Mitigation of Multicollinearity Analysis

Alvin Sheng

2022-07-24

```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(GGally)
```

```
## Loading required package: ggplot2
```

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

```r
library(usdm)
```

```
## Loading required package: sp
```

```
## Loading required package: raster
```

```r
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract() masks raster::extract()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x dplyr::select()  masks raster::select()
```

Checking for multicollinearity among the covariates

```r
fhs_model_df <- readRDS(here("intermediary_data/fhs_model_df_all_census_tract_reorg_prev.rds"))
```

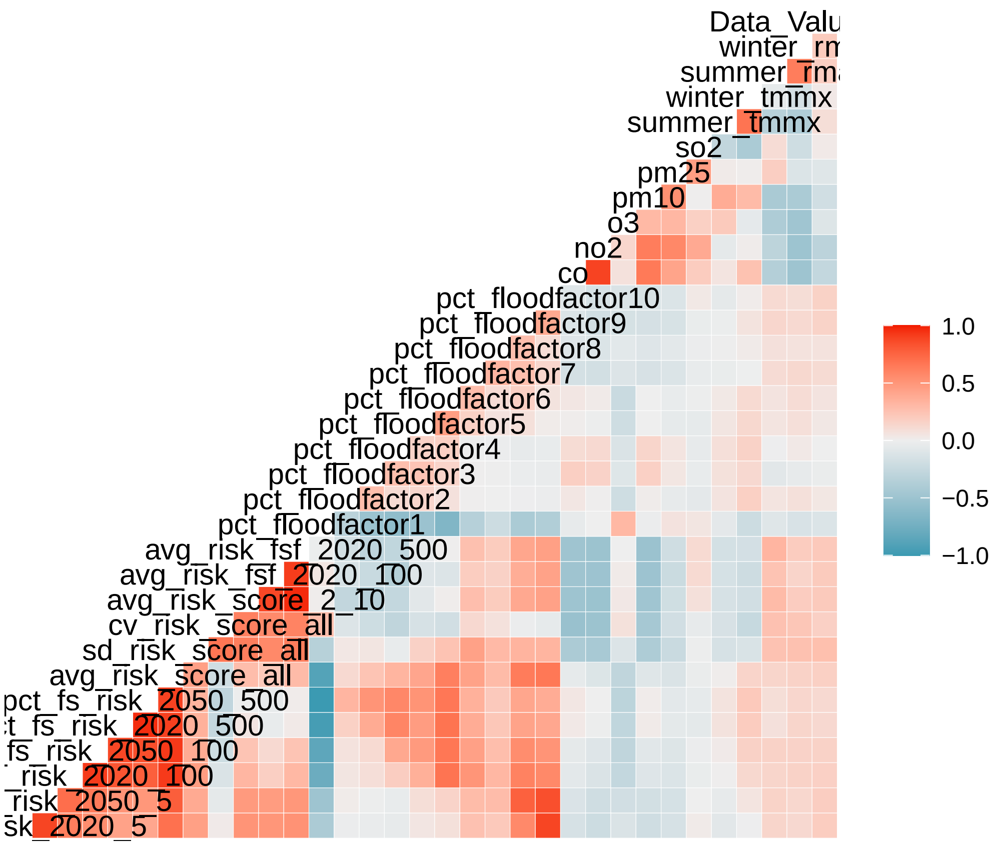
All climate-related variables: flood risk, pollution, and GRIDMET variables

Excluding variables in collin_var_names

```r
fr_index <- 19:40

climate_var_idx <- c(fr_index, 57:66)
```

```r
ggcorr(data = fhs_model_df[, c(climate_var_idx, ncol(fhs_model_df) - 3)])
```



Flood risk variables

```r
fr_index <- 19:40
```

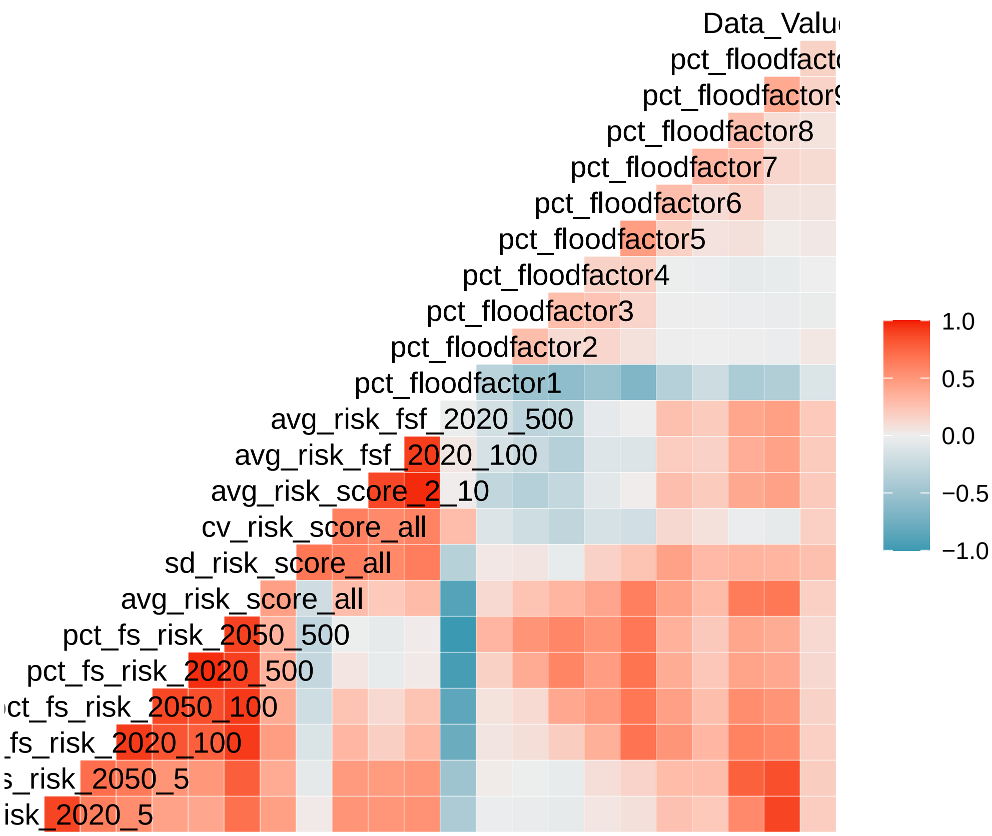
Variances of the flood risk variables

```r
apply(fhs_model_df[fr_index], 2, function(vec) var(vec, na.rm = T))
```

```
##     pct_fs_risk_2020_5     pct_fs_risk_2050_5   pct_fs_risk_2020_100
##           0.0057589256           0.0094357940           0.0233741334
##   pct_fs_risk_2050_100   pct_fs_risk_2020_500   pct_fs_risk_2050_500
##           0.0310844965           0.0429016485           0.0502528142
##     avg_risk_score_all      sd_risk_score_all      cv_risk_score_all
##           1.2951736984           0.5867333532           0.0995405360
##    avg_risk_score_2_10 avg_risk_fsf_2020_100 avg_risk_fsf_2020_500
##           1.8270343154           1.2240897449           1.7439802173
##       pct_floodfactor1       pct_floodfactor2       pct_floodfactor3
##           0.0502753313           0.0013176697           0.0037748277
```

```
##      pct_floodfactor4      pct_floodfactor5      pct_floodfactor6
##          0.0089191745          0.0009741984          0.0072042201
##      pct_floodfactor7      pct_floodfactor8      pct_floodfactor9
##          0.0006632554          0.0001194391          0.0021788527
##     pct_floodfactor10
##          0.0038558597
```

```
ggcorr(data = fhs_model_df[, c(fr_index, ncol(fhs_model_df) - 3)])
```



For each variable, I take the summary of its correlations with other variables, not including itself.

```
flood_cor <- cor(fhs_model_df[complete.cases(fhs_model_df[, c(fr_index, ncol(fhs_model_df))]), c(fr_inde

diag(flood_cor) <- NA

summary(flood_cor)
```

```
##  pct_fs_risk_2020_5 pct_fs_risk_2050_5 pct_fs_risk_2020_100
##  Min.   :-0.41258   Min.   :-0.5036    Min.   :-0.8052
##  1st Qu.: 0.03317   1st Qu.: 0.0418    1st Qu.: 0.1848
##  Median : 0.42470   Median : 0.4285    Median : 0.4121
##  Mean   : 0.32982   Mean   : 0.3539    Mean   : 0.3872
##  3rd Qu.: 0.54557   3rd Qu.: 0.6171    3rd Qu.: 0.6688
##  Max.   : 0.88390   Max.   : 0.8829    Max.   : 0.9373
##  NA's   :1          NA's   :1          NA's   :1
##  pct_fs_risk_2050_100 pct_fs_risk_2020_500 pct_fs_risk_2050_500
##  Min.   :-0.8672      Min.   :-0.9656      Min.   :-1.0000
##  1st Qu.: 0.1670      1st Qu.: 0.1012      1st Qu.: 0.1173
##  Median : 0.4207      Median : 0.4081      Median : 0.3988
```

```
## Mean   : 0.3762      Mean   : 0.3388      Mean   : 0.3366
## 3rd Qu.: 0.6291      3rd Qu.: 0.5681      3rd Qu.: 0.5650
## Max.   : 0.9373      Max.   : 0.9656      Max.   : 0.9656
## NA's   :1            NA's   :1            NA's   :1
## avg_risk_score_all sd_risk_score_all cv_risk_score_all  avg_risk_score_2_10
## Min.   :-0.9013    Min.   :-0.3256   Min.   :-0.45146   Min.   :-0.36560
## 1st Qu.: 0.2599    1st Qu.: 0.1610   1st Qu.:-0.31476   1st Qu.:-0.01594
## Median : 0.4296    Median : 0.3352   Median :-0.11407   Median : 0.25807
## Mean   : 0.4103    Mean   : 0.3021   Mean   :-0.03583   Mean   : 0.23244
## 3rd Qu.: 0.6805    3rd Qu.: 0.4341   3rd Qu.: 0.04764   3rd Qu.: 0.47516
## Max.   : 0.9332    Max.   : 0.6054   Max.   : 0.57772   Max.   : 0.96516
## NA's   :1          NA's   :1         NA's   :1          NA's   :1
## avg_risk_fsf_2020_100 avg_risk_fsf_2020_500 pct_floodfactor1
## Min.   :-0.35635      Min.   :-0.308819     Min.   :-1.0000
## 1st Qu.:-0.05283      1st Qu.:-0.003222     1st Qu.:-0.6554
## Median : 0.18534      Median : 0.255205     Median :-0.4204
## Mean   : 0.20935      Mean   : 0.243471     Mean   :-0.4274
## 3rd Qu.: 0.45731      3rd Qu.: 0.489202     3rd Qu.:-0.2847
## Max.   : 0.91241      Max.   : 0.965161     Max.   : 0.4514
## NA's   :1             NA's   :1             NA's   :1
## pct_floodfactor2   pct_floodfactor3   pct_floodfactor4   pct_floodfactor5
## Min.   :-0.33290   Min.   :-0.52073   Min.   :-0.57504   Min.   :-0.53499
## 1st Qu.:-0.01982   1st Qu.:-0.03180   1st Qu.:-0.04915   1st Qu.: 0.02091
## Median : 0.03029   Median : 0.01718   Median :-0.01931   Median : 0.11932
## Mean   : 0.02048   Mean   : 0.03583   Mean   : 0.03601   Mean   : 0.13565
## 3rd Qu.: 0.12165   3rd Qu.: 0.23844   3rd Qu.: 0.21566   3rd Qu.: 0.34498
## Max.   : 0.33275   Max.   : 0.52045   Max.   : 0.58485   Max.   : 0.53506
## NA's   :1          NA's   :1          NA's   :1          NA's   :1
## pct_floodfactor6   pct_floodfactor7  pct_floodfactor8   pct_floodfactor9
## Min.   :-0.68214   Min.   :-0.3593   Min.   :-0.27107   Min.   :-0.42826
## 1st Qu.: 0.04623   1st Qu.: 0.0763   1st Qu.: 0.04401   1st Qu.: 0.01582
## Median : 0.16419   Median : 0.2623   Median : 0.25821   Median : 0.38830
## Mean   : 0.19445   Mean   : 0.2218   Mean   : 0.19782   Mean   : 0.29026
## 3rd Qu.: 0.41876   3rd Qu.: 0.3772   3rd Qu.: 0.34150   3rd Qu.: 0.44108
## Max.   : 0.69827   Max.   : 0.5021   Max.   : 0.46059   Max.   : 0.78917
## NA's   :1          NA's   :1         NA's   :1          NA's   :1
## pct_floodfactor10  Data_Value_MHLTH
## Min.   :-0.38505   Min.   :-0.066084
## 1st Qu.: 0.02112   1st Qu.: 0.001076
## Median : 0.36135   Median : 0.030605
## Mean   : 0.28196   Mean   : 0.027229
## 3rd Qu.: 0.45601   3rd Qu.: 0.053374
## Max.   : 0.88390   Max.   : 0.088977
## NA's   :1          NA's   :1
```

Many of the flood risk variables are very correlated.

PCA with Centering AND Scaling

I think scaling all covariates twice, before and after PCA, will lead to more interpretable results

Conduct PCA on the correlated flood risk variables

```r
first_var <- 19
```

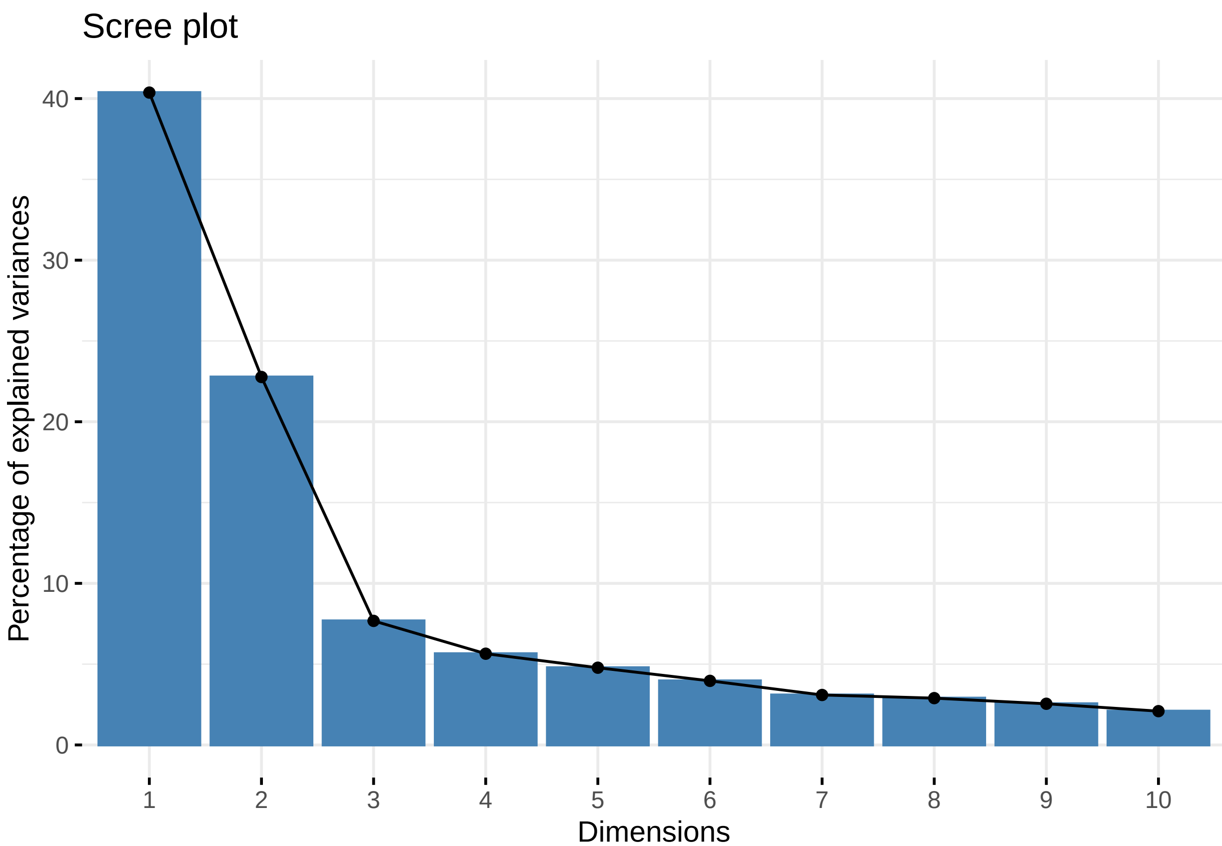
```r
fr_index <- first_var:(first_var + 21)

flood_risk <- fhs_model_df[, fr_index]
```

```r
fr_pca <- prcomp(flood_risk[complete.cases(flood_risk),], center = T, scale. = T)

fr_loadings <- fr_pca$rotation
```

```r
fviz_eig(fr_pca)
```




```r
summ_pca <- summary(fr_pca)

summ_pca$importance[,1:10]
```

```
##                             PC1      PC2      PC3      PC4      PC5       PC6
## Standard deviation     2.980165 2.238176 1.299683 1.114667 1.025496 0.9341738
## Proportion of Variance 0.403700 0.227700 0.076780 0.056480 0.047800 0.0396700
## Cumulative Proportion  0.403700 0.631400 0.708180 0.764660 0.812460 0.8521300
##                              PC7       PC8      PC9      PC10
## Standard deviation     0.8249525 0.7983534 0.748906 0.6784936
## Proportion of Variance 0.0309300 0.0289700 0.025490 0.0209300
## Cumulative Proportion  0.8830600 0.9120300 0.937530 0.9584500
```

We started out with 22 variables. Including five PC scores would include >80% of the variance. Perhaps I can also look at the top 8 PCs, to get >90% variance explained.

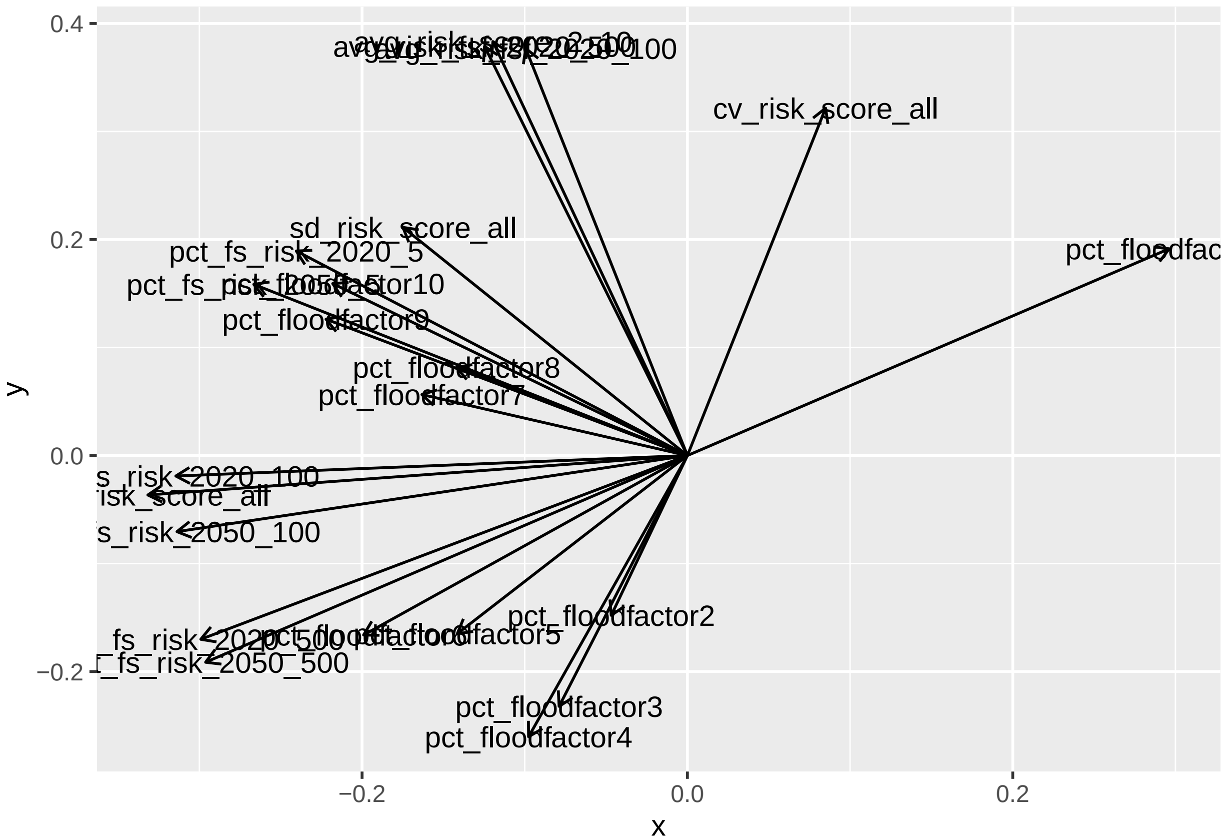
Printing out the loadings, from most negative to least

```r
round(fr_loadings[, 1:8], digits = 2)
```

```
##                           PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
## pct_fs_risk_2020_5      -0.24  0.19 -0.31  0.12 -0.03  0.00 -0.05  0.23
## pct_fs_risk_2050_5      -0.27  0.16 -0.33  0.04 -0.12  0.06 -0.05 -0.02
## pct_fs_risk_2020_100    -0.31 -0.02  0.00 -0.18  0.03  0.05  0.12  0.08
## pct_fs_risk_2050_100    -0.31 -0.07  0.02 -0.15  0.11  0.00  0.10 -0.04
## pct_fs_risk_2020_500    -0.30 -0.17  0.06  0.01  0.09 -0.14  0.02 -0.01
## pct_fs_risk_2050_500    -0.30 -0.19  0.07  0.12  0.03 -0.05  0.00 -0.04
## avg_risk_score_all      -0.33 -0.04 -0.06 -0.02  0.03  0.02  0.00  0.02
## sd_risk_score_all       -0.17  0.21  0.38  0.25  0.02 -0.09  0.02  0.17
## cv_risk_score_all        0.08  0.32  0.38  0.20  0.06 -0.10  0.05  0.02
## avg_risk_score_2_10     -0.12  0.38  0.11 -0.01  0.19 -0.06  0.02 -0.13
## avg_risk_fsf_2020_100 -0.10  0.38  0.09  0.16  0.08  0.04 -0.07 -0.15
## avg_risk_fsf_2020_500 -0.12  0.38  0.12  0.08  0.14  0.01  0.03 -0.17
## pct_floodfactor1         0.30  0.19 -0.07 -0.12 -0.03  0.05  0.00  0.04
## pct_floodfactor2        -0.05 -0.15  0.07  0.51 -0.41  0.35  0.58 -0.13
## pct_floodfactor3        -0.08 -0.23  0.10  0.47 -0.15 -0.06 -0.62  0.11
## pct_floodfactor4        -0.10 -0.26  0.02  0.14  0.21 -0.68  0.19 -0.21
## pct_floodfactor5        -0.14 -0.16  0.25 -0.02  0.19  0.48 -0.37 -0.14
## pct_floodfactor6        -0.20 -0.17  0.26 -0.26  0.26  0.28  0.17  0.01
## pct_floodfactor7        -0.16  0.06  0.31 -0.28 -0.37 -0.10  0.08  0.55
## pct_floodfactor8        -0.14  0.08  0.15 -0.28 -0.60 -0.19 -0.19 -0.25
## pct_floodfactor9        -0.22  0.13 -0.17 -0.11 -0.22  0.04 -0.08 -0.51
## pct_floodfactor10       -0.22  0.16 -0.40  0.16  0.10  0.04  0.00  0.35
```

```r
# Extract loadings of the variables
fr_loadings_df <- data.frame(Variables = rownames(fr_pca$rotation), fr_pca$rotation)

    # Plot
ggplot(fr_loadings_df) +
  geom_segment(data = fr_loadings_df, aes(x = 0, y = 0, xend = PC1,
     yend = PC2), arrow = arrow(length = unit(1/2, "picas")),
     color = "black") +
  annotate("text", x = (fr_loadings_df$PC1), y = (fr_loadings_df$PC2),
     label = fr_loadings_df$Variables)
```



Data pre-processing decision: Use first 5 flood risk PCs.

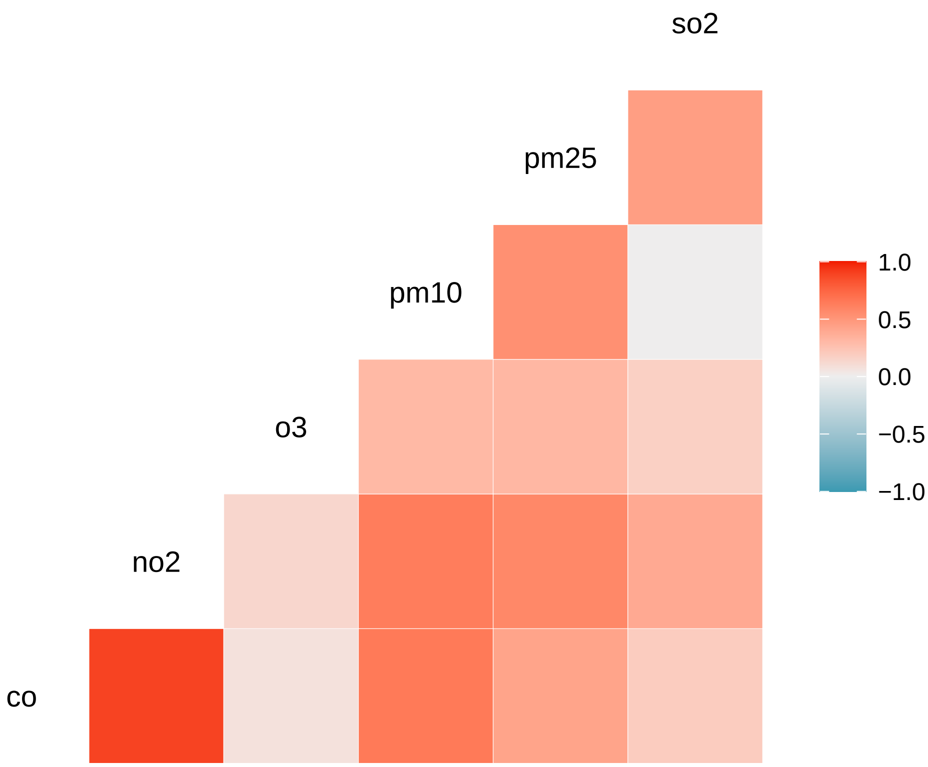
Pollution Variables

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

```
pollute_index <- 57:62

pollute_var <- fhs_model_df[, pollute_index]
```

```
ggcorr(data = pollute_var)
```




```r
cor(pollute_var[complete.cases(pollute_var),])
```

```
##              co       no2         o3        pm10      pm25         so2
## co   1.00000000 0.8915705 0.07482752 0.650197673 0.4244457 0.195405604
## no2  0.89157049 1.0000000 0.13748067 0.632891996 0.5768223 0.392550909
## o3   0.07482752 0.1374807 1.00000000 0.305919820 0.3164809 0.173882481
## pm10 0.65019767 0.6328920 0.30591982 1.000000000 0.5337475 0.003072452
## pm25 0.42444574 0.5768223 0.31648094 0.533747469 1.0000000 0.457689473
## so2  0.19540560 0.3925509 0.17388248 0.003072452 0.4576895 1.000000000
```

PCA

```r
pollute_index <- 57:62

pollute_var <- fhs_model_df[, pollute_index]
```

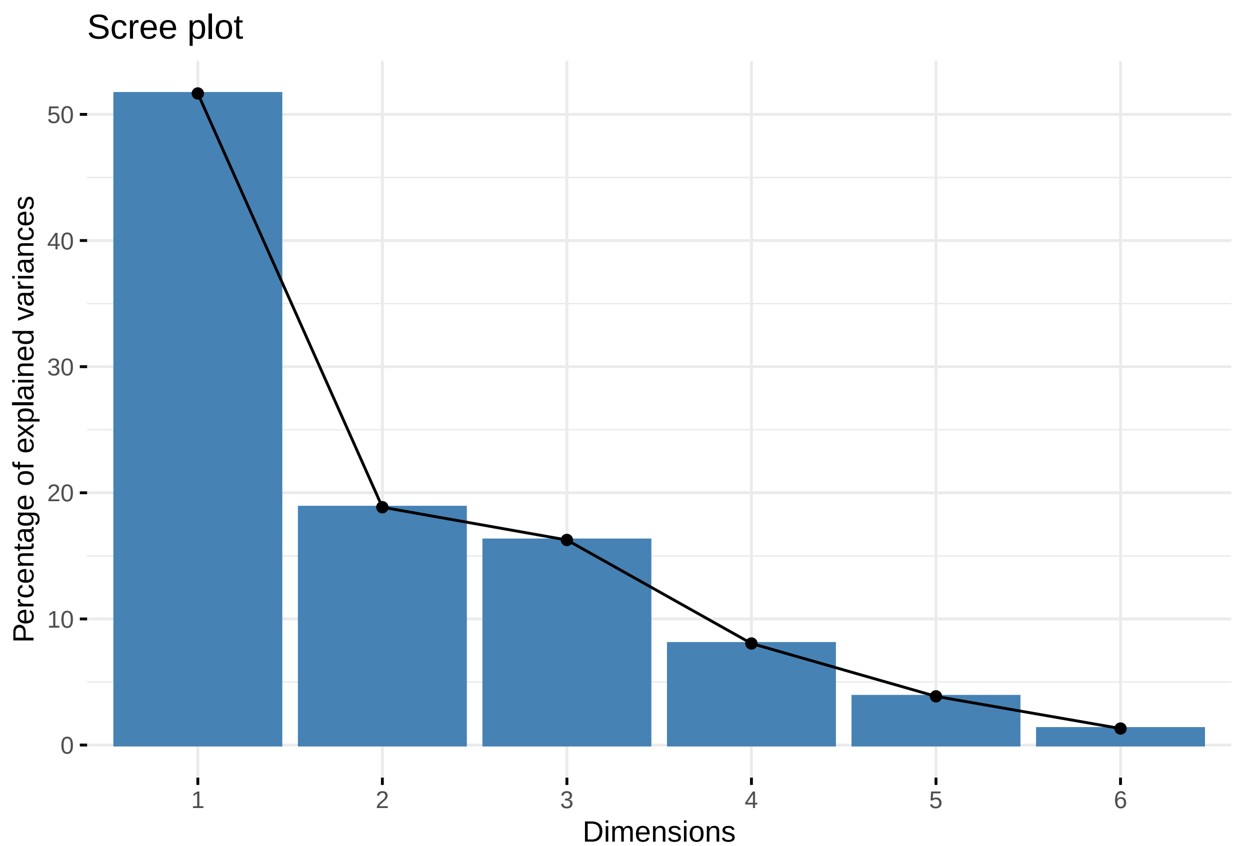
```r
var(pollute_var, na.rm = T)
```

```
##               co        no2          o3        pm10       pm25        so2
## co   0.008190043  0.4564204  0.03503498  0.31907222 0.08920073 0.01739680
## no2  0.456420387 31.9987131  4.02351171 19.41316530 7.57724647 2.18449761
## o3   0.035034978  4.0235117 26.76669605  8.58233557 3.80231481 0.88499852
## pm10 0.319072222 19.4131653  8.58233557 29.40363247 6.72108596 0.01638985
## pm25 0.089200729  7.5772465  3.80231481  6.72108596 5.39269996 1.04559472
## so2  0.017396803  2.1844976  0.88499852  0.01638985 1.04559472 0.96778449
```

```r
pollute_pca <- prcomp(pollute_var[complete.cases(pollute_var),], center = T, scale. = T)

pollute_loadings <- pollute_pca$rotation
```

```
fviz_eig(pollute_pca)
```



First 3 PCs should be sufficient.

```
summ_pca <- summary(pollute_pca)

summ_pca$importance
```

```
##                             PC1      PC2       PC3     PC4       PC5       PC6
## Standard deviation     1.760563 1.063661 0.9876988 0.69512 0.4813467 0.2803715
## Proportion of Variance 0.516600 0.188560 0.1625900 0.08053 0.0386200 0.0131000
## Cumulative Proportion  0.516600 0.705160 0.8677500 0.94828 0.9869000 1.0000000
```

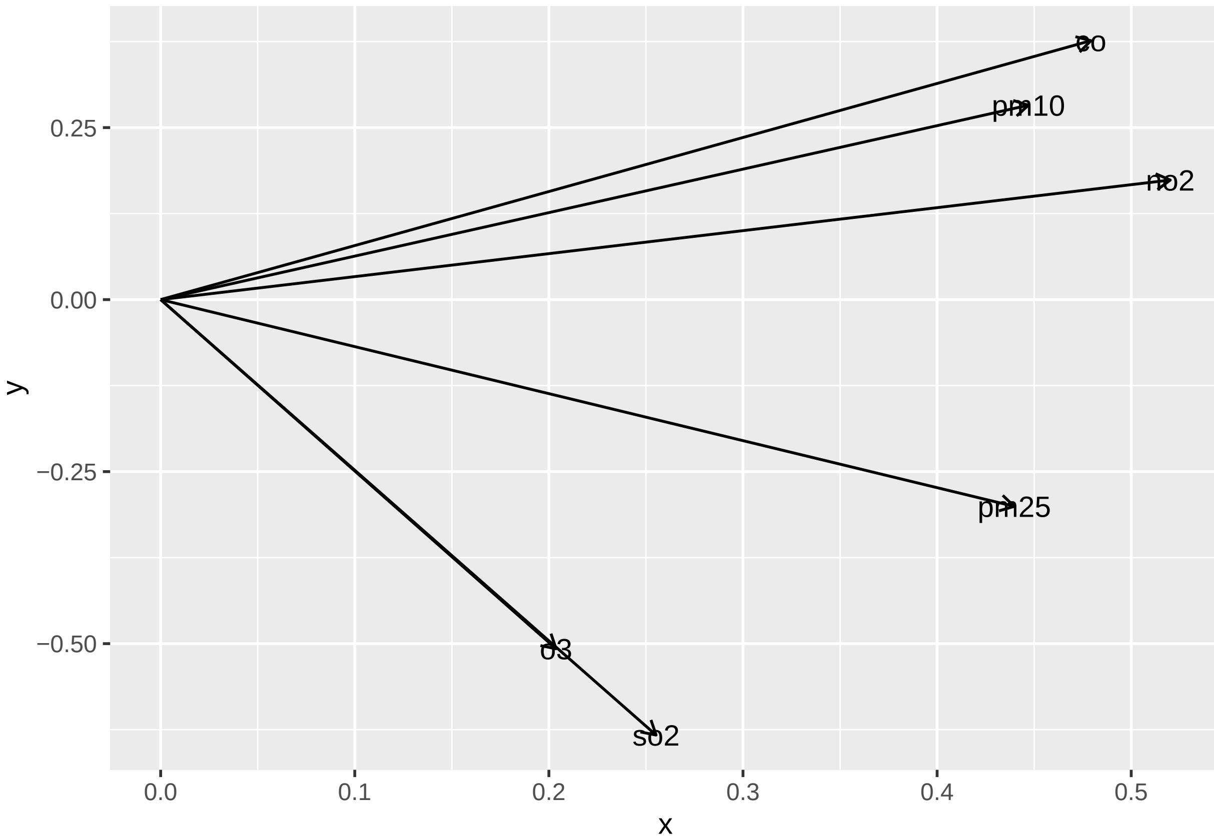
```
round(pollute_loadings, digits = 2)
```

```
##       PC1   PC2   PC3   PC4   PC5   PC6
## co   0.48  0.38 -0.14  0.36 -0.23  0.65
## no2  0.52  0.17 -0.21  0.25 -0.21 -0.74
## o3   0.20 -0.51  0.71  0.42 -0.16  0.00
## pm10 0.45  0.28  0.38 -0.25  0.72 -0.02
## pm25 0.44 -0.30  0.00 -0.73 -0.41  0.11
## so2  0.26 -0.63 -0.54  0.18  0.44  0.12
```

```
# Extract loadings of the variables
pollute_loadings_df <- data.frame(Variables = rownames(pollute_pca$rotation), pollute_pca$rotation)

    # Plot
ggplot(pollute_loadings_df) +
  geom_segment(data = pollute_loadings_df, aes(x = 0, y = 0, xend = PC1,
```

```r
    yend = PC2), arrow = arrow(length = unit(1/2, "picas")),
    color = "black") +
  annotate("text", x = (pollute_loadings_df$PC1), y = (pollute_loadings_df$PC2),
    label = pollute_loadings_df$Variables)
```



Data pre-processing decision: Use first 3 pollution PCs.

Meteorological Variables

```r
mean_df_GRIDMET <- readRDS(file = here("intermediary_data/mean_df_GRIDMET.rds"))

mean_df_GRIDMET$fips <- as.numeric(mean_df_GRIDMET$fips)

mean_df_GRIDMET <- mean_df_GRIDMET[!(mean_df_GRIDMET$fips %in% c(6075980401, 12087980100)), ]
```

```r
all.equal(fhs_model_df$fips, mean_df_GRIDMET$fips)
```

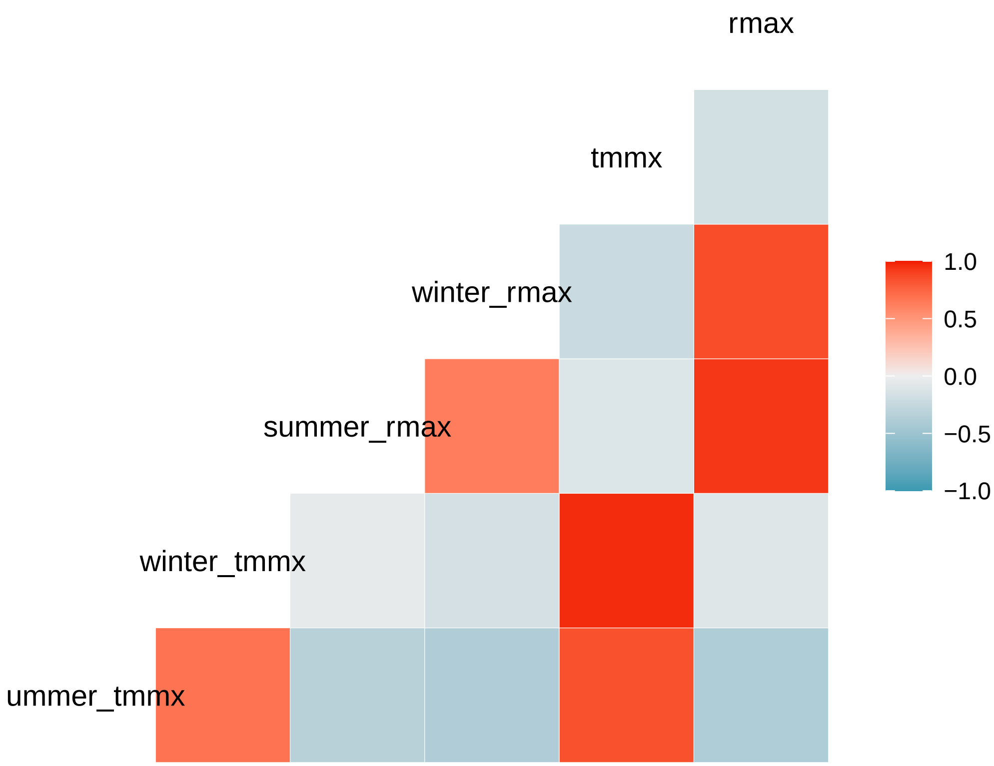
```
## [1] TRUE
```

```r
met_index <- 63:66

met_var <- fhs_model_df[, met_index]
```

```r
all_met_var <- cbind(met_var, mean_df_GRIDMET[, -1])

ggcorr(data = all_met_var)
```




```r
cor(all_met_var[complete.cases(all_met_var),])
```

```
##              summer_tmmx winter_tmmx summer_rmax winter_rmax       tmmx
## summer_tmmx   1.0000000  0.67627429 -0.33882533  -0.3849305  0.8436397
## winter_tmmx   0.6762743  1.00000000 -0.04948591  -0.1593851  0.9623741
## summer_rmax  -0.3388253 -0.04948591  1.00000000   0.6320484 -0.1096798
## winter_rmax  -0.3849305 -0.15938510  0.63204841   1.0000000 -0.2323956
## tmmx          0.8436397  0.96237411 -0.10967984  -0.2323956  1.0000000
## rmax         -0.3894133 -0.09691431  0.93132795   0.8556919 -0.1735431
##                    rmax
## summer_tmmx -0.38941326
## winter_tmmx -0.09691431
## summer_rmax  0.93132795
## winter_rmax  0.85569194
## tmmx        -0.17354309
## rmax         1.00000000
```

PCA

```r
met_index <- 63:66

met_var <- fhs_model_df[, met_index]
```

```r
var(met_var[complete.cases(met_var),])
```

```
##              summer_tmmx winter_tmmx summer_rmax winter_rmax
## summer_tmmx   11.302969   16.306899  -13.215165   -9.797234
## winter_tmmx   16.306899   51.440415   -4.117506   -8.654159
## summer_rmax  -13.215165   -4.117506  134.586313   55.510570
```

```
## winter_rmax   -9.797234   -8.654159   55.510570   57.312559
```

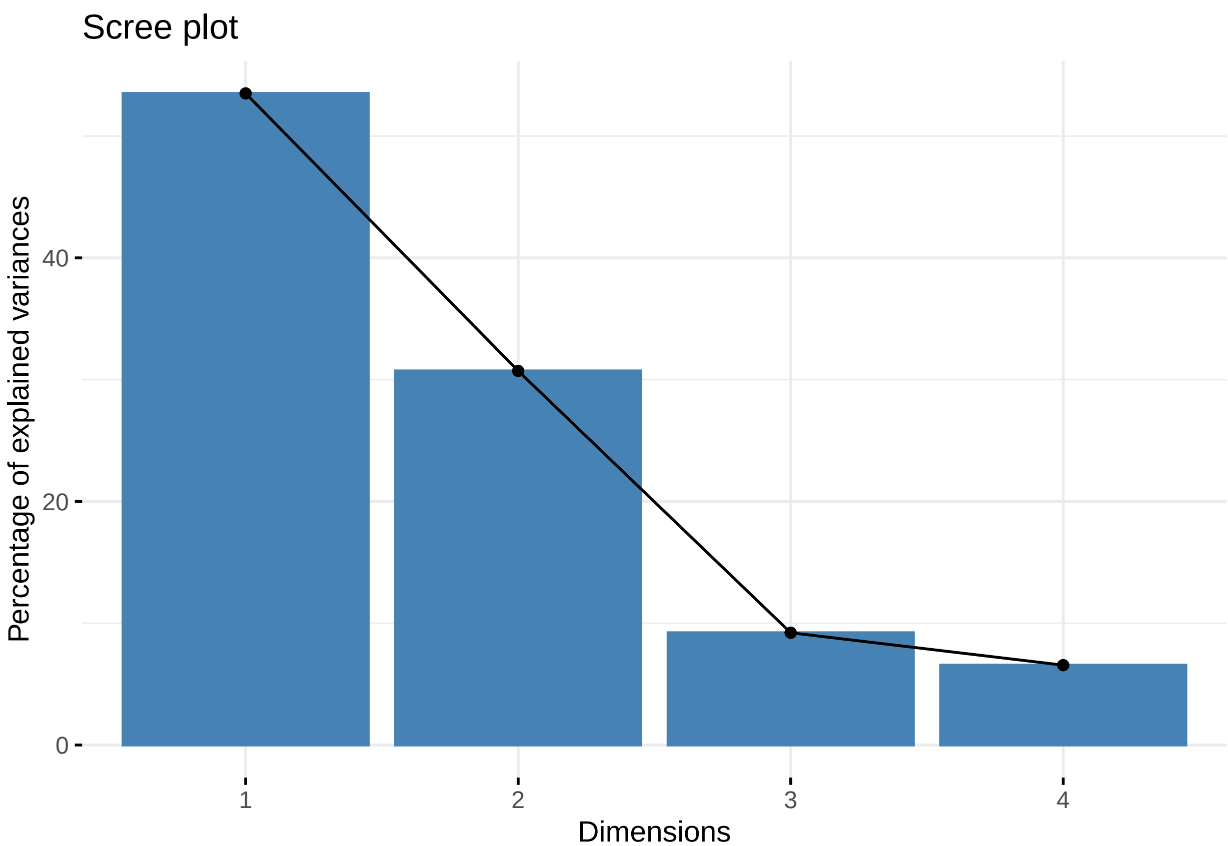
```r
cor(met_var[complete.cases(met_var),])
```

```
##             summer_tmmx winter_tmmx summer_rmax winter_rmax
## summer_tmmx   1.0000000  0.67627429 -0.33882533  -0.3849305
## winter_tmmx   0.6762743  1.00000000 -0.04948591  -0.1593851
## summer_rmax  -0.3388253 -0.04948591  1.00000000   0.6320484
## winter_rmax  -0.3849305 -0.15938510  0.63204841   1.0000000
```

```r
met_pca <- prcomp(met_var[complete.cases(met_var),], center = T, scale. = T)

met_loadings <- met_pca$rotation
```

```r
fviz_eig(met_pca)
```




```r
summ_pca <- summary(met_pca)

summ_pca$importance
```

```
##                             PC1      PC2       PC3       PC4
## Standard deviation     1.462968 1.108487 0.6072067 0.5121346
## Proportion of Variance 0.535070 0.307190 0.0921700 0.0655700
## Cumulative Proportion  0.535070 0.842250 0.9344300 1.0000000
```

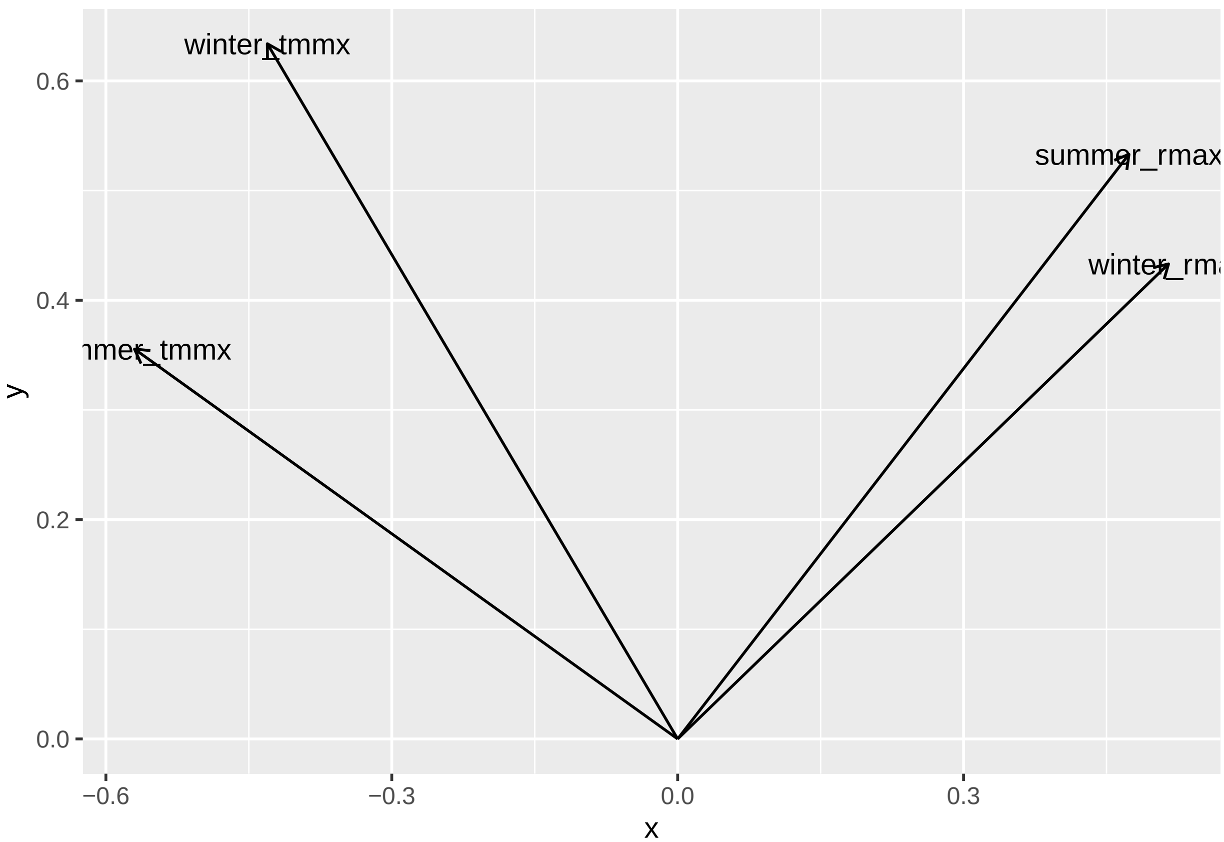
```r
round(met_loadings, digits = 2)
```

```
##               PC1  PC2   PC3   PC4
## summer_tmmx -0.57 0.36 -0.23 -0.71
## winter_tmmx -0.43 0.63  0.10  0.63
```

```
## summer_rmax  0.47 0.53  0.63 -0.31
## winter_rmax  0.52 0.43 -0.74  0.04
```

```r
# Extract loadings of the variables
met_loadings_df <- data.frame(Variables = rownames(met_pca$rotation), met_pca$rotation)

    # Plot
ggplot(met_loadings_df) +
  geom_segment(data = met_loadings_df, aes(x = 0, y = 0, xend = PC1,
      yend = PC2), arrow = arrow(length = unit(1/2, "picas")),
      color = "black") +
  annotate("text", x = (met_loadings_df$PC1), y = (met_loadings_df$PC2),
      label = met_loadings_df$Variables)
```



Data pre-processing decision: Use combined tmmx and rmax variables instead of splitting into summer and winter.

New data frame (all the new things you did)

Summary of data pre-processing decisions to reduce multicollinearity:

- Use first 5 flood risk PCs, scaling before and after PCA.
- Use first 3 pollution PCs, scaling before and after PCA.
- Use combined tmmx and rmax variables instead of splitting into summer and winter.

Below is the new set of covariates to be put in the model:

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

names(fhs_model_df)[19:(ncol(fhs_model_df) - 4)]
```

```
##  [1] "flood_risk_pc1"      "flood_risk_pc2"      "flood_risk_pc3"
##  [4] "flood_risk_pc4"      "flood_risk_pc5"      "EP_POV"
##  [7] "EP_UNEMP"            "EP_PCI"              "EP_NOHSDP"
## [10] "EP_AGE65"            "EP_AGE17"            "EP_DISABL"
## [13] "EP_SNGPNT"           "EP_MINRTY"           "EP_LIMENG"
## [16] "EP_MUNIT"            "EP_MOBILE"           "EP_CROWD"
## [19] "EP_NOVEH"            "EP_GROUPQ"           "EP_UNINSUR"
## [22] "pollute_conc_pc1"    "pollute_conc_pc2"    "pollute_conc_pc3"
## [25] "tmmx"                "rmax"                "Data_Value_CSMOKING"
```

VIF

Let's see if the data pre-processing has improved multicollinearity. With less multicollinearity, the beta estimates will have smaller standard errors.

```r
X <- fhs_model_df[, 19:(ncol(fhs_model_df) - 4)]

X           <- scale(X) # Scale covariates

X <- data.frame(X)
```

```r
vif(X)
```

```
##             Variables      VIF
## 1      flood_risk_pc1 1.199051
## 2      flood_risk_pc2 1.759936
## 3      flood_risk_pc3 1.112027
## 4      flood_risk_pc4 1.065996
## 5      flood_risk_pc5 1.040064
## 6              EP_POV 3.774213
## 7            EP_UNEMP 1.931246
## 8              EP_PCI 2.967336
## 9           EP_NOHSDP 5.412524
## 10           EP_AGE65 2.273221
## 11           EP_AGE17 2.641766
## 12          EP_DISABL 2.639541
## 13          EP_SNGPNT 2.708538
## 14          EP_MINRTY 3.683281
## 15          EP_LIMENG 3.555161
## 16           EP_MUNIT 1.936362
## 17          EP_MOBILE 1.662569
## 18           EP_CROWD 2.550514
## 19           EP_NOVEH 3.049356
## 20          EP_GROUPQ 1.485806
## 21         EP_UNINSUR 2.380176
## 22   pollute_conc_pc1 2.422443
## 23   pollute_conc_pc2 2.027510
## 24   pollute_conc_pc3 2.075880
## 25               tmmx 1.906402
## 26               rmax 1.557039
```

```
## 27 Data_Value_CSMOKING 6.046868
```

```r
vifstep(X)
```

```
## No variable from the 27 input variables has collinearity problem.
##
## The linear correlation coefficients ranges between:
## min correlation ( EP_CROWD ~ EP_MOBILE ):  0.000185482
## max correlation ( Data_Value_CSMOKING ~ EP_PCI ):  -0.711423
##
## ---------- VIFs of the remained variables --------
##              Variables      VIF
## 1       flood_risk_pc1 1.201002
## 2       flood_risk_pc2 1.679987
## 3       flood_risk_pc3 1.129659
## 4       flood_risk_pc4 1.049141
## 5       flood_risk_pc5 1.042458
## 6               EP_POV 3.510726
## 7             EP_UNEMP 1.906651
## 8               EP_PCI 2.891976
## 9            EP_NOHSDP 5.188742
## 10            EP_AGE65 2.260566
## 11            EP_AGE17 2.575492
## 12           EP_DISABL 2.686858
## 13           EP_SNGPNT 2.553137
## 14           EP_MINRTY 3.629656
## 15           EP_LIMENG 3.461468
## 16            EP_MUNIT 1.937904
## 17           EP_MOBILE 1.625602
## 18            EP_CROWD 2.622778
## 19            EP_NOVEH 2.853478
## 20           EP_GROUPQ 1.444714
## 21          EP_UNINSUR 2.433627
## 22    pollute_conc_pc1 2.330802
## 23    pollute_conc_pc2 1.971736
## 24    pollute_conc_pc3 2.008339
## 25                tmmx 1.871034
## 26                rmax 1.549218
## 27 Data_Value_CSMOKING 5.597920
```

`vifstep` wasn't able to detect multicollinearity issues. The VIF caps out at 6 for Data_Value_CSmoking. All the climate-related variables and principal components have VIFs that caps out at 2.4.

Non-spatial modeling (Preview of results prior to Bayesian CAR modeling)

```r
Y <- fhs_model_df$Data_Value_CHD

X <- fhs_model_df[, 19:(ncol(fhs_model_df) - 4)]
```

```r
X           <- scale(X) # Scale covariates
X[is.na(X)] <- 0        # Fill in missing values with the mean
```

```r
fhs_lm <- lm(Y ~ X)
```

```r
summary(fhs_lm)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9636 -0.4815 -0.0187  0.4573 17.8527
##
## Coefficients:
##                        Estimate Std. Error  t value Pr(>|t|)
## (Intercept)            6.660097   0.003143 2118.805  < 2e-16 ***
## Xflood_risk_pc1       -0.011466   0.003585   -3.198  0.00138 **
## Xflood_risk_pc2        0.025701   0.004026    6.384 1.73e-10 ***
## Xflood_risk_pc3        0.005510   0.003451    1.597  0.11037
## Xflood_risk_pc4        0.009898   0.003345    2.959  0.00308 **
## Xflood_risk_pc5        0.009249   0.003337    2.772  0.00558 **
## XEP_POV                0.340208   0.005889   57.773  < 2e-16 ***
## XEP_UNEMP              0.012945   0.004342    2.982  0.00287 **
## XEP_PCI               -0.039565   0.005174   -7.647 2.08e-14 ***
## XEP_NOHSDP             0.210132   0.007385   28.454  < 2e-16 ***
## XEP_AGE65              1.475965   0.004740  311.363  < 2e-16 ***
## XEP_AGE17              0.351097   0.005291   66.358  < 2e-16 ***
## XEP_DISABL             0.346387   0.005135   67.457  < 2e-16 ***
## XEP_SNGPNT            -0.104944   0.005083  -20.645  < 2e-16 ***
## XEP_MINRTY            -0.083293   0.005914  -14.083  < 2e-16 ***
## XEP_LIMENG            -0.006972   0.006224   -1.120  0.26265
## XEP_MUNIT             -0.056433   0.004479  -12.601  < 2e-16 ***
## XEP_MOBILE             0.039969   0.003981   10.040  < 2e-16 ***
## XEP_CROWD             -0.078198   0.005218  -14.986  < 2e-16 ***
## XEP_NOVEH              0.042593   0.005558    7.663 1.84e-14 ***
## XEP_GROUPQ            -0.072995   0.003877  -18.830  < 2e-16 ***
## XEP_UNINSUR            0.153359   0.004806   31.908  < 2e-16 ***
## Xpollute_conc_pc1      0.002400   0.004886    0.491  0.62334
## Xpollute_conc_pc2     -0.013366   0.004273   -3.128  0.00176 **
## Xpollute_conc_pc3     -0.082797   0.004637  -17.856  < 2e-16 ***
## Xtmmx                  0.150575   0.004282   35.166  < 2e-16 ***
## Xrmax                  0.091136   0.003969   22.962  < 2e-16 ***
## XData_Value_CSMOKING   0.853197   0.007408  115.167  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.842 on 71807 degrees of freedom
##   (702 observations deleted due to missingness)
## Multiple R-squared:  0.8545, Adjusted R-squared:  0.8545
## F-statistic: 1.562e+04 on 27 and 71807 DF,  p-value: < 2.2e-16
```