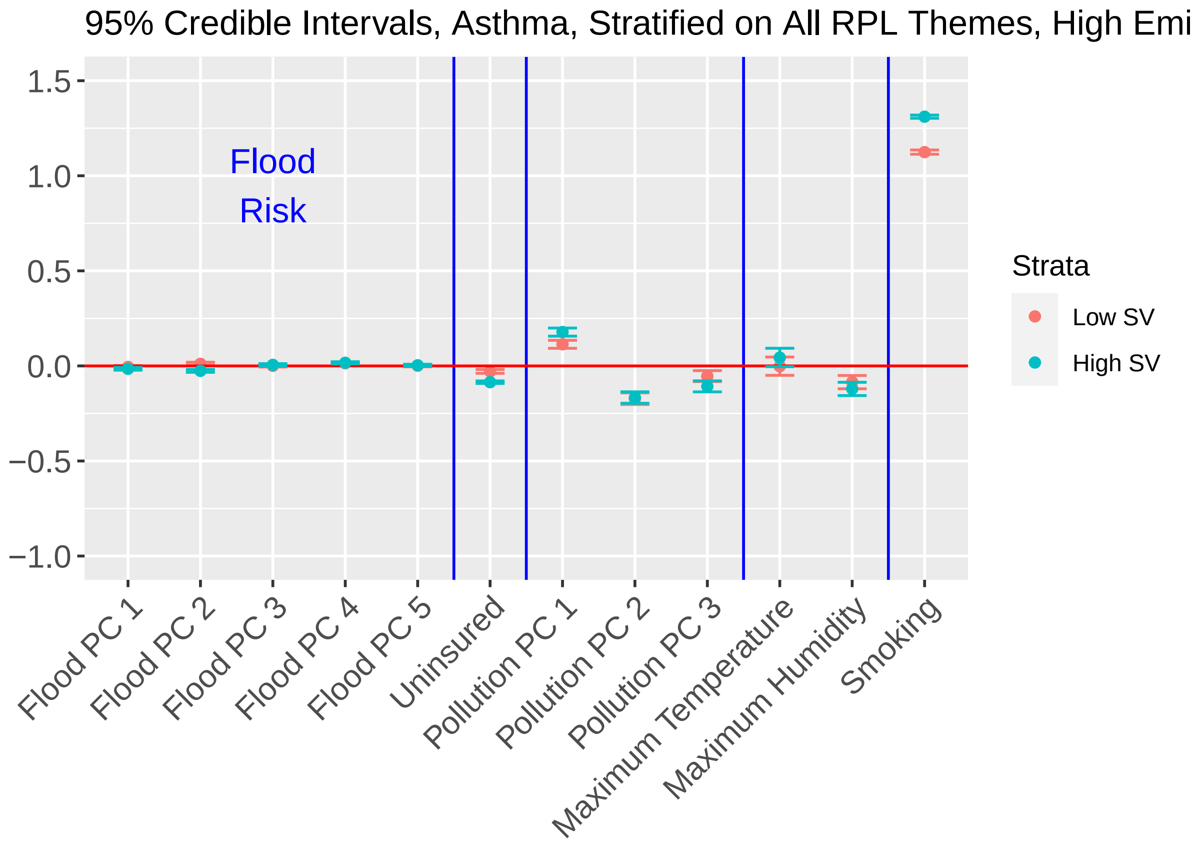
```
beta_inference_df_strat0 <- beta_inference_df[1:(nrow(beta_inference_df)/2),]

beta_inference_df_strat1 <- beta_inference_df[(nrow(beta_inference_df)/2 + 1):nrow(beta_inference_df),]
```

Note: The intercept for both strata is not included.

```
p <- ggplot(beta_inference_df_strat0[-1, ], aes(x = var_name, y = post_median, color = strat)) +
  geom_point() +
  ylim(c(-1, 1.5)) +
  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), col = "#F8766D") +
  geom_vline(xintercept = c(5.5, 6.5, 9.5, 11.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3, y = 0.95, label = "Flood\nRisk",
           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",
                              "Uninsured",
                              "Pollution PC 1", "Pollution PC 2", "Pollution PC 3",
                              "Maximum Temperature", "Maximum Humidity",
                              "Smoking")) + ggtitle("95% Credible Intervals, Asthma, Stratified on All 
  geom_point(data = beta_inference_df_strat1[-1, ], col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_inference_df_strat1[-1, ], aes(ymin = post_2.5, ymax = post_97.5, width = 0
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)

p
```

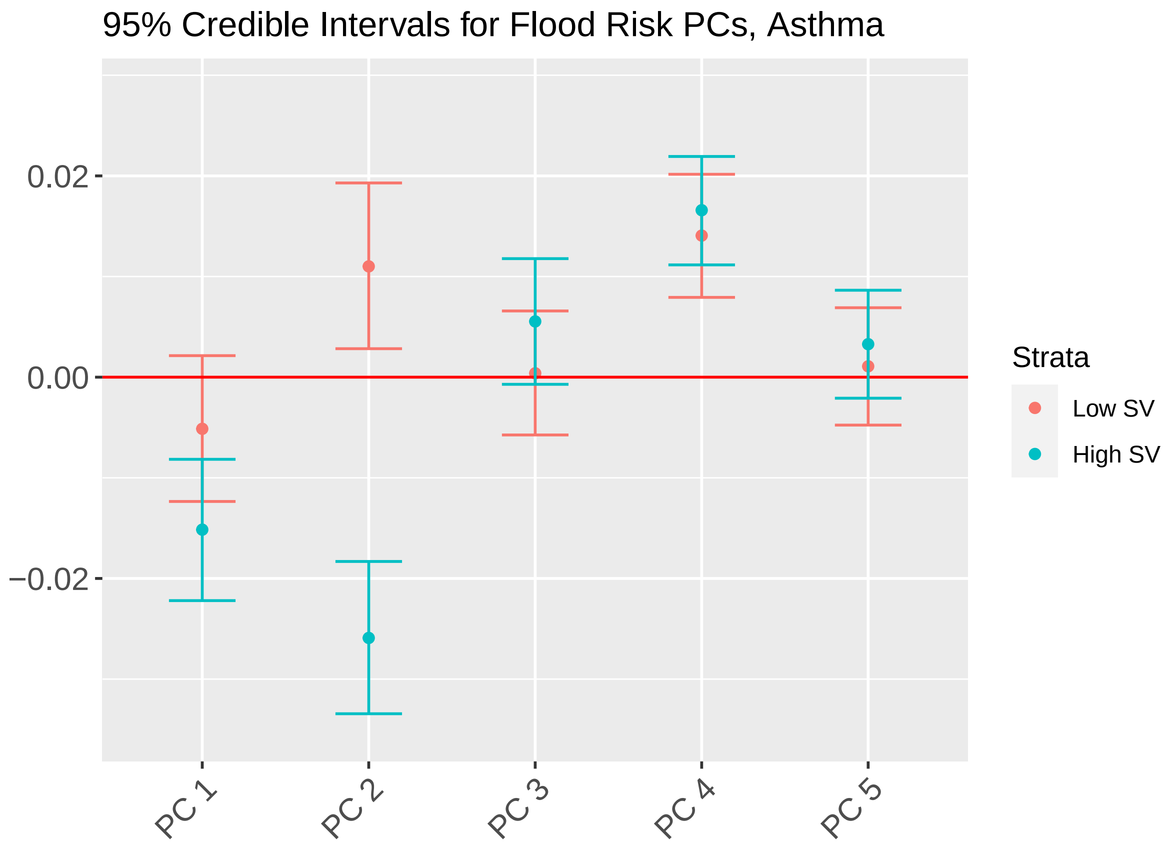


```
ggsave(here("figures/final_figures/sensitivity_analysis/high_ver/CASTHMA_CI_rpls.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")
```

```
pc_extract_idx <- 2:6

beta_CASTHMA_pcs <- beta_data_frames_extract(beta_inference_df, pc_extract_idx)
```

```
p <- ggplot(beta_CASTHMA_pcs$beta_pcs_strat0, aes(x = var_idx, y = post_median, color = strat)) +
  geom_point() +
  ylim(c(-0.035, 0.0285)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(), axi
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 25)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4), col = "#F8766D") +
  geom_hline(yintercept = 0, col = "red") +
  scale_x_discrete(labels = c("PC 1", "PC 2", "PC 3", "PC 4", "PC 5"), 6) + ggtitle("95% Credible Inter
  geom_point(data = beta_CASTHMA_pcs$beta_pcs_strat1, col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_CASTHMA_pcs$beta_pcs_strat1, aes(ymin = post_2.5, ymax = post_97.5, width =
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)
p
```



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

## CAR model results, Poor Mental Health Stratified on RPL_THEMES

```r
load(here("modeling_files/sensitivity_analysis/high_ver/all_census_tract_MHLTH.RData"))
```

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

colnames(beta_samples_matrix) <- names_high_ver_strat
```

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

pc_idx <- c(2:6,
            nrow(beta_inference)/2 + 2:6)

# 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%
## strat0:Intercept           14.15603 14.14327 14.16854
## strat0:flood_risk_pc1       0.00115 -0.01246  0.01477
## strat0:flood_risk_pc2       0.02620  0.01082  0.04181
## strat0:flood_risk_pc3      -0.02465 -0.03617 -0.01307
## strat0:flood_risk_pc4       0.01706  0.00555  0.02849
## strat0:flood_risk_pc5      -0.01210 -0.02308 -0.00118
## strat0:EP_UNINSUR           0.01699 -0.00208  0.03574
## strat0:pollute_conc_pc1     0.65015  0.61018  0.68935
## strat0:pollute_conc_pc2     0.37453  0.31818  0.43217
## strat0:pollute_conc_pc3    -0.50832 -0.56184 -0.45436
## strat0:tmmx                 0.04162 -0.04887  0.13501
## strat0:rmax                -0.08791 -0.15531 -0.02192
## strat0:Data_Value_CSMOKING  3.32357  3.30236  3.34499
## strat1:Intercept           14.34794 14.33718 14.35872
## strat1:flood_risk_pc1       0.00454 -0.00872  0.01770
## strat1:flood_risk_pc2      -0.01139 -0.02556  0.00287
## strat1:flood_risk_pc3      -0.00555 -0.01729  0.00619
## strat1:flood_risk_pc4       0.00465 -0.00556  0.01467
## strat1:flood_risk_pc5      -0.00497 -0.01508  0.00511
## strat1:EP_UNINSUR           0.04671  0.03417  0.05928
## strat1:pollute_conc_pc1     0.68869  0.64730  0.72781
## strat1:pollute_conc_pc2     0.38483  0.32932  0.44219
## strat1:pollute_conc_pc3    -0.48858 -0.54302 -0.43426
## strat1:tmmx                 0.09022 -0.00084  0.18392
## strat1:rmax                -0.09627 -0.16385 -0.03063
## strat1:Data_Value_CSMOKING  3.28632  3.27031  3.30236
```

```
saveRDS(beta_inference, file = here("modeling_files/sensitivity_analysis/high_ver/beta_inference_files/
```

List of significant beta coefficients:

```
row.names(beta_inference)[sign(beta_inference[, 2]) == sign(beta_inference[, 3])]
```

```
##  [1] "strat0:Intercept"           "strat0:flood_risk_pc2"
##  [3] "strat0:flood_risk_pc3"      "strat0:flood_risk_pc4"
##  [5] "strat0:flood_risk_pc5"      "strat0:pollute_conc_pc1"
##  [7] "strat0:pollute_conc_pc2"    "strat0:pollute_conc_pc3"
##  [9] "strat0:rmax"                "strat0:Data_Value_CSMOKING"
## [11] "strat1:Intercept"           "strat1:EP_UNINSUR"
## [13] "strat1:pollute_conc_pc1"    "strat1:pollute_conc_pc2"
## [15] "strat1:pollute_conc_pc3"    "strat1:rmax"
## [17] "strat1: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$var_name <- substring(beta_inference_df$var_name, first = 8)
beta_inference_df$var_name <- factor(beta_inference_df$var_name,
                                     levels = unique(beta_inference_df$var_name))
beta_inference_df$strat <- factor(c(rep("Low SV", (nrow(beta_inference_df)/2)),
                                    rep("High SV", (nrow(beta_inference_df)/2))), levels = c("Low SV",
```

Splitting up the beta coefficients for each strata

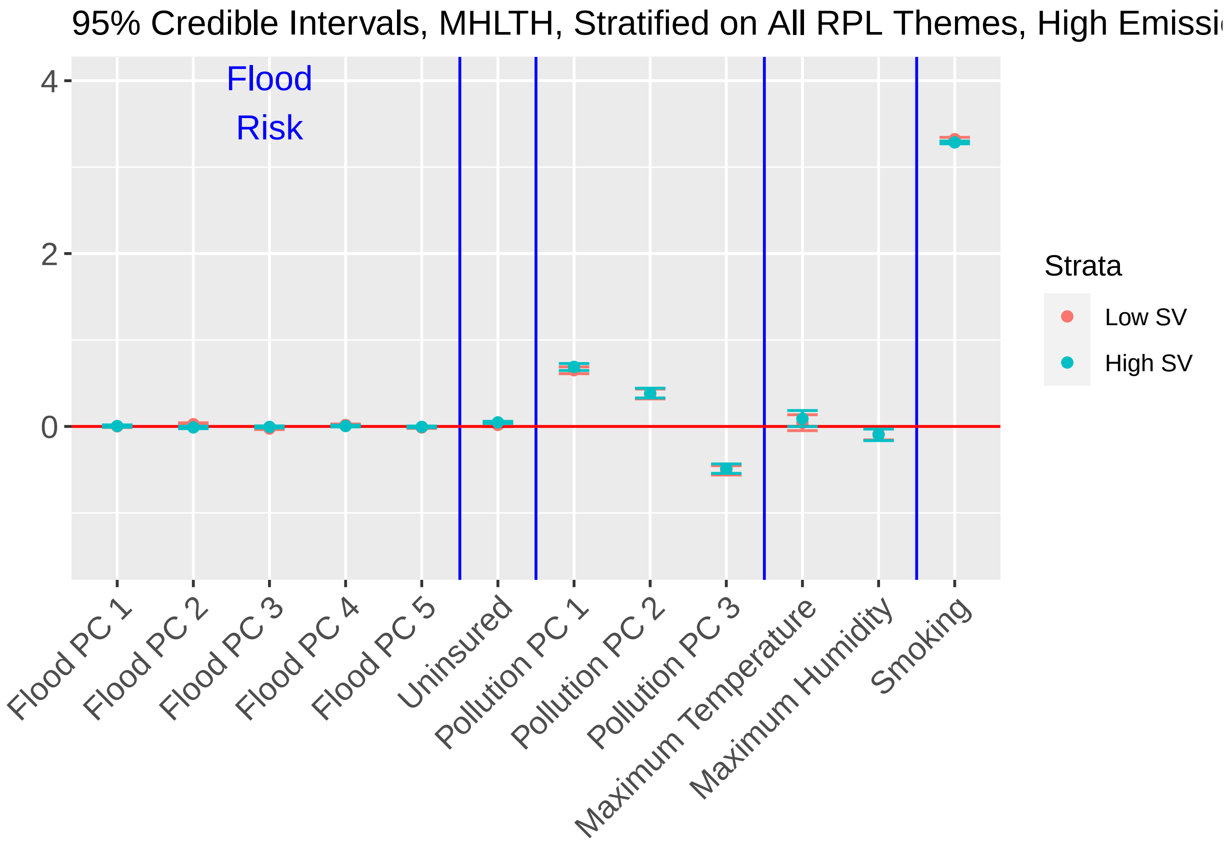
```
beta_inference_df_strat0 <- beta_inference_df[1:(nrow(beta_inference_df)/2),]

beta_inference_df_strat1 <- beta_inference_df[(nrow(beta_inference_df)/2 + 1):nrow(beta_inference_df),]
```

Note: The intercept for both strata is not included.

```
p <- ggplot(beta_inference_df_strat0[-1, ], aes(x = var_name, y = post_median, color = strat)) +
  geom_point() +
  ylim(c(-1.5, 4)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(), axi
        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), col = "#F8766D") +
  geom_vline(xintercept = c(5.5, 6.5, 9.5, 11.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3, y = 3.75, label = "Flood\nRisk",
           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",
                              "Uninsured",
                              "Pollution PC 1", "Pollution PC 2", "Pollution PC 3",
                              "Maximum Temperature", "Maximum Humidity",
                              "Smoking")) + ggtitle("95% Credible Intervals, MHLTH, Stratified on All R
  geom_point(data = beta_inference_df_strat1[-1, ], col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_inference_df_strat1[-1, ], aes(ymin = post_2.5, ymax = post_97.5, width = 0
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)

p
```

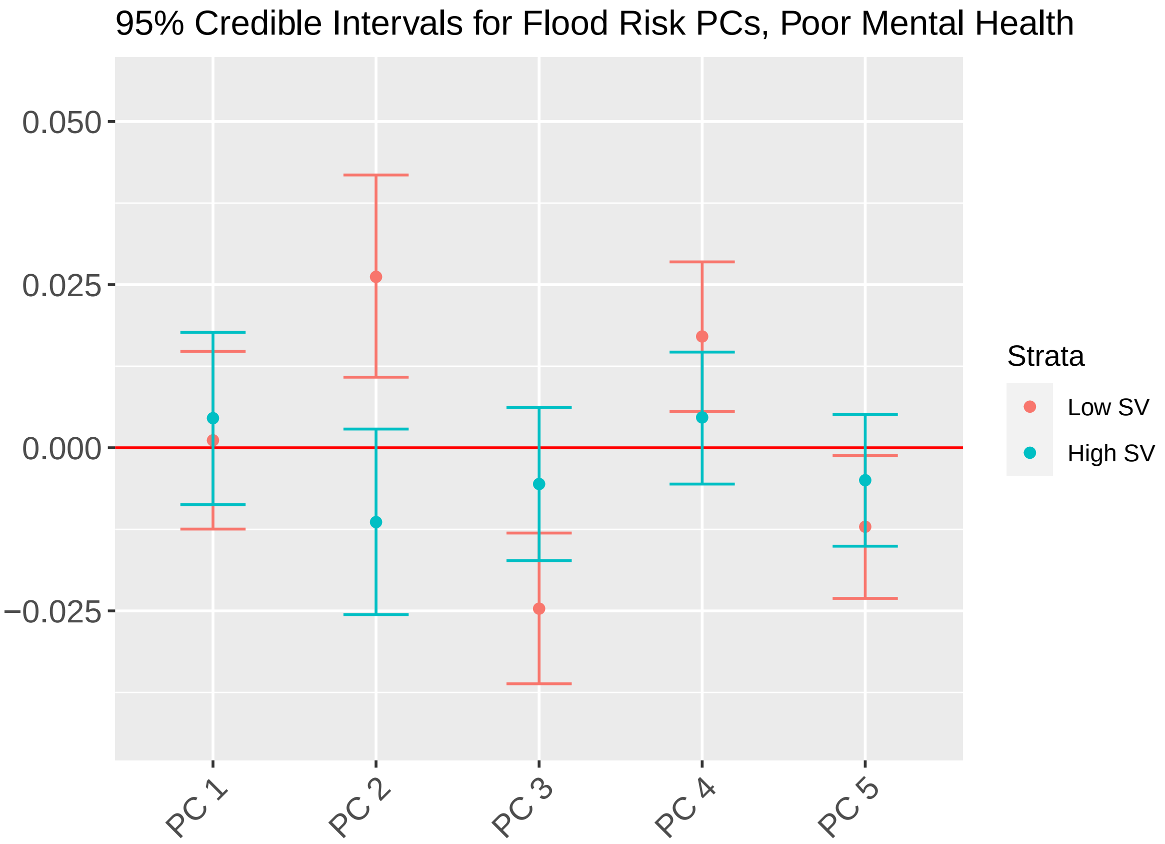


```r
ggsave(here("figures/final_figures/sensitivity_analysis/high_ver/MHLTH_CI_rpls.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")
```

```r
pc_extract_idx <- 2:6

beta_MHLTH_pcs <- beta_data_frames_extract(beta_inference_df, pc_extract_idx)
```

```r
p <- ggplot(beta_MHLTH_pcs$beta_pcs_strat0, aes(x = var_idx, y = post_median, color = strat)) +
  geom_point() +
  ylim(c(-0.043, 0.055)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(), axi
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 25)) +
  geom_errorbar(aes(ymin = post_2.5, ymax = post_97.5, width = 0.4), col = "#F8766D") +
  geom_hline(yintercept = 0, col = "red") +
  scale_x_discrete(labels = c("PC 1", "PC 2", "PC 3", "PC 4", "PC 5"), 6) + ggtitle("95% Credible Inter
  geom_point(data = beta_MHLTH_pcs$beta_pcs_strat1, col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_MHLTH_pcs$beta_pcs_strat1, aes(ymin = post_2.5, ymax = post_97.5, width = 0
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)
p
```



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