Sensitivity Analysis: Non-Spatial Models

Alvin Sheng

Contents

Modeling Set-up

```
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(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()
```

```
library(spdep)
```

```
## Loading required package: sp
## Loading required package: spData
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
## Loading required package: sf
## Linking to GEOS 3.8.1, GDAL 3.2.1, PROJ 7.2.1
```

```
fhs_model_df <- readRDS(here("intermediary_data/fhs_model_df_fr_and_pollute_pc.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")
```

Reading in adjacency matrix

```
W <- readRDS(file = here("intermediary_data", "census_tract_adj_reorganize_all_census_tract.rds"))

# converting it to a form mat2listw() can understand
W <- as(W, "dMatrix")

# converting it to a form to be used in moran.test()
W_listw <- mat2listw(W)
```

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

```r
# extract the four response variables
responses <- fhs_model_df[, (ncol(fhs_model_df) - 3):ncol(fhs_model_df)]
# extract the other covariates, except the SVI variables
covariates <- select(fhs_model_df[, -((ncol(fhs_model_df) - 3):ncol(fhs_model_df))], -EP_POV,
                     -EP_UNEMP, -EP_PCI, -EP_NOHSDP,
                       -EP_AGE65, -EP_AGE17, -EP_DISABL, -EP_SNGPNT,
                       -EP_MINRTY, -EP_LIMENG,
                       -EP_MUNIT, -EP_MOBILE, -EP_CROWD, -EP_NOVEH, -EP_GROUPQ)

first_var <- which(names(covariates) == "flood_risk_pc1")

strat_covariate <- fhs_model_df$RPL_THEMES
```

```r
# CHD
covariates_CHD <- data.frame(covariates, Data_Value_CHD = responses$Data_Value_CHD)
# BPHIGH
covariates_BPHIGH <- data.frame(covariates, Data_Value_BPHIGH = responses$Data_Value_BPHIGH)
# CASTHMA
covariates_CASTHMA <- data.frame(covariates, Data_Value_CASTHMA = responses$Data_Value_CASTHMA)
# MHLTH
covariates_MHLTH <- data.frame(covariates, Data_Value_MHLTH = responses$Data_Value_MHLTH)
```

Helper Functions

Function to run the stratified non-spatial model

```r
source(here("scripts/sensitivity_analysis/non_spatial_strat_model.R"))
```

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 = "*"))
}
```

```r
# 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, ]
```

```r
  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

```r
chd_res <- non_spatial_strat_model(covariates_CHD, first_var, strat_covariate, strat_fn = median)

beta_results <- summary(chd_res$lm_obj)$coefficients
```

Moran's I test for spatial autocorrelation

```r
# mean imputing the residuals corresponding to missing Y values
resid_w_impute <- rep(NA, nrow(fhs_model_df))

resid_w_impute[!is.na(fhs_model_df$Data_Value_CHD)] <- chd_res$lm_obj$residuals

resid_w_impute[is.na(fhs_model_df$Data_Value_CHD)] <- 0
```

```r
(moran_results <- moran.test(resid_w_impute, W_listw))
```

```
##
##  Moran I test under randomisation
##
## data:  resid_w_impute
## weights: W_listw
##
## Moran I statistic standard deviate = 199.25, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      4.167454e-01     -1.378626e-05      4.375076e-06
```

The p-value is negligible, so we can reject the null hypothesis of zero spatial autocorrelation. Since the observed value of the Moran I statistic is significantly greater then the expected value, the residuals are positively autocorrelated, in contrast to negatively autocorrelated. Thus, using a CAR model is justified.

```r
row.names(beta_results) <- names_non_spat_strat
```

```r
beta_inference <- cbind(beta_results[, 1],
                        beta_results[, 1] - 2 * beta_results[, 2],
                        beta_results[, 1] + 2 * beta_results[, 2])

colnames(beta_inference) <- c("pt_est", "lb", "ub")

beta_inference <- round(beta_inference, 5)
```

```r
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
```

```
##                              pt_est        lb        ub
## strat0:Intercept            6.20734   6.18203   6.23266
## strat0:flood_risk_pc1       0.29900   0.27898   0.31901
## strat0:flood_risk_pc2       0.12765   0.10411   0.15120
## strat0:flood_risk_pc3       0.00772  -0.01139   0.02683
## strat0:flood_risk_pc4      -0.05570  -0.07576  -0.03564
## strat0:flood_risk_pc5       0.04084   0.02105   0.06063
## strat0:EP_UNINSUR          -0.14258  -0.17638  -0.10879
## strat0:pollute_conc_pc1    -0.50207  -0.52805  -0.47609
## strat0:pollute_conc_pc2    -0.14877  -0.17189  -0.12566
## strat0:pollute_conc_pc3    -0.11643  -0.13951  -0.09334
## strat0:tmmx                 0.27562   0.25326   0.29798
## strat0:rmax                 0.01957  -0.00301   0.04216
## strat0:Data_Value_CSMOKING  0.90161   0.87311   0.93010
## strat1:Intercept            6.96475   6.94253   6.98696
## strat1:flood_risk_pc1       0.16400   0.14415   0.18384
## strat1:flood_risk_pc2       0.18791   0.16728   0.20854
## strat1:flood_risk_pc3      -0.00212  -0.02170   0.01746
## strat1:flood_risk_pc4      -0.08027  -0.09786  -0.06268
## strat1:flood_risk_pc5      -0.03117  -0.04901  -0.01333
## strat1:EP_UNINSUR          -0.21052  -0.22967  -0.19138
## strat1:pollute_conc_pc1    -0.37572  -0.39552  -0.35592
## strat1:pollute_conc_pc2    -0.07416  -0.09699  -0.05133
## strat1:pollute_conc_pc3    -0.12533  -0.14516  -0.10550
## strat1:tmmx                 0.42498   0.40214   0.44783
## strat1:rmax                 0.02574   0.00535   0.04614
## strat1:Data_Value_CSMOKING  1.19944   1.17728   1.22161
```

List of significant beta coefficients:

```r
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_pc4"
##  [5] "strat0:flood_risk_pc5"      "strat0:EP_UNINSUR"
##  [7] "strat0:pollute_conc_pc1"    "strat0:pollute_conc_pc2"
##  [9] "strat0:pollute_conc_pc3"    "strat0:tmmx"
## [11] "strat0:Data_Value_CSMOKING" "strat1:Intercept"
## [13] "strat1:flood_risk_pc1"      "strat1:flood_risk_pc2"
## [15] "strat1:flood_risk_pc4"      "strat1:flood_risk_pc5"
## [17] "strat1:EP_UNINSUR"          "strat1:pollute_conc_pc1"
## [19] "strat1:pollute_conc_pc2"    "strat1:pollute_conc_pc3"
## [21] "strat1:tmmx"                "strat1:rmax"
## [23] "strat1:Data_Value_CSMOKING"
```

Credible Interval plots for the coefficients, in ggplot

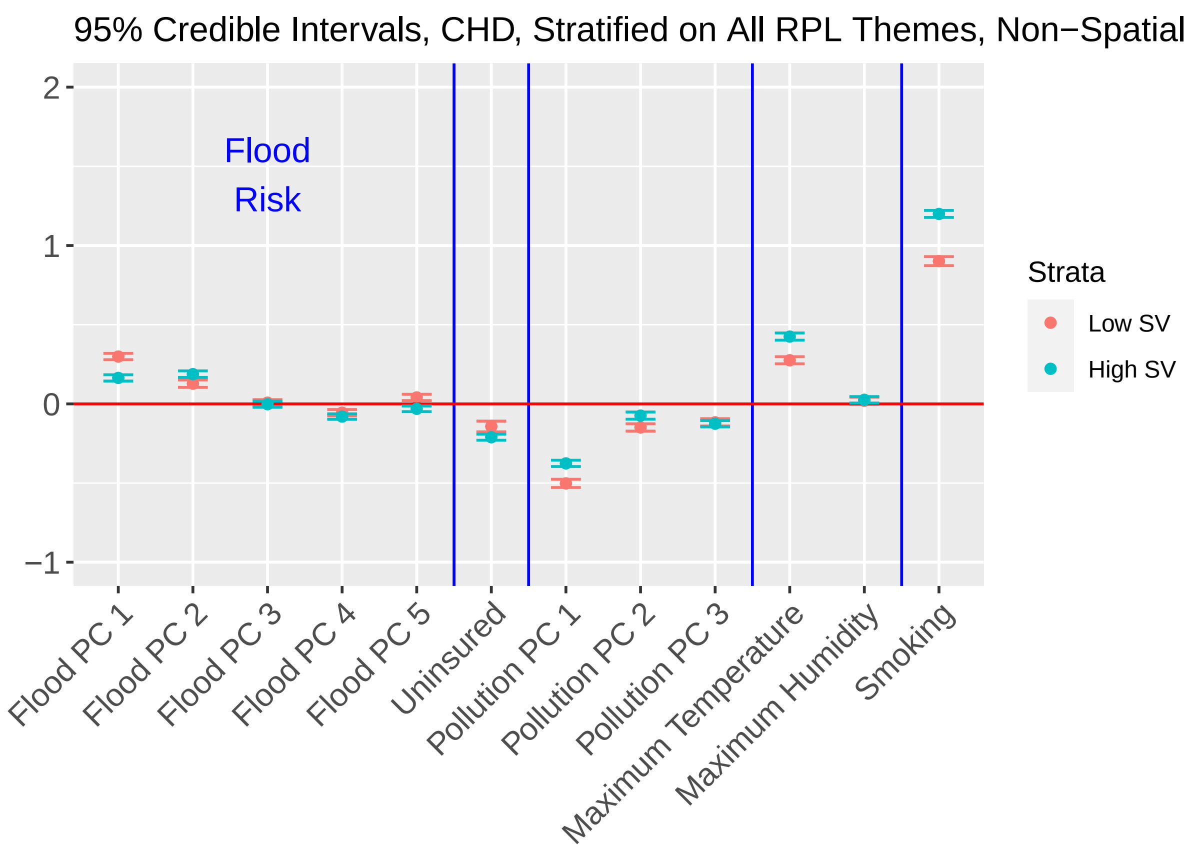
```
# first, process the beta_inference matrix in a form ggplot can understand
beta_inference_df <- as.data.frame(beta_inference)
beta_inference_df <- mutate(beta_inference_df, var_name = row.names(beta_inference_df))
beta_inference_df$var_name <- substring(beta_inference_df$var_name, first = 8)
beta_inference_df$var_name <- factor(beta_inference_df$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.

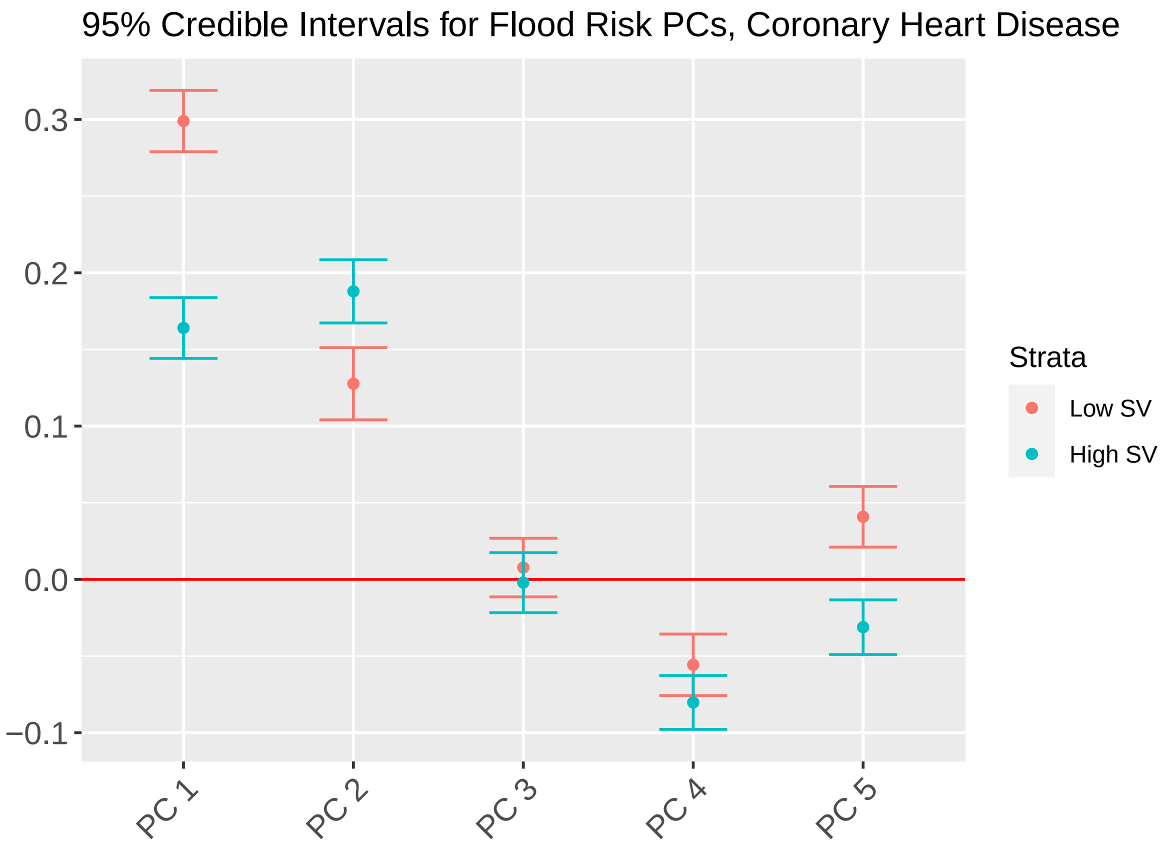
```
p <- ggplot(beta_inference_df_strat0[-1, ], aes(x = var_name, y = pt_est, color = strat)) +
  geom_point() +
  ylim(c(-1, 2)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(), axi
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 25)) +
  geom_errorbar(aes(ymin = lb, ymax = ub, 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 = lb, ymax = ub, width = 0.4), col = "#
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)

p
```




```
ggsave(here("figures/final_figures/sensitivity_analysis/non_spatial/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$beta_pcs_strat0, aes(x = var_idx, y = pt_est, color = strat)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1), axis.title.x = element_blank(), axis
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 25)) +
  geom_errorbar(aes(ymin = lb, ymax = ub, width = 0.4), col = "#F8766D") +
  geom_hline(yintercept = 0, col = "red") +
  scale_x_discrete(labels = c("PC 1", "PC 2", "PC 3", "PC 4", "PC 5"), 6) + ggtitle("95% Credible Inter
  geom_point(data = beta_CHD_pcs$beta_pcs_strat1, col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_CHD_pcs$beta_pcs_strat1, aes(ymin = lb, ymax = ub, width = 0.4), col = "#00
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)
p
```




```
ggsave(here("figures/final_figures/sensitivity_analysis/non_spatial/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

```
bphigh_res <- non_spatial_strat_model(covariates_BPHIGH, first_var, strat_covariate, strat_fn = median)

beta_results <- summary(bphigh_res$lm_obj)$coefficients
```

Moran's I test for spatial autocorrelation

```
# mean imputing the residuals corresponding to missing Y values
resid_w_impute <- rep(NA, nrow(fhs_model_df))

resid_w_impute[!is.na(fhs_model_df$Data_Value_BPHIGH)] <- bphigh_res$lm_obj$residuals

resid_w_impute[is.na(fhs_model_df$Data_Value_BPHIGH)] <- 0
```

```
(moran_results <- moran.test(resid_w_impute, W_listw))
```

```
##
##  Moran I test under randomisation
##
## data:  resid_w_impute
## weights: W_listw
```

```
##
## Moran I statistic standard deviate = 232.23, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      4.857587e-01     -1.378626e-05      4.375356e-06
```

The p-value is negligible, so we can reject the null hypothesis of zero spatial autocorrelation. Since the observed value of the Moran I statistic is significantly greater then the expected value, the residuals are positively autocorrelated, in contrast to negatively autocorrelated. Thus, using a CAR model is justified.

```r
row.names(beta_results) <- names_non_spat_strat
```

```r
beta_inference <- cbind(beta_results[, 1],
                        beta_results[, 1] - 2 * beta_results[, 2],
                        beta_results[, 1] + 2 * beta_results[, 2])

colnames(beta_inference) <- c("pt_est", "lb", "ub")

beta_inference <- round(beta_inference, 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
```

```
##                              pt_est       lb       ub
## strat0:Intercept           30.73013 30.64740 30.81285
## strat0:flood_risk_pc1       0.70636  0.64096  0.77175
## strat0:flood_risk_pc2       0.30457  0.22762  0.38151
## strat0:flood_risk_pc3       0.08334  0.02089  0.14578
## strat0:flood_risk_pc4      -0.13820 -0.20375 -0.07265
## strat0:flood_risk_pc5       0.07678  0.01212  0.14145
## strat0:EP_UNINSUR          -0.42172 -0.53215 -0.31128
## strat0:pollute_conc_pc1    -1.59672 -1.68161 -1.51183
## strat0:pollute_conc_pc2    -1.17673 -1.25226 -1.10119
## strat0:pollute_conc_pc3     0.12649  0.05105  0.20193
## strat0:tmmx                 1.49213  1.41908  1.56519
## strat0:rmax                 0.71100  0.63720  0.78481
## strat0:Data_Value_CSMOKING  2.45027  2.35716  2.54339
## strat1:Intercept           33.13258 33.06000 33.20517
## strat1:flood_risk_pc1      -0.01265 -0.07750  0.05220
## strat1:flood_risk_pc2       0.07606  0.00864  0.14347
## strat1:flood_risk_pc3       0.13166  0.06767  0.19566
## strat1:flood_risk_pc4       0.03108 -0.02639  0.08855
## strat1:flood_risk_pc5      -0.00335 -0.06166  0.05496
## strat1:EP_UNINSUR          -0.85655 -0.91912 -0.79399
## strat1:pollute_conc_pc1    -0.62788 -0.69259 -0.56317
## strat1:pollute_conc_pc2    -1.54546 -1.62005 -1.47086
## strat1:pollute_conc_pc3    -0.66607 -0.73087 -0.60127
```

```
## strat1:tmmx                  2.36678  2.29213  2.44143
## strat1:rmax                  0.79117  0.72453  0.85782
## strat1:Data_Value_CSMOKING  3.88775  3.81533  3.96018
```

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:tmmx"                "strat0:rmax"
## [13] "strat0:Data_Value_CSMOKING" "strat1:Intercept"
## [15] "strat1:flood_risk_pc2"      "strat1:flood_risk_pc3"
## [17] "strat1:EP_UNINSUR"          "strat1:pollute_conc_pc1"
## [19] "strat1:pollute_conc_pc2"    "strat1:pollute_conc_pc3"
## [21] "strat1:tmmx"                "strat1:rmax"
## [23] "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$var_name <- substring(beta_inference_df$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",
```

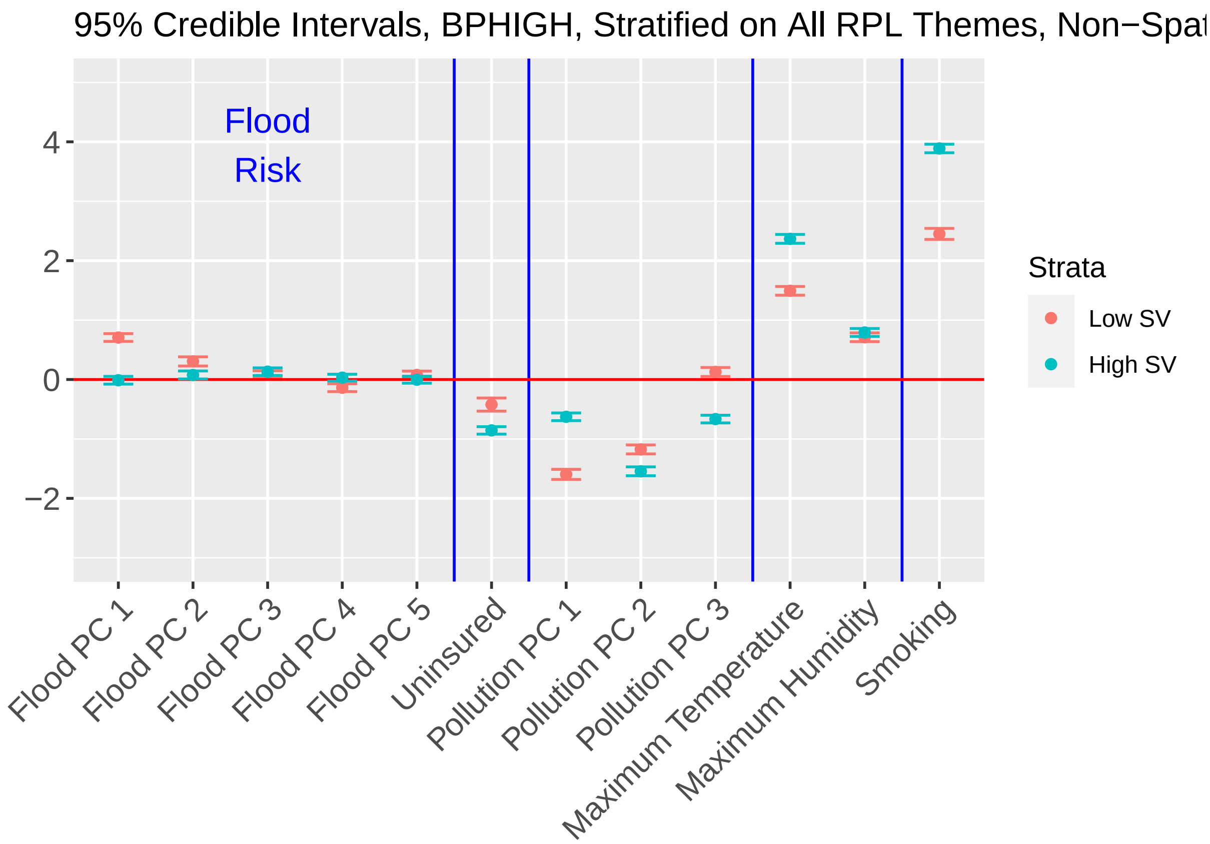
Splitting up the beta coefficients for each strata

```
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 = pt_est, 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(), axis
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 25)) +
  geom_errorbar(aes(ymin = lb, ymax = ub, 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 = lb, ymax = ub, width = 0.4), col = "#
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)

p
```

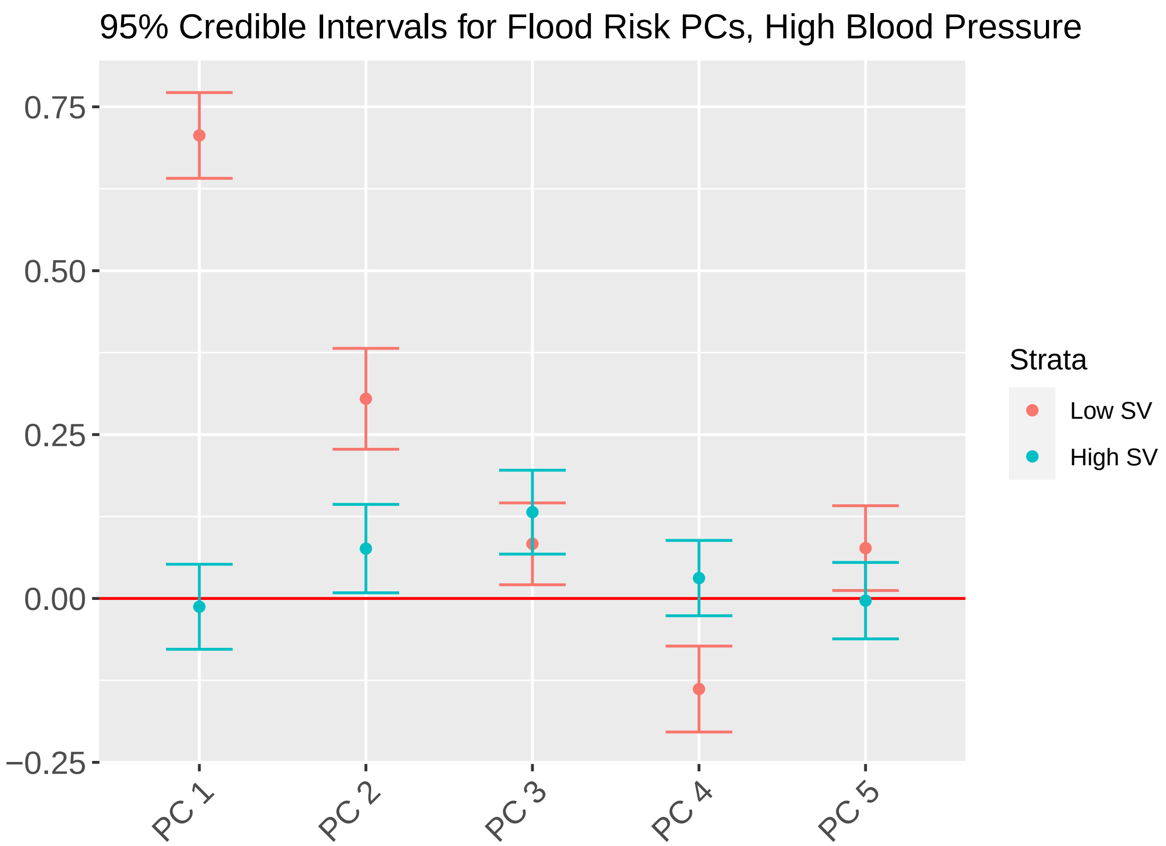



```
ggsave(here("figures/final_figures/sensitivity_analysis/non_spatial/BPHIGH_CI_rpls.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")
```

```
pc_extract_idx <- 2:6

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

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

```
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)
p
```




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

CAR model results, Asthma Stratified on RPL_THEMES

```
casthma_res <- non_spatial_strat_model(covariates_CASTHMA, first_var, strat_covariate, strat_fn = median

beta_results <- summary(casthma_res$lm_obj)$coefficients
```

Moran's I test for spatial autocorrelation

```
# mean imputing the residuals corresponding to missing Y values
resid_w_impute <- rep(NA, nrow(fhs_model_df))

resid_w_impute[!is.na(fhs_model_df$Data_Value_CASTHMA)] <- casthma_res$lm_obj$residuals

resid_w_impute[is.na(fhs_model_df$Data_Value_CASTHMA)] <- 0
```

```
(moran_results <- moran.test(resid_w_impute, W_listw))
```

```
##
##  Moran I test under randomisation
##
## data:  resid_w_impute
## weights: W_listw
##
## Moran I statistic standard deviate = 342.44, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      7.162850e-01     -1.378626e-05      4.375428e-06
```

The p-value is negligible, so we can reject the null hypothesis of zero spatial autocorrelation. Since the observed value of the Moran I statistic is significantly greater then the expected value, the residuals are positively autocorrelated, in contrast to negatively autocorrelated. Thus, using a CAR model is justified.

```
row.names(beta_results) <- names_non_spat_strat
```

```
beta_inference <- cbind(beta_results[, 1],
                        beta_results[, 1] - 2 * beta_results[, 2],
                        beta_results[, 1] + 2 * beta_results[, 2])

colnames(beta_inference) <- c("pt_est", "lb", "ub")

beta_inference <- round(beta_inference, 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
```

```
##                               pt_est       lb       ub
## strat0:Intercept             9.51735  9.50353  9.53117
## strat0:flood_risk_pc1       -0.04812 -0.05905 -0.03720
## strat0:flood_risk_pc2        0.04318  0.03033  0.05603
## strat0:flood_risk_pc3       -0.03481 -0.04524 -0.02438
## strat0:flood_risk_pc4       -0.00907 -0.02001  0.00188
## strat0:flood_risk_pc5       -0.02076 -0.03156 -0.00996
## strat0:EP_UNINSUR           -0.03892 -0.05737 -0.02048
## strat0:pollute_conc_pc1     -0.09957 -0.11375 -0.08540
## strat0:pollute_conc_pc2     -0.03182 -0.04444 -0.01920
## strat0:pollute_conc_pc3     -0.20417 -0.21677 -0.19157
## strat0:tmmx                 -0.18704 -0.19924 -0.17484
## strat0:rmax                 -0.27993 -0.29226 -0.26760
## strat0:Data_Value_CSMOKING   0.85935  0.84380  0.87491
## strat1:Intercept            10.07571 10.06359 10.08784
## strat1:flood_risk_pc1       -0.08182 -0.09265 -0.07098
## strat1:flood_risk_pc2       -0.07783 -0.08909 -0.06657
## strat1:flood_risk_pc3       -0.01080 -0.02149 -0.00011
## strat1:flood_risk_pc4        0.05325  0.04365  0.06285
```

```
## strat1:flood_risk_pc5       0.01051  0.00078  0.02025
## strat1:EP_UNINSUR          -0.20966 -0.22011 -0.19921
## strat1:pollute_conc_pc1     0.07572  0.06492  0.08653
## strat1:pollute_conc_pc2    -0.00755 -0.02001  0.00491
## strat1:pollute_conc_pc3    -0.18350 -0.19432 -0.17267
## strat1:tmmx                -0.17869 -0.19116 -0.16622
## strat1:rmax                -0.25626 -0.26739 -0.24512
## strat1:Data_Value_CSMOKING  1.33353  1.32143  1.34563
```

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:EP_UNINSUR"
##  [7] "strat0:pollute_conc_pc1"    "strat0:pollute_conc_pc2"
##  [9] "strat0:pollute_conc_pc3"    "strat0:tmmx"
## [11] "strat0:rmax"                "strat0:Data_Value_CSMOKING"
## [13] "strat1:Intercept"           "strat1:flood_risk_pc1"
## [15] "strat1:flood_risk_pc2"      "strat1:flood_risk_pc3"
## [17] "strat1:flood_risk_pc4"      "strat1:flood_risk_pc5"
## [19] "strat1:EP_UNINSUR"          "strat1:pollute_conc_pc1"
## [21] "strat1:pollute_conc_pc3"    "strat1:tmmx"
## [23] "strat1:rmax"                "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$var_name <- substring(beta_inference_df$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",
```

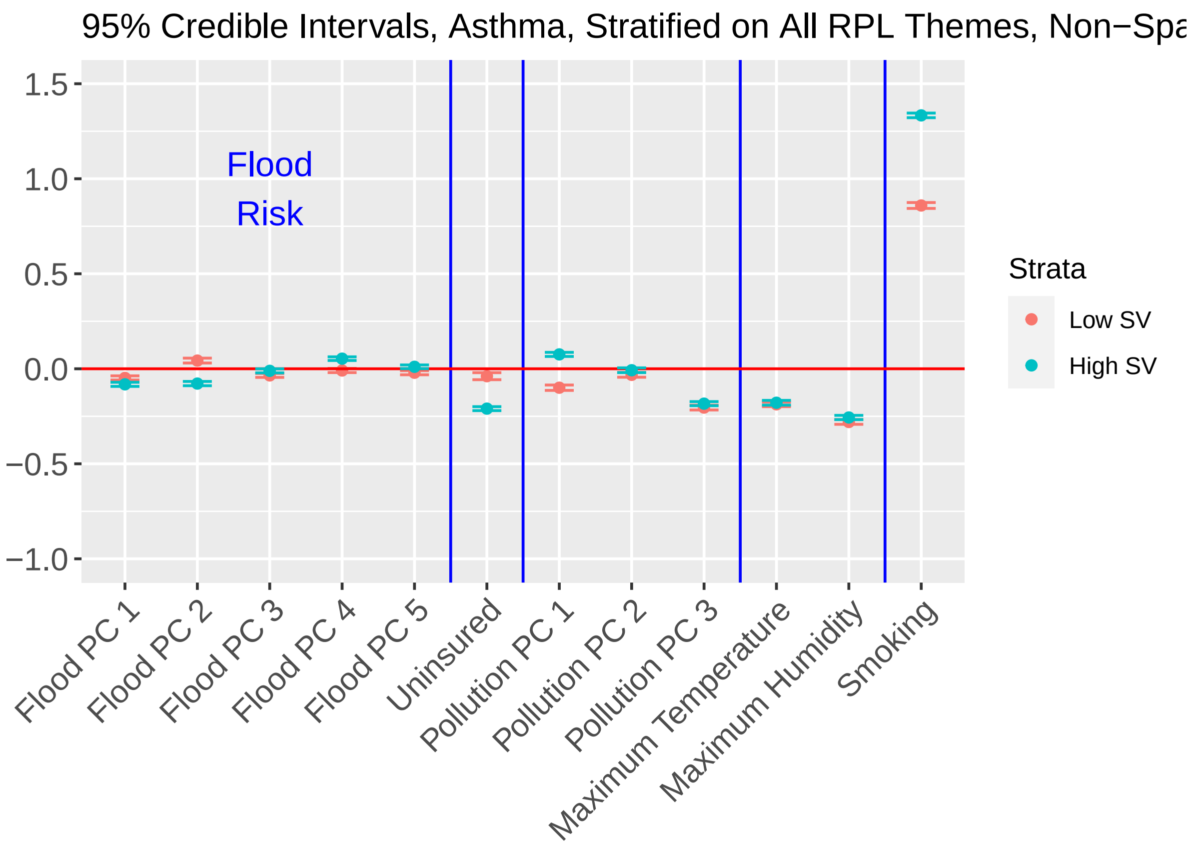
Splitting up the beta coefficients for each strata

```
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 = pt_est, 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(), axi
        axis.text=element_text(size=12),
        plot.margin = margin(5.5, 5.5, 5.5, 25)) +
  geom_errorbar(aes(ymin = lb, ymax = ub, 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",
```

```r
            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 R
  geom_point(data = beta_inference_df_strat1[-1, ], col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_inference_df_strat1[-1, ], aes(ymin = lb, ymax = ub, width = 0.4), col = "#0
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)

p
```

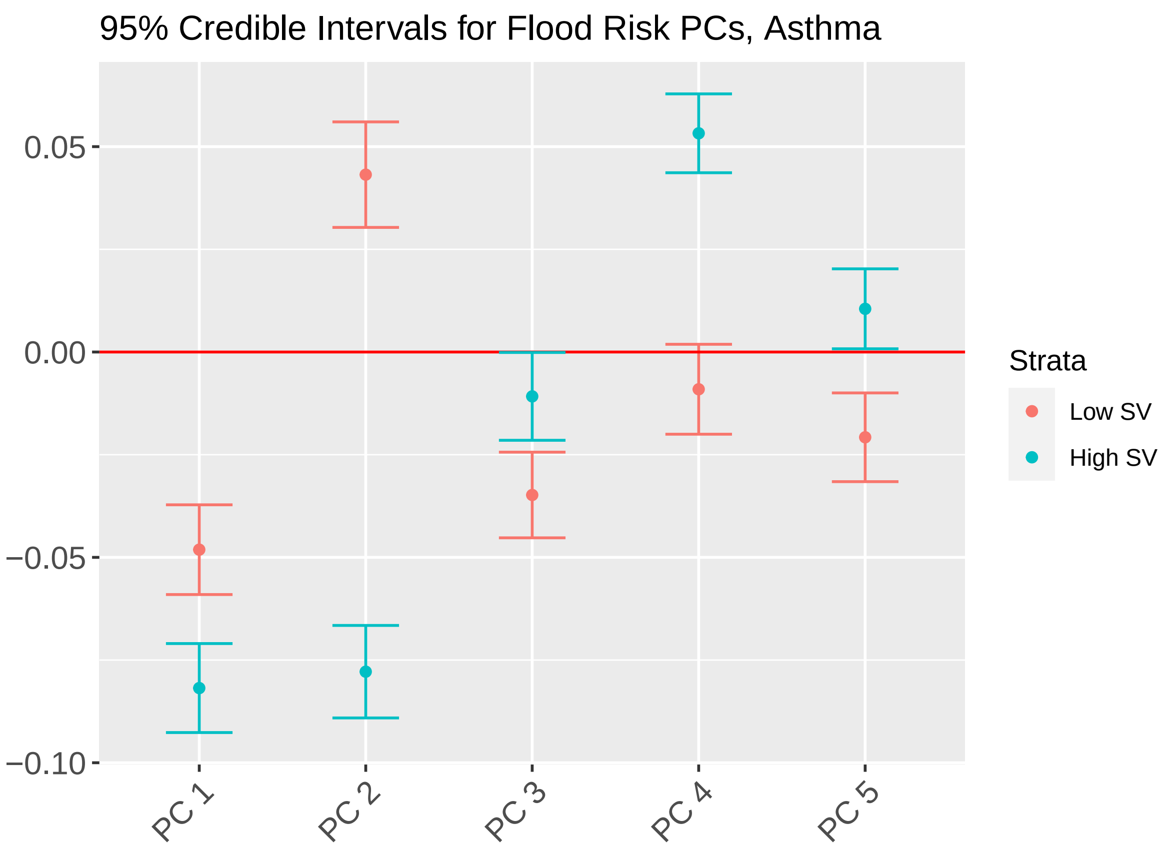



```r
ggsave(here("figures/final_figures/sensitivity_analysis/non_spatial/CASTHMA_CI_rpls.pdf"),
       plot = p, device = "pdf",
       width = 8, height = 6, units = "in")
```

```r
pc_extract_idx <- 2:6

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

```r
p <- ggplot(beta_CASTHMA_pcs$beta_pcs_strat0, aes(x = var_idx, y = pt_est, color = strat)) +
  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 = lb, ymax = ub, 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 = lb, ymax = ub, width = 0.4), col =
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)
p
```




```
ggsave(here("figures/final_figures/sensitivity_analysis/non_spatial/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

```
mhlth_res <- non_spatial_strat_model(covariates_MHLTH, first_var, strat_covariate, strat_fn = median)

beta_results <- summary(mhlth_res$lm_obj)$coefficients
```

Moran's I test for spatial autocorrelation

```
# mean imputing the residuals corresponding to missing Y values
resid_w_impute <- rep(NA, nrow(fhs_model_df))
```

```r
resid_w_impute[!is.na(fhs_model_df$Data_Value_MHLTH)] <- mhlth_res$lm_obj$residuals

resid_w_impute[is.na(fhs_model_df$Data_Value_MHLTH)] <- 0
```

```r
(moran_results <- moran.test(resid_w_impute, W_listw))
```

```
##
##  Moran I test under randomisation
##
## data:  resid_w_impute
## weights: W_listw
##
## Moran I statistic standard deviate = 268.25, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      5.610329e-01     -1.378626e-05      4.374529e-06
```

The p-value is negligible, so we can reject the null hypothesis of zero spatial autocorrelation. Since the observed value of the Moran I statistic is significantly greater then the expected value, the residuals are positively autocorrelated, in contrast to negatively autocorrelated. Thus, using a CAR model is justified.

```r
row.names(beta_results) <- names_non_spat_strat
```

```r
beta_inference <- cbind(beta_results[, 1],
                        beta_results[, 1] - 2 * beta_results[, 2],
                        beta_results[, 1] + 2 * beta_results[, 2])

colnames(beta_inference) <- c("pt_est", "lb", "ub")

beta_inference <- round(beta_inference, 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
```

```
##                             pt_est       lb       ub
## strat0:Intercept          13.82648 13.80507 13.84789
## strat0:flood_risk_pc1     -0.05243 -0.06936 -0.03551
## strat0:flood_risk_pc2     -0.03904 -0.05895 -0.01912
## strat0:flood_risk_pc3     -0.03741 -0.05358 -0.02125
## strat0:flood_risk_pc4      0.00273 -0.01424  0.01969
## strat0:flood_risk_pc5     -0.02932 -0.04605 -0.01258
## strat0:EP_UNINSUR          0.26889  0.24031  0.29748
## strat0:pollute_conc_pc1    0.16126  0.13928  0.18323
## strat0:pollute_conc_pc2   -0.07410 -0.09365 -0.05455
## strat0:pollute_conc_pc3   -0.19632 -0.21585 -0.17679
## strat0:tmmx                0.34283  0.32392  0.36174
## strat0:rmax               -0.08611 -0.10521 -0.06701
```

```
## strat0:Data_Value_CSMOKING  2.65060  2.62650  2.67470
## strat1:Intercept           14.56535 14.54656 14.58414
## strat1:flood_risk_pc1      -0.00217 -0.01896  0.01461
## strat1:flood_risk_pc2      -0.08990 -0.10735 -0.07245
## strat1:flood_risk_pc3       0.01672  0.00015  0.03328
## strat1:flood_risk_pc4       0.03849  0.02361  0.05337
## strat1:flood_risk_pc5      -0.00266 -0.01775  0.01243
## strat1:EP_UNINSUR           0.07523  0.05903  0.09142
## strat1:pollute_conc_pc1     0.42166  0.40491  0.43841
## strat1:pollute_conc_pc2     0.17542  0.15611  0.19473
## strat1:pollute_conc_pc3     0.00992 -0.00685  0.02669
## strat1:tmmx                 0.39273  0.37340  0.41205
## strat1:rmax                 0.01504 -0.00221  0.03229
## strat1:Data_Value_CSMOKING  2.94687  2.92812  2.96561
```

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:EP_UNINSUR"
##  [7] "strat0:pollute_conc_pc1"    "strat0:pollute_conc_pc2"
##  [9] "strat0:pollute_conc_pc3"    "strat0:tmmx"
## [11] "strat0:rmax"                "strat0:Data_Value_CSMOKING"
## [13] "strat1:Intercept"           "strat1:flood_risk_pc2"
## [15] "strat1:flood_risk_pc3"      "strat1:flood_risk_pc4"
## [17] "strat1:EP_UNINSUR"          "strat1:pollute_conc_pc1"
## [19] "strat1:pollute_conc_pc2"    "strat1:tmmx"
## [21] "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$var_name <- substring(beta_inference_df$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",
```

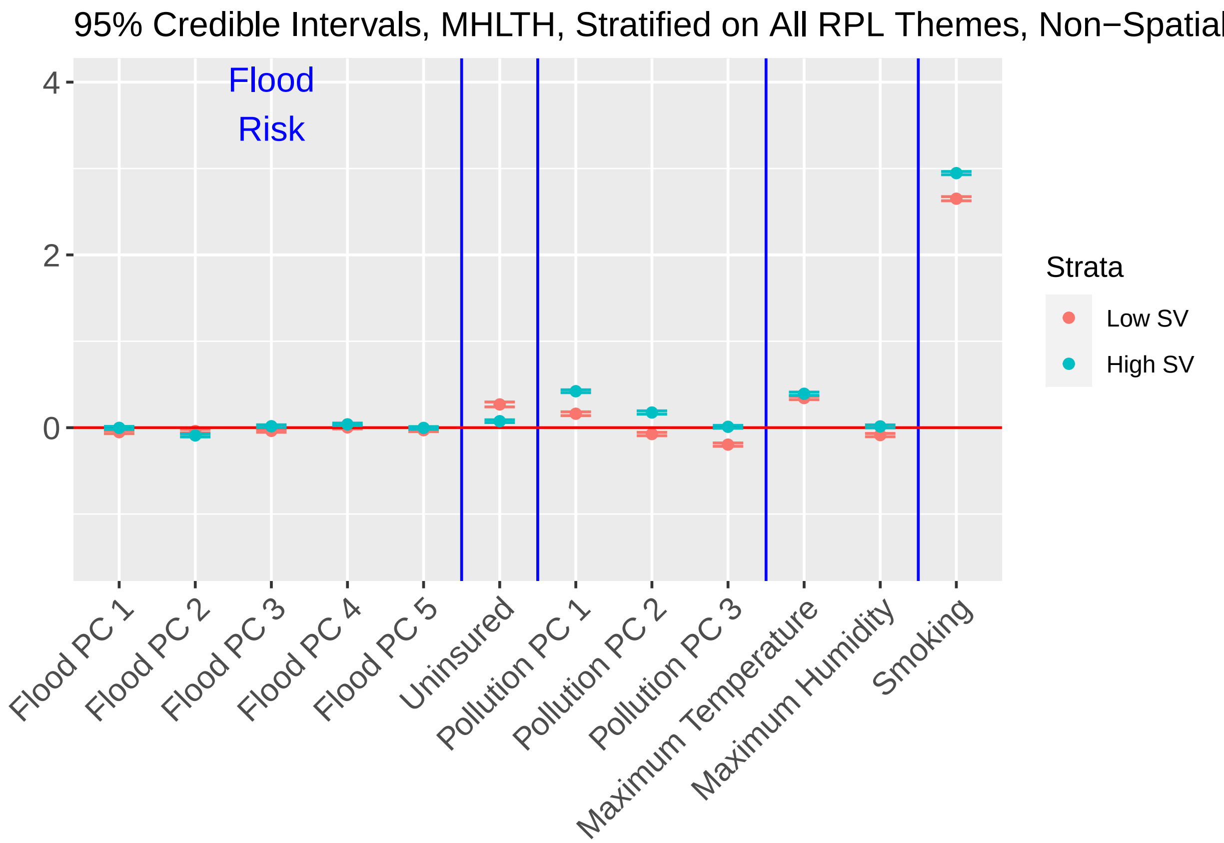
Splitting up the beta coefficients for each strata

```
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 = pt_est, 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(), axis
        axis.text=element_text(size=12),
```

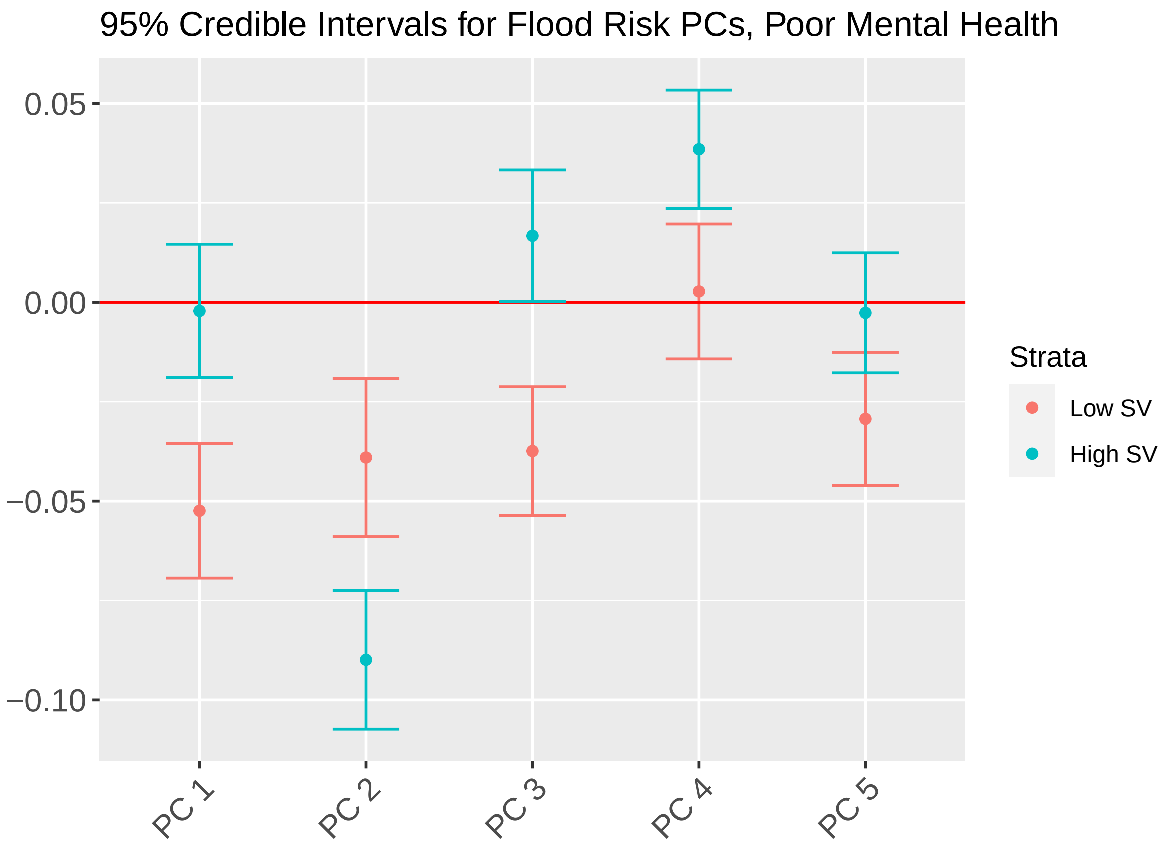
```
        plot.margin = margin(5.5, 5.5, 5.5, 25)) +
  geom_errorbar(aes(ymin = lb, ymax = ub, width = 0.4), col = "#F8766D") +
  geom_vline(xintercept = c(5.5, 6.5, 9.5, 11.5), col = "blue") +
  geom_hline(yintercept = 0, col = "red") +
  annotate(geom = "text", x = 3, y = 3.75, label = "Flood\nRisk",
           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 RP
  geom_point(data = beta_inference_df_strat1[-1, ], col = "#00BFC4") + # strat 1
  geom_errorbar(data = beta_inference_df_strat1[-1, ], aes(ymin = lb, ymax = ub, width = 0.4), col = "#
  scale_color_manual(name = "Strata",
                     values = c("#F8766D", "#00BFC4"),
                     drop = FALSE)

p
```




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

```
pc_extract_idx <- 2:6

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

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




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