NMF Final (Only nndsvd 5 component without ozone)

William Zhang, Eva, Jerry, Meredith

2025-01-24

```
# load the packages
library(NMF)
library(tidyverse)
library(gridExtra)
library(readxl)
library(circular)
library(lwgeom)
library(units)
```

Procedure

1. Remove hourly observation with missing observation for any chemical
2. Remove background noise level using min values (except for chemicals with minimum value < 2*LOD and maximum value > 100*LOD)
3. Zero values are converted to a random value between 0 and 0.5*LOD
4. Normalize using min and max
5. Remove Ozone (wouldn't affect # of obs.)

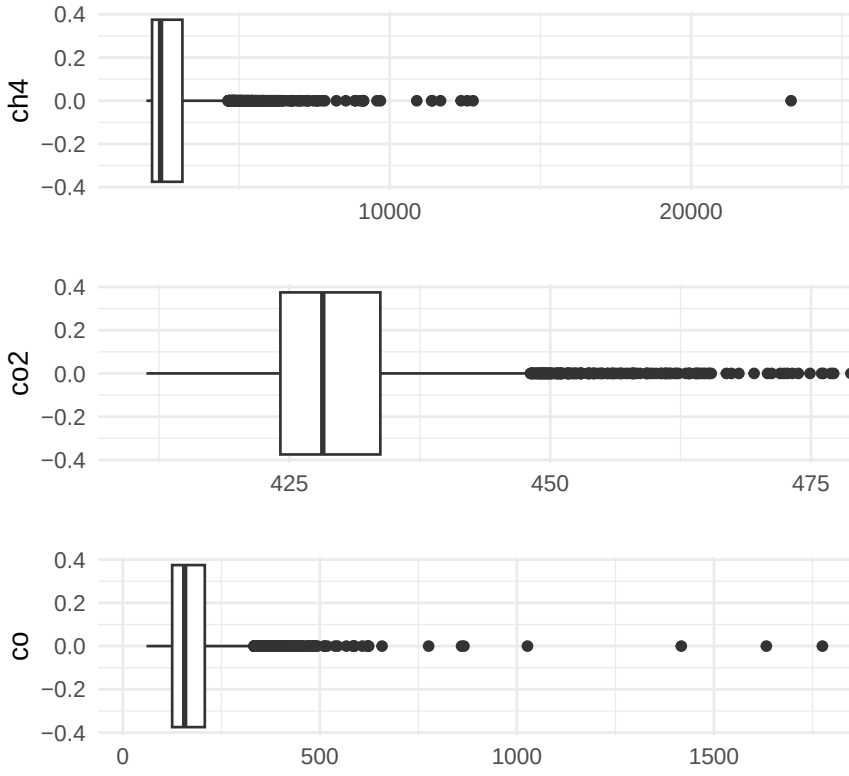
Reading the data

```
hourly_data <- readRDS("../DataProcessing/Trailer_hourly_merge_20240905.rds")
```

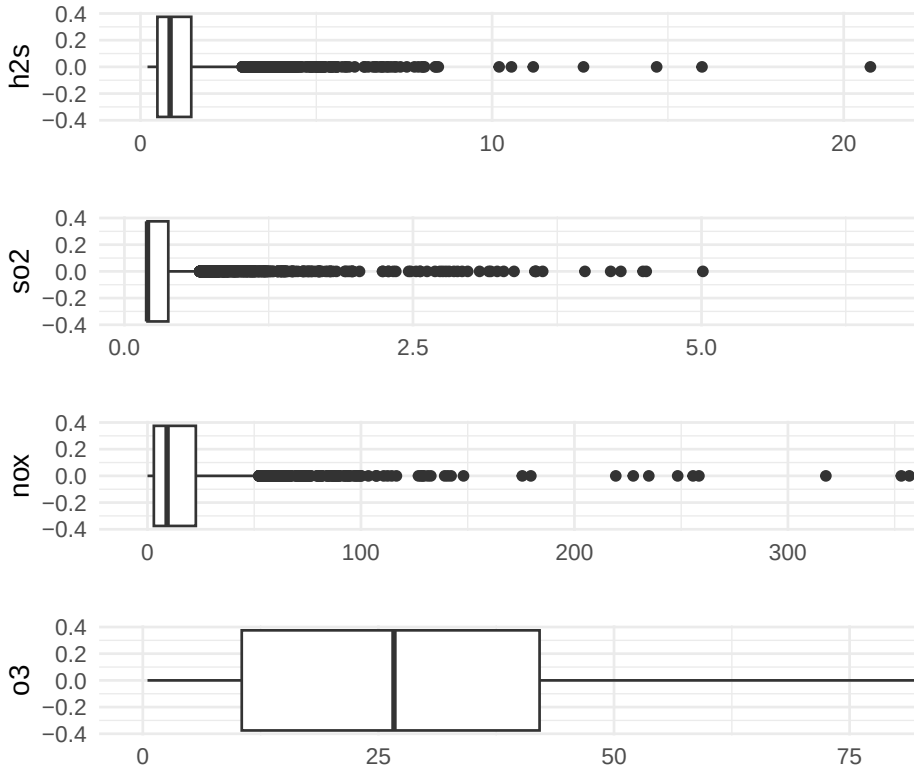
```
# PROCEDURE STEP 1:
hourly_data <- hourly_data %>% rename('co2' = 'co2_ppm')

vocs <- c("ethane", "ethene", "propane", "propene",
                                        "1_3-butadiene", "i-butane", "n-butane",
                                        "acetylene", "cyclopentane", "i-pentane",
                                        "n-pentane", "n-hexane", "isoprene", "n-heptane",
                                        "benzene", "n-octane", "toluene", "ethyl-benzene",
                                        "m&p-xylene", "o-xylene")


non_vocs <- c('ch4', 'co2', 'co', 'h2s', 'so2', 'nox', 'o3')

# remove row with missing obs for any chemical
hourly_nona <- hourly_data %>%
  select(any_of(c('day', 'time_utc', vocs, non_vocs, 'wdr_deg', 'wsp_ms'))) %>%
  na.omit()

# retrieving the vocs, removing everything else except the vocs
```

```r
hourly_vocs <- hourly_nona %>% select(any_of(vocs))

# retrieving the non-vocs: co2_ppm, nox, ch4, h2s, so2, o3
# double check this
hourly_non_vocs <- hourly_nona %>% select(any_of(non_vocs))

hourly_full_nona <- cbind(hourly_non_vocs, hourly_vocs)

# retrive a vector of yearmonth
hourly_dates <- hourly_nona %>%
  mutate(yearmonth = substring(day, 0, 7)) %>%
  pull(yearmonth)
```

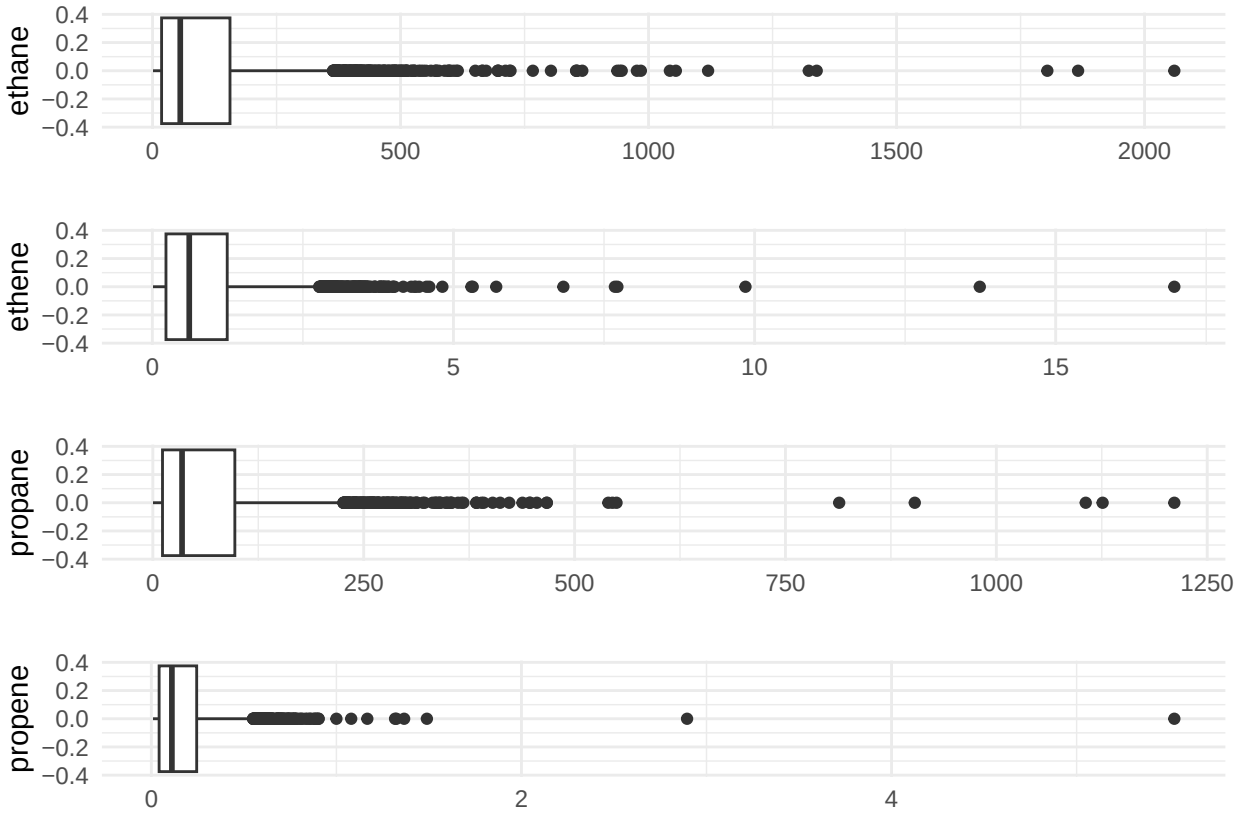
Data visualisation

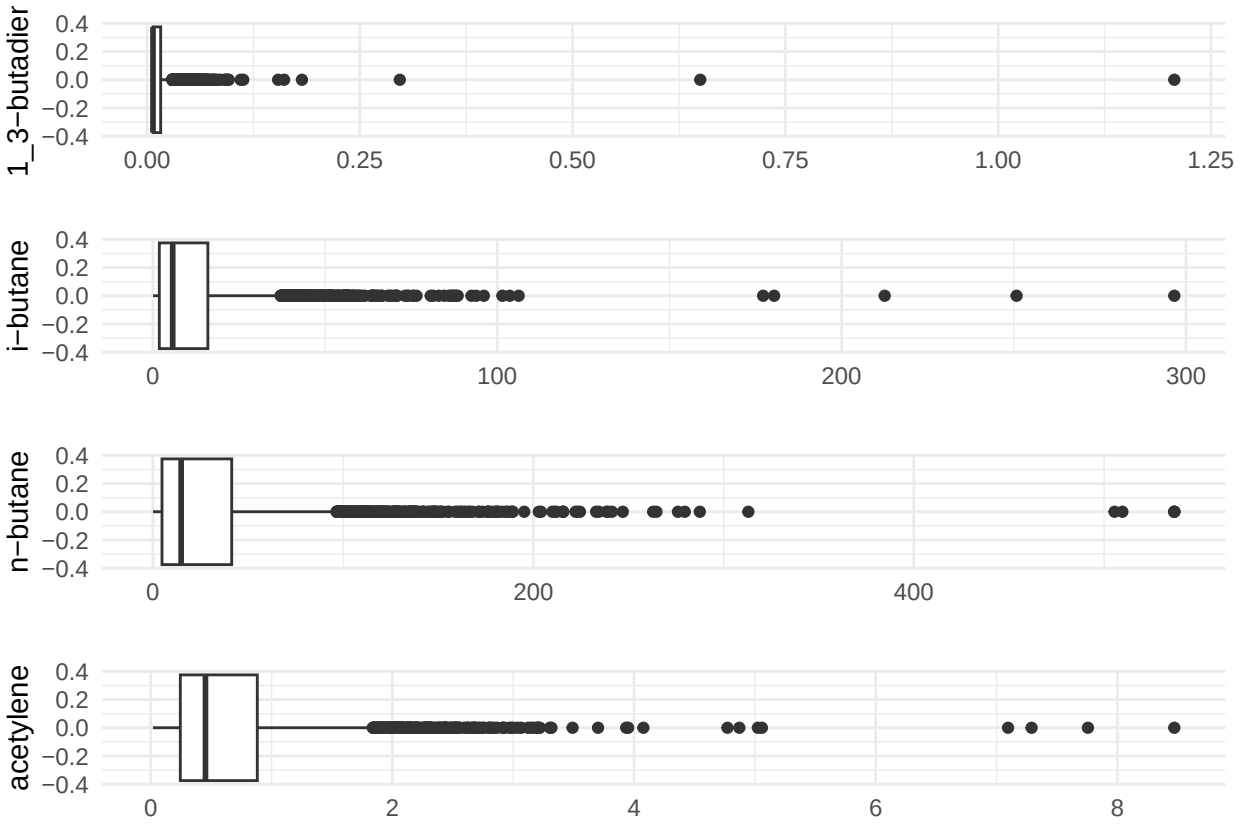


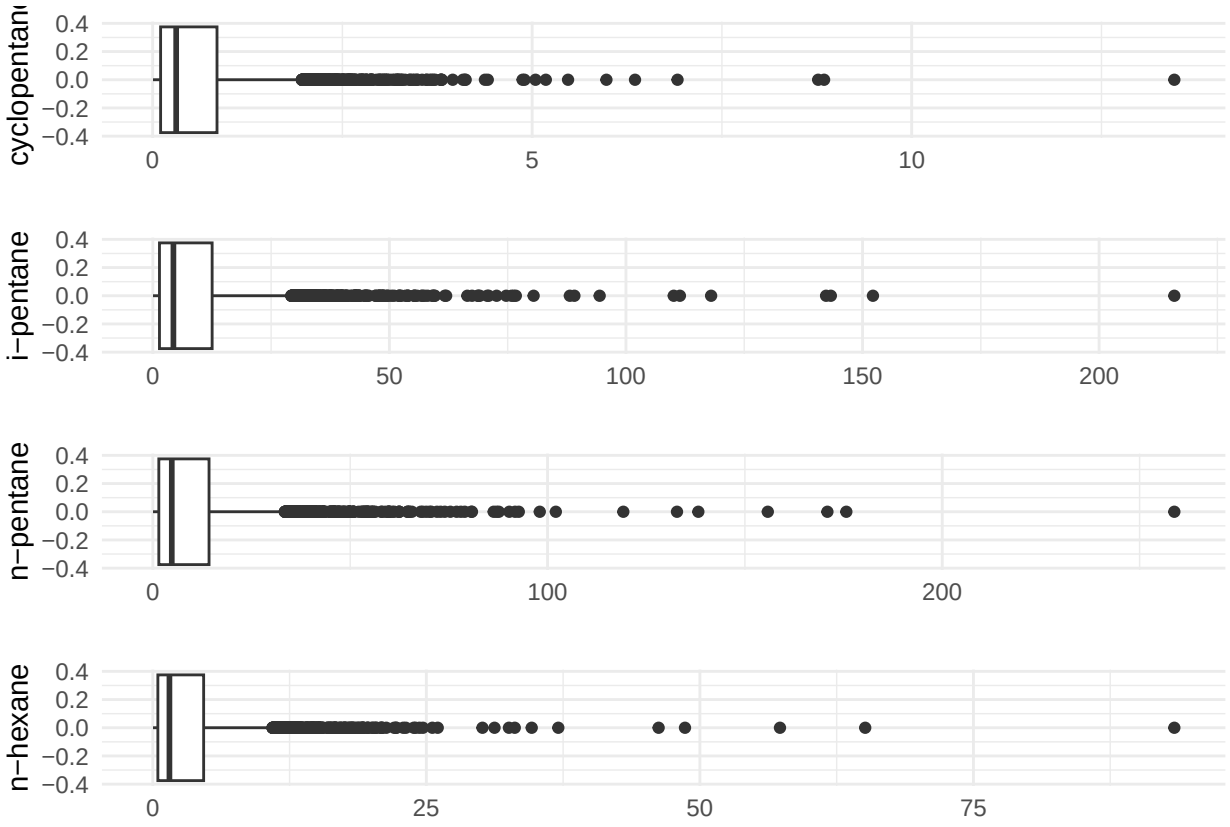
- Boxplots of the hourly concentrations non-voc

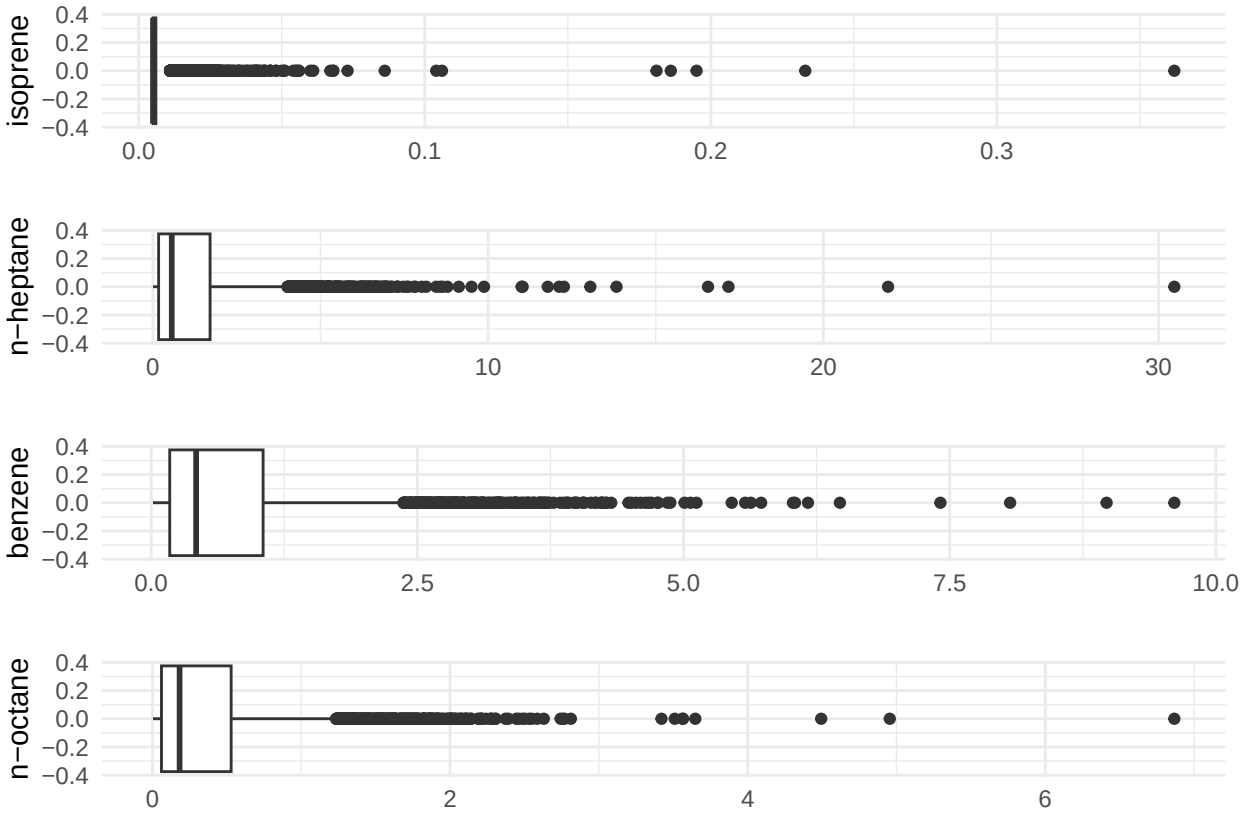


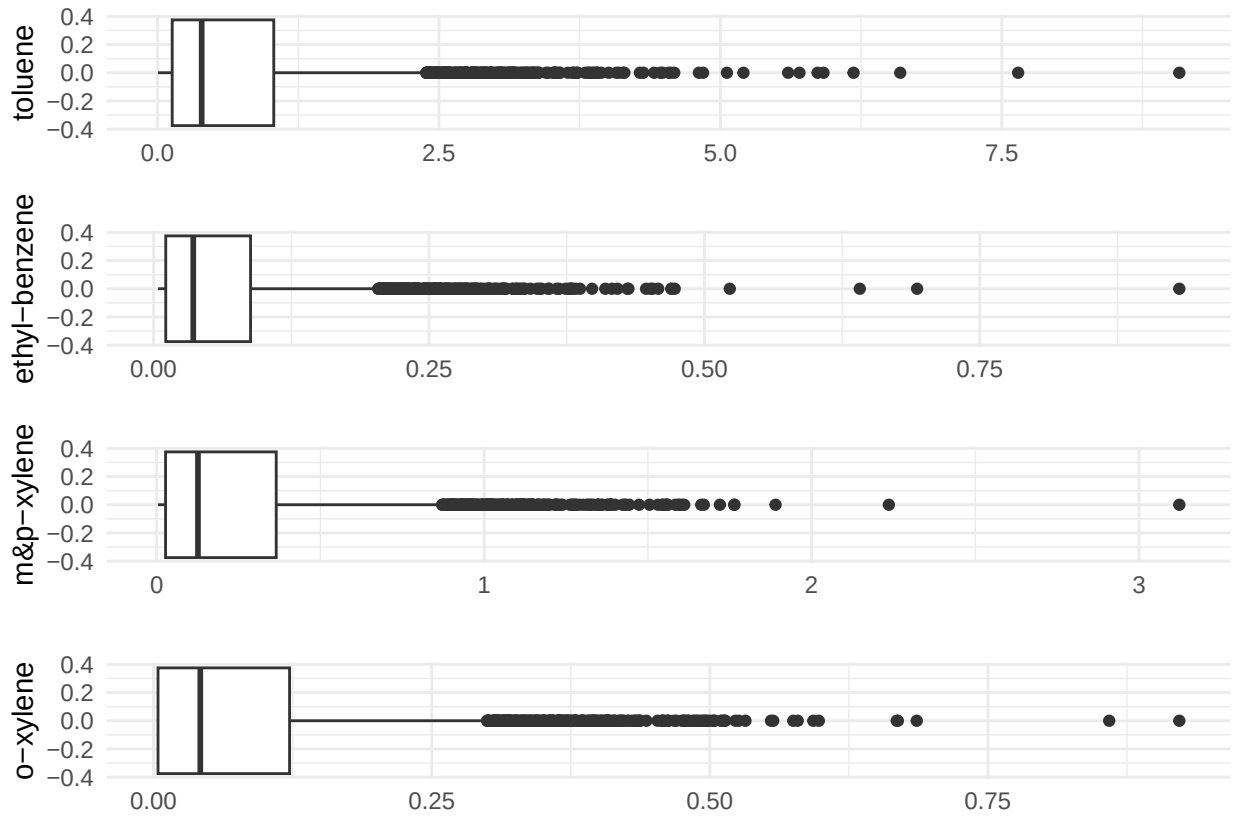
- Boxplots sulfur compounds, NOx, ozone
- Boxplots VOCs


ethane
0.4
0.2
0.0
−0.2
−0.4
0
500
1000
1500
2000
ethene
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
propane
0.4
0.2
0.0
−0.2
−0.4
0
250
500
750
1000
1250
propene
0.4
0.2
0.0
−0.2
−0.4
0
2
4


1_3−butadier
0.4
0.2
0.0
−0.2
−0.4
0.00
0.25
0.50
0.75
1.00
1.25
i−butane
0
100
200
300
n−butane
0
200
400
acetylene
0
2
4
6
8


cyclopentan
i−pentane
n−pentane
n−hexane
0.4
0.2
0.0
−0.2
−0.4
0
5
10
50
100
150
200
25
75


isoprene
0.4
0.2
0.0
−0.2
−0.4
0.0
0.1
0.2
0.3
n−heptane
0
10
20
30
benzene
0.0
2.5
5.0
7.5
10.0
n−octane
0
2
4
6



Data pre-processing

- STEP 1: Limits of detection

```
# Define LOD for each chemical
LOD_non_voc <- c('ch4' = 0.9,
                 'co2' = 0.0433,
                 'co' = 40,
                 'h2s' = 0.4,
                 'so2' = 0.4,
                 'nox' = 0.05,
                 'o3' = 1)

# LOD_voc_monthly <- read_csv('../data/LNM_VOC_LOD_Rounded.csv') %>% select(-1)
#
# # extract the yearmonth from date variables
# LOD_voc_monthly <- LOD_voc_monthly %>%
#   mutate(yearmonth = strftime(as.POSIXct(start_date, format = '%Y-%m-%d %H:%M:%S',
#                                          tz = 'UTC'), '%Y-%m'))
#
# LOD_voc_monthly <- LOD_voc_monthly %>%
#   select(-c(start_date, end_date)) %>%
#   select(!any_of(ends_with('half_ldl')))
#
# colnames(LOD_voc_monthly) <- str_replace_all(names(LOD_voc_monthly), '_ldl', '')
```

```
LOD_voc_avg <- read_xlsx('../data/LNM_VOC_Uncertainties.xlsx', skip = 1)
LOD_voc_avg <- LOD_voc_avg %>%
  select(1, 4) %>%
  rename('LOD' = 2, 'chemical' = 1) %>%
  head(20)
```

- STEP 2: Background correction

```
##           ch4           co2            co           h2s           so2
##      1928.000       411.300        59.910         0.200         0.200
##           nox            o3        ethane        ethene       propane
##         0.025         0.500         0.916         0.011         0.224
##       propene 1_3-butadiene      i-butane      n-butane     acetylene
##         0.009         0.007         0.035         0.090         0.019
##  cyclopentane     i-pentane     n-pentane      n-hexane      isoprene
##         0.005         0.038         0.042         0.021         0.005
##     n-heptane       benzene      n-octane       toluene ethyl-benzene
##         0.004         0.017         0.004         0.004         0.004
##    m&p-xylene      o-xylene
##         0.004         0.004
```

- Summary statistics of backgrounds and extremes

```
get_info <- function(column) {
  N <- length(column)
  background <- quantile(column, 0)
  quantile1 <- quantile(column, 0.01)
  quantile99 <- quantile(column, 0.99)
  n_background <- sum(column == background)
  max <- max(column)
  return(c(N, quantile1, quantile99, max, background, n_background))
}

info_table <- hourly_full_nona %>%
  reframe(across(everything(), ~ get_info(.x)))

info_table <- info_table %>%
  mutate(rownames = c('N', '1st percentile', '99th percentile', 'Max',
                      'Background', '# Background')) %>%
  pivot_longer(-rownames) %>%
  pivot_wider(names_from = rownames, values_from = value)

knitr::kable(info_table)
```

name	N	1st percentile	99th percentile	Max	Background	# Background
ch4	4788	1962.98700	6286.12400	34010.900	1928.000	1
co2	4788	416.47870	460.62260	503.990	411.300	1
co	4788	84.23050	442.08860	2513.440	59.910	1
h2s	4788	0.20000	5.20986	27.700	0.200	829
so2	4788	0.20000	1.78686	8.578	0.200	3266
nox	4788	0.22974	89.72371	452.959	0.025	2
o3	4788	0.50000	76.02600	103.100	0.500	259
ethane	4788	1.84422	526.44700	2060.000	0.916	1
ethene	4788	0.01100	3.50826	16.970	0.011	163

name	N	1st percentile	99th percentile	Max	Background	# Background
propane	4788	0.84674	300.79000	1211.000	0.224	1
propene	4788	0.00900	0.69739	5.528	0.009	411
1_3-butadiene	4788	0.00700	0.05900	1.207	0.007	3357
i-butane	4788	0.15148	60.89400	296.600	0.035	1
n-butane	4788	0.37248	166.52100	536.900	0.090	1
acetylene	4788	0.04900	2.61304	8.471	0.019	2
cyclopentane	4788	0.00500	3.06899	13.460	0.005	96
i-pentane	4788	0.10987	49.60210	215.900	0.038	1
n-pentane	4788	0.10487	55.95980	258.800	0.042	1
n-hexane	4788	0.04300	18.17780	93.360	0.021	2
isoprene	4788	0.00500	0.03313	0.362	0.005	2816
n-heptane	4788	0.01500	6.57669	30.470	0.004	5
benzene	4788	0.02800	3.78693	9.610	0.017	3
n-octane	4788	0.00400	2.00839	6.867	0.004	100
toluene	4788	0.01300	3.52165	9.077	0.004	11
ethyl-benzene	4788	0.00400	0.31613	0.931	0.004	918
m&p-xylene	4788	0.00400	1.29156	3.123	0.004	851
o-xylene	4788	0.00400	0.45700	0.922	0.004	1330

- STEP 2 processing continued: background correction
- adjustments that were made according to paper: Gunnar's paper section 2.2 and Guha 3.3
- Check whether chemical has background noise level that needs to be removed
- NO ADJUSTMENT if minimum value < 2xLOD and maximum value > 100xLOD

```
adjusting_neg_bg_from_lod <- function(chemical, LOD, background, hourly_data){
    # get min and max
    min_value <- min(hourly_data[chemical], na.rm = TRUE)
    max_value <- max(hourly_data[chemical], na.rm = TRUE)
    # if min less than double LOD or max > 100 times LOD
    # adjust to -100 (for entire column???)
    if (min_value < 2 * LOD & max_value > 100 * LOD ){
      return (0)
    }
  return (background)
}
```

- Check if background is negligible for non voc
- merge background and LOD

```
background_lod_non_voc <- tibble(chemical = non_vocs,
                                 LOD = LOD_non_voc,
                                 background = unname(background_levels[non_vocs]))
adjusted_background_non_voc <- background_lod_non_voc %>%
  rowwise() %>%
  mutate(min = min(hourly_full_nona[chemical], na.rm = TRUE),
         LODx2 = 2 * LOD,
         criterion1 = min(hourly_full_nona[chemical], na.rm = TRUE) < 2 * LOD,
         max = max(hourly_full_nona[chemical], na.rm = TRUE),
         LODx100 = 100 * LOD,
         criterion2 = max(hourly_full_nona[chemical], na.rm = TRUE) > 100 * LOD,
         adjusted_background = adjusting_neg_bg_from_lod(chemical, LOD, background,
                                                         hourly_full_nona))
```

- Check if background is negligible for voc
- merge background and LOD

```
background_lod_voc <- LOD_voc_avg %>%
  left_join(tibble(chemical = setdiff(names(background_levels), non_vocs),
                   background = background_levels[setdiff(names(background_levels),
                                                          non_vocs)]))
adjusted_background_voc <- background_lod_voc %>%
  rowwise() %>%
  mutate(min = min(hourly_full_nona[chemical], na.rm = TRUE),
         LODx2 = 2 * LOD,
         criterion1 = min(hourly_full_nona[chemical], na.rm = TRUE) < 2 * LOD,
         max = max(hourly_full_nona[chemical], na.rm = TRUE),
         LODx100 = 100 * LOD,
         criterion2 = max(hourly_full_nona[chemical], na.rm = TRUE) > 100 * LOD,
         adjusted_background = adjusting_neg_bg_from_lod(chemical, LOD, background,
                                                         hourly_full_nona))
```

- create dataset with background removed

```
# So now we have the adjusted background concentrations
hourly_nona_bgrm <- hourly_full_nona %>%
  mutate(across(adjusted_background_non_voc$chemical,
                ~  .x - adjusted_background_non_voc$adjusted_background[
                    adjusted_background_non_voc$chemical == cur_column()]))
hourly_nona_bgrm <- hourly_nona_bgrm %>%
  mutate(across(adjusted_background_voc$chemical,
                ~  .x - adjusted_background_voc$adjusted_background[
                    adjusted_background_voc$chemical == cur_column()]))
```

- check number of 0 values per compound

```
# look at zero values
colSums(hourly_nona_bgrm == 0)
```

```
##           ch4           co2            co           h2s           so2
##             1             1             1           829          3266
##           nox            o3       ethane       ethene       propane
##             0             0             1             0             1
##       propene 1_3-butadiene      i-butane      n-butane     acetylene
##             0          3357             1             1             0
##  cyclopentane     i-pentane     n-pentane      n-hexane      isoprene
##             0             1             1             2          2816
##     n-heptane       benzene      n-octane       toluene ethyl-benzene
##             0             0             0             0             0
##    m&p-xylene      o-xylene
##             0             0
```

- STEP 3: replace zero values with random values between 0 and 0.5xLOD

```
set.seed(123)
replace_zero_with_random <- function(column, name, LOD_df){
  LOD <- LOD_df$LOD[LOD_df$chemical == name]
  column <- if_else(column == 0, round(runif(length(column), 0, 0.5 * LOD), 3), column)
  return (column)
}
```

```
hourly_nona_bgrm_zerorepl <- hourly_nona_bgrm %>%
  mutate(across(adjusted_background_non_voc$chemical,
                ~ replace_zero_with_random(.x, cur_column(), adjusted_background_non_voc)))

hourly_nona_bgrm_zerorepl <- hourly_nona_bgrm_zerorepl %>%
  mutate(across(adjusted_background_voc$chemical,
                ~ replace_zero_with_random(.x, cur_column(), adjusted_background_voc)))
```

- STEP 4: Normalize the non-vocs

```
#normalizing function
normalize_column <- function(column){
  background <- quantile(column, 0)
  max <- quantile(column, 1) # this could be adjusted
  return ((column - background)/(max - background))
}
```

- STEP 4: Normalize all

```
# normalize all
hourly_nona_bgrm_zerorepl_norm <- as_tibble(sapply(as.list(hourly_nona_bgrm_zerorepl),
                                                   normalize_column))
#normalize the NON_VOC
summary(hourly_nona_bgrm_zerorepl_norm)
```

```
##       ch4                co2                co                h2s
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00579   1st Qu.:0.1384   1st Qu.:0.02592   1st Qu.:0.01022
##  Median :0.01460   Median :0.1823   Median :0.03884   Median :0.02335
##  Mean   :0.02683   Mean   :0.2000   Mean   :0.04761   Mean   :0.03501
##  3rd Qu.:0.03720   3rd Qu.:0.2418   3rd Qu.:0.05970   3rd Qu.:0.04525
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##       so2                nox                o3              ethane
##  Min.   :0.000000   Min.   :0.000000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.007878   1st Qu.:0.006534   1st Qu.:0.09747   1st Qu.:0.008385
##  Median :0.015994   Median :0.020262   Median :0.25487   Median :0.026671
##  Mean   :0.026287   Mean   :0.036440   Mean   :0.26676   Mean   :0.050992
##  3rd Qu.:0.023633   3rd Qu.:0.049978   3rd Qu.:0.40546   3rd Qu.:0.075375
##  Max.   :1.000000   Max.   :1.000000   Max.   :1.00000   Max.   :1.000000
##      ethene             propane            propene          1_3-butadiene
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.01268   1st Qu.:0.009283   1st Qu.:0.005979   1st Qu.:0.002500
##  Median :0.03547   Median :0.028409   Median :0.018482   Median :0.004167
##  Mean   :0.05042   Mean   :0.053803   Mean   :0.028772   Mean   :0.007371
##  3rd Qu.:0.07266   3rd Qu.:0.080130   3rd Qu.:0.042761   3rd Qu.:0.007500
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.000000   Max.   :1.000000
##     i-butane           n-butane          acetylene        cyclopentane
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.00614   1st Qu.:0.008777   1st Qu.:0.02674   1st Qu.:0.007432
##  Median :0.01925   Median :0.027522   Median :0.05135   Median :0.022668
##  Mean   :0.03837   Mean   :0.054900   Mean   :0.07436   Mean   :0.043730
##  3rd Qu.:0.05369   3rd Qu.:0.077042   3rd Qu.:0.10211   3rd Qu.:0.062653
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.00000   Max.   :1.000000
##    i-pentane          n-pentane          n-hexane          isoprene
##  Min.   :0.000000   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000
```

```
##  1st Qu.:0.006303   1st Qu.:0.005681   1st Qu.:0.004703   1st Qu.:0.002801
##  Median :0.019941   Median :0.018371   Median :0.016039   Median :0.005602
##  Mean   :0.041094   Mean   :0.038859   Mean   :0.034979   Mean   :0.010315
##  3rd Qu.:0.057857   3rd Qu.:0.054837   3rd Qu.:0.049544   3rd Qu.:0.011204
##  Max.   :1.000000   Max.   :1.000000   Max.   :1.000000   Max.   :1.000000
##    n-heptane          benzene           n-octane           toluene
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.005473   1st Qu.:0.01637   1st Qu.:0.008269   1st Qu.:0.01389
##  Median :0.018348   Median :0.04222   Median :0.026009   Median :0.04276
##  Mean   :0.039328   Mean   :0.07655   Mean   :0.054341   Mean   :0.07825
##  3rd Qu.:0.055866   3rd Qu.:0.10779   3rd Qu.:0.076497   3rd Qu.:0.11333
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.000000   Max.   :1.00000
##  ethyl-benzene        m&p-xylene          o-xylene
##  Min.   :0.000000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.007551   1st Qu.:0.007374   1st Qu.:0.00000
##  Median :0.034520   Median :0.039115   Median :0.04139
##  Mean   :0.062378   Mean   :0.077508   Mean   :0.08650
##  3rd Qu.:0.090615   3rd Qu.:0.115742   3rd Qu.:0.12881
##  Max.   :1.000000   Max.   :1.000000   Max.   :1.00000
```

- FINAL step: create matrix of processed and normalized concentrations for NMF

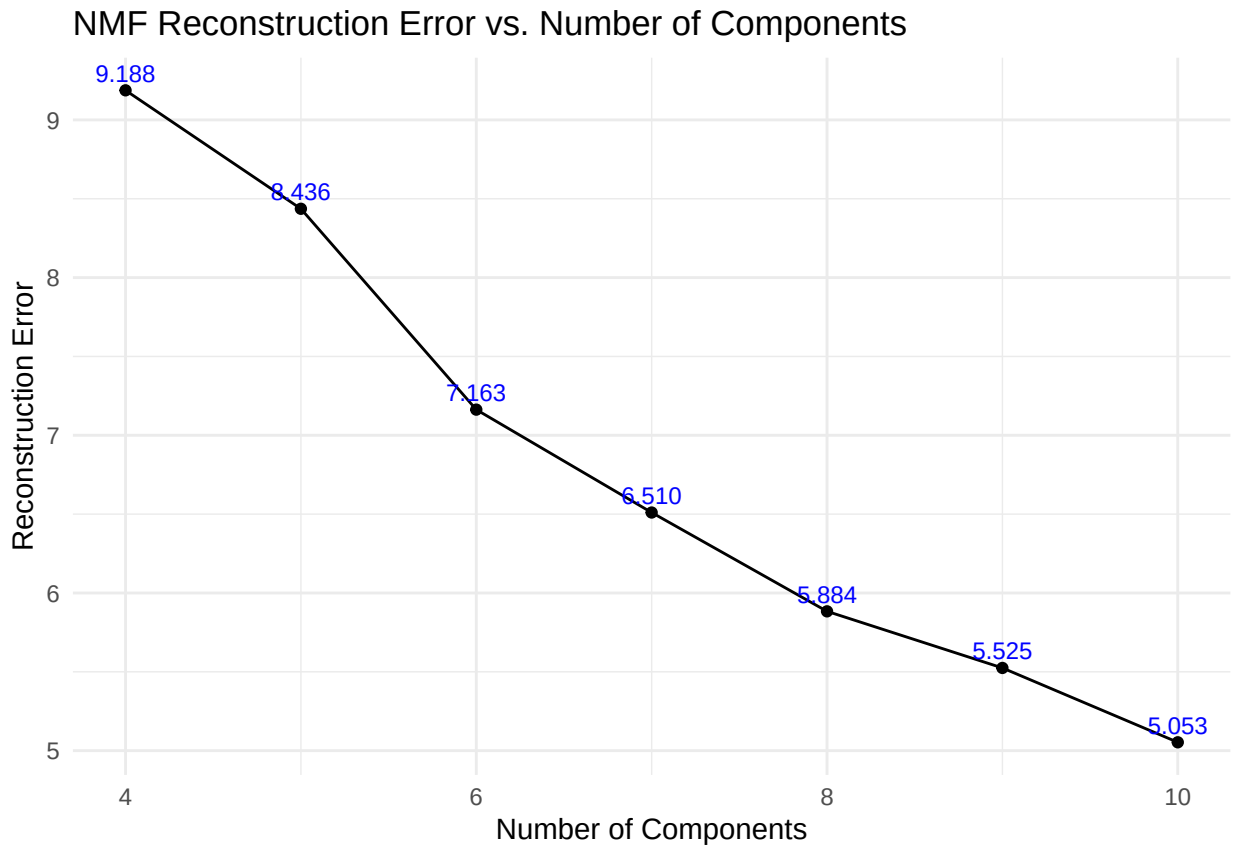
```
normalized_matrix <- as.matrix(hourly_nona_bgrm_zerorepl_norm)
#important: using the normalized VOCs for this file
```

NMF section

Helper for source contributions plots

Apply NMF using 'nndsvd' seed and KL divergence

```
components <- 4:10
errors <- numeric(length(components) - 4)

# Loop over the number of components
# for (n in components) {
#   nmf_result <- nmf(normalized_matrix, rank = n, method = "KL", seed='nndsvd')
#   reconstruction <- basis(nmf_result) %*% coef(nmf_result)
#   error <- norm(normalized_matrix - reconstruction, type = "F")
#   errors[n-3] <- error
#   print(paste0('Completed ', n - 3, ' out of 7'))
# }
#
# saveRDS(errors, 'errors_norm.rds')

errors <- readRDS('errors_norm.rds')
```



NMF with 5 source factors without ozone

- remove ozone
- use KL divergence loss with svd seed
- Extract W (basis) and H (coefs) matrices
- Calculate variance explained in all 5 factors
- Calculate variance explained by each factor

```
normalized_matrix_less_o3 <-
  normalized_matrix[ ,setdiff(colnames(normalized_matrix), "o3")]

nmf_result_5c_less_o3 <- nmf(normalized_matrix_less_o3, rank = 5,
                             method = "KL", seed='nndsvd')
```

```
## Warning in sqrt(S[i] * termn) * uun: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termn) * vvn: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termn) * uun: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termn) * vvn: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termn) * uun: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.
```

```
## Warning in sqrt(S[i] * termn) * vvn: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termp) * uup: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termp) * vvp: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.
```

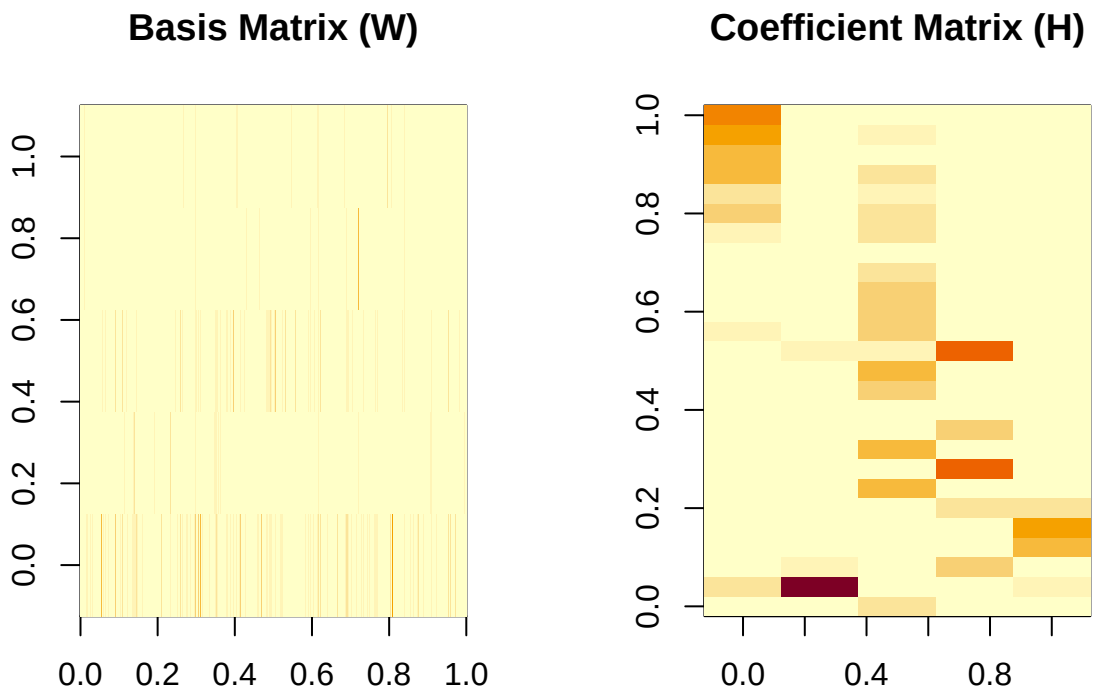
```r
basis_matrix_5c_less_o3 <- basis(nmf_result_5c_less_o3) #W
coef_matrix_5c_less_o3 <- coef(nmf_result_5c_less_o3) #H

# get variance explained by the factors (total residuals)
reconstruct<-fitted(nmf_result_5c_less_o3)

tss <- sum((normalized_matrix_less_o3 - mean(normalized_matrix_less_o3))^2)
rss <- sum((normalized_matrix_less_o3 - reconstruct)^2)
variance_explained <- 1 - (rss / tss)
variance_explained
```

```
## [1] 0.9212864
```

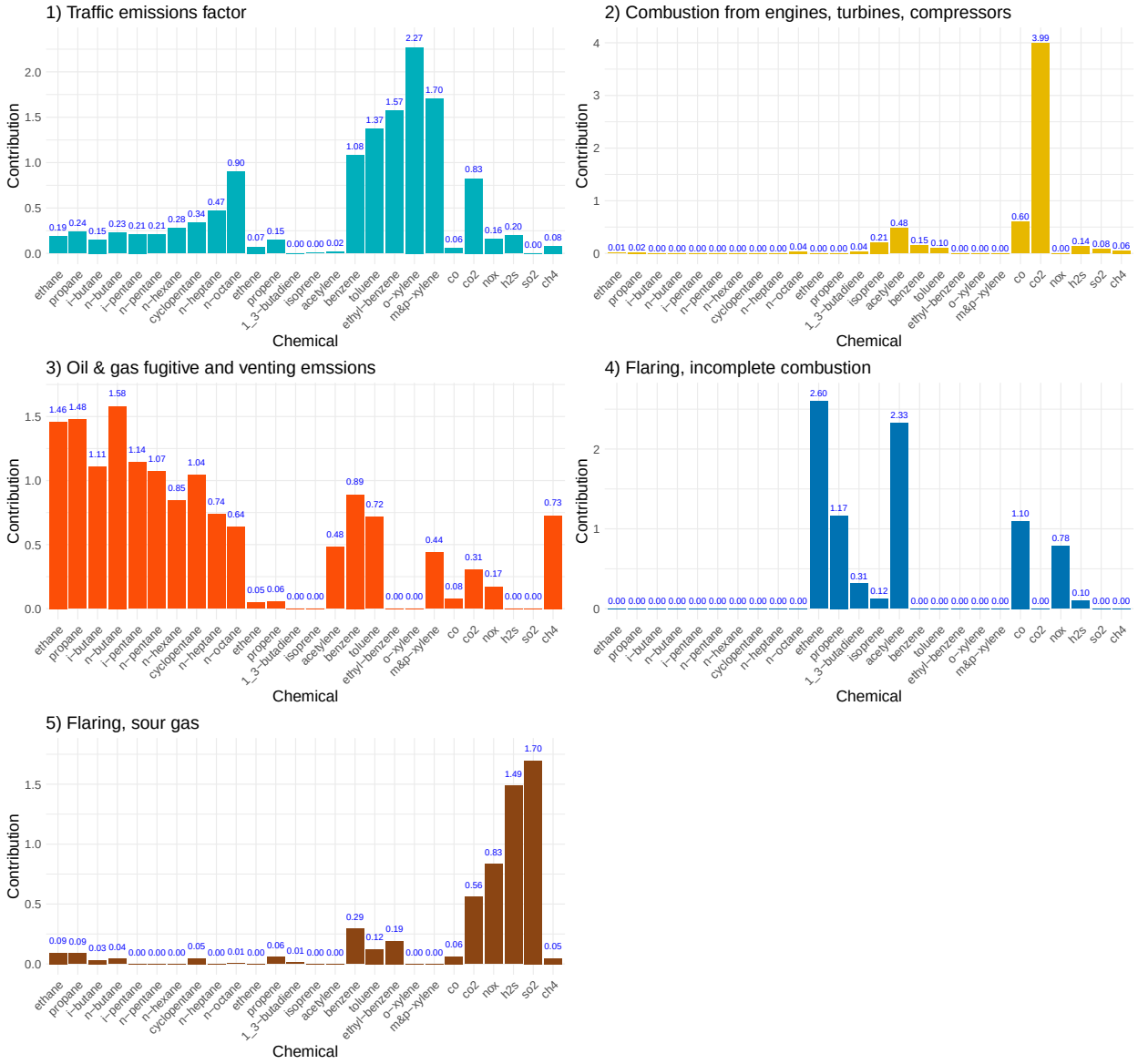
```r
# get variance explained by each factor separately
# Compute variance explained by each factor
# Initialize variance explained tracker
variance_explained_factors <- numeric(5)

# Incrementally add factors and calculate variance explained
reconstruction <- matrix(0, nrow = nrow(basis_matrix_5c_less_o3), ncol = ncol(coef_matrix_5c_less_o3))

for (i in 1:5) {
  # Add the i-th factor to the reconstruction
  reconstruction <- reconstruction + (basis_matrix_5c_less_o3[, i, drop=FALSE] %*% coef_matrix_5c_less_

  # Compute Residual Sum of Squares (RSS)
  rss_f <- sum((normalized_matrix_less_o3 - reconstruction)^2)

  # Compute Variance Explained by adding this factor
  variance_explained_factors[i] <- 1 - (rss_f / tss)
}

# Print variance explained by each factor cumulatively
variance_explained_factors
```

```
## [1] 0.2395401 0.5113683 0.8113445 0.8921360 0.9212864
```

```r
par(mfrow = c(1, 2))
image(basis_matrix_5c_less_o3, main = "Basis Matrix (W)")
image(coef_matrix_5c_less_o3, main = "Coefficient Matrix (H)")
```



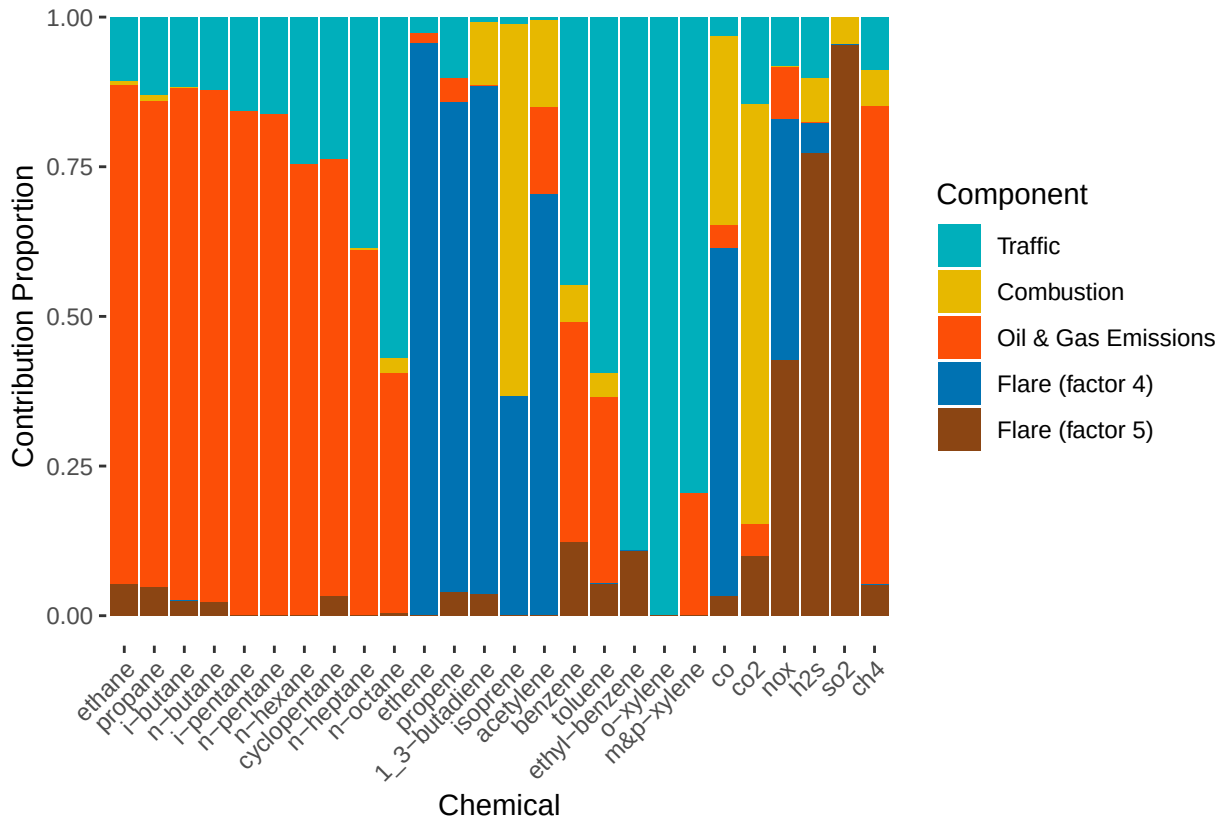

```
# Convert H to a data frame for ggplot
H_df_5c_less_o3 <- as.data.frame(coef_matrix_5c_less_o3)
# Add a column for chemicals
H_df_5c_less_o3$Component <- rownames(H_df_5c_less_o3)

# Reshape data to long format
H_long_5c_less_o3 <- pivot_longer(H_df_5c_less_o3, cols = -Component,
                                  names_to = "Chemical", values_to = "Contribution")

# Plot
nmfplt_1_svd_5c_less_o3 <- get_component_plot(H_long_5c_less_o3,
                                              '1', '1) Traffic emissions factor')
nmfplt_2_svd_5c_less_o3 <- get_component_plot(H_long_5c_less_o3,
                                              '2', '2) Combustion from engines, turbines, compressors')
nmfplt_3_svd_5c_less_o3 <- get_component_plot(H_long_5c_less_o3,
                                              '3', '3) Oil & gas fugitive and venting emssions')
nmfplt_4_svd_5c_less_o3 <- get_component_plot(H_long_5c_less_o3,
                                              '4', '4) Flaring, incomplete combustion')
nmfplt_5_svd_5c_less_o3 <- get_component_plot(H_long_5c_less_o3,
                                              '5', '5) Flaring, sour gas')
```



Fingerprint plot

```
custom_colors <- c("Traffic" = "#00AFBB",
                   "Combustion" = "#E7B800",
                   "Oil & Gas Emissions" = "#FC4E07",
                   "Flare (factor 4)" = "#0072B2",
                   "Flare (factor 5)" = "#8B4513")

contrib_prop <- apply(H_df_5c_less_o3[,1:(length(H_df_5c_less_o3)-1)], MARGIN = 2,
                      FUN = function(x) {x/sum(x)})

contrib_prop %>%
  as_tibble() %>%
  mutate(Component = c('Traffic', 'Combustion', 'Oil & Gas Emissions', 'Flare (factor 4)', 'Flare (facto
  mutate(Component = factor(Component, levels = c('Traffic', 'Combustion', 'Oil & Gas Emissions', 'Flare
  pivot_longer(cols = -Component, names_to = "Chemical", values_to = "Contribution_prop") %>%
```

```
  mutate(Chemical = factor(Chemical, levels = desired_order)) %>%
  ggplot(aes(fill = Component, y = Contribution_prop, x = Chemical)) +
  geom_bar(position = "fill", stat = "identity") +
  scale_fill_manual(values = custom_colors) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Chemical", y = "Contribution Proportion") +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank())
```



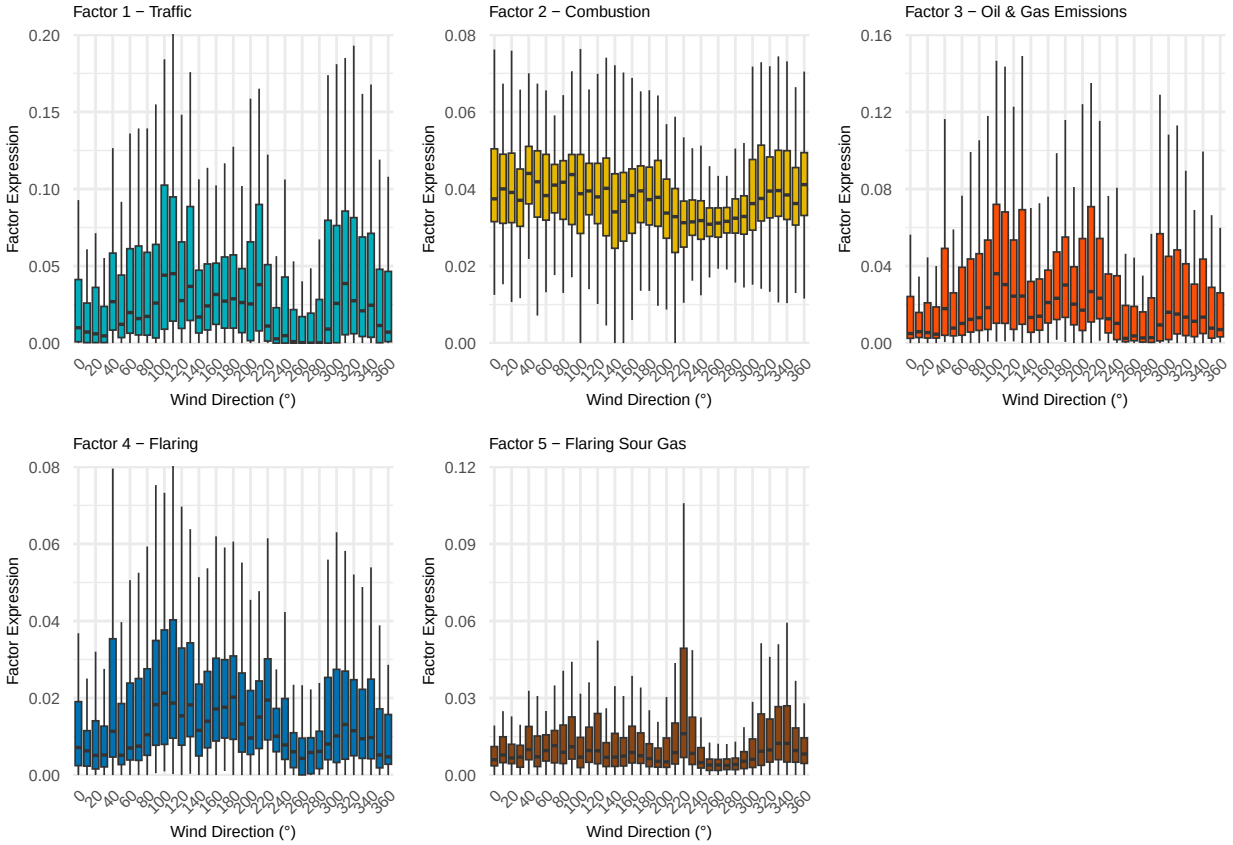

```
#ggsave("fingerprint.png", c)
```

Wind plots

```
hourly_wind_nona <- hourly_nona %>%
  select(wdr_deg, wsp_ms)

data_to_plot <- tibble(
  component1 = basis(nmf_result_5c_less_o3)[,1],
  component2 = basis(nmf_result_5c_less_o3)[,2],
  component3 = basis(nmf_result_5c_less_o3)[,3],
  component4 = basis(nmf_result_5c_less_o3)[,4],
  component5 = basis(nmf_result_5c_less_o3)[,5],
  wd = round(hourly_wind_nona$wdr_deg, -1)
)
```

```r
color_pal <- c("#00AFBB", "#E7B800", "#FC4E07", "#0072B2", "#8B4513")


data_long <- data_to_plot %>%
  pivot_longer(cols = starts_with("component"), names_to = "Factor", values_to = "Expression")


factor_labels <- c(
  "component1" = "Factor 1 - Traffic",
  "component2" = "Factor 2 - Combustion",
  "component3" = "Factor 3 - Oil & Gas Emissions",
  "component4" = "Factor 4 - Flaring",
  "component5" = "Factor 5 - Flaring Sour Gas"
)


data_long <- data_long %>%
  mutate(wd = factor(wd, levels = sort(unique(wd))))

# Select every second wind direction for labeling
every_second_label <- levels(data_long$wd)[seq(1, length(levels(data_long$wd)), by = 2)]


y_axis_limits <- list(
  "component1" = c(0, 0.2),
  "component2" = c(0, 0.08),
  "component3" = c(0, 0.16),
  "component4" = c(0, 0.08),
  "component5" = c(0, 0.12)
)


plots <- lapply(1:5, function(i) {
  factor_name <- paste0("component", i)

  ggplot(data_long %>% filter(Factor == factor_name),
         aes(x = wd, y = Expression, fill = as.factor(wd))) +
    geom_boxplot(outliers=F, size=0.3) +
    scale_fill_manual(values = rep(color_pal[i], length(unique(data_long$wd)))) +
    scale_x_discrete(breaks = every_second_label) +
    coord_cartesian(ylim = y_axis_limits[[factor_name]]) +
    scale_y_continuous(
      limits = c(0, NA),
      breaks = seq(0, y_axis_limits[[factor_name]][2], length.out = 5) ,
      expand=expansion(mult=c(0))
    ) +
    labs(title = factor_labels[factor_name],
         x = "Wind Direction (°)",
         y = "Factor Expression") +
    theme_minimal() +
    theme(
      legend.position = "none",
      plot.title = element_text(size = 6),  # Smaller title text
```

```
      axis.title = element_text(size = 6),  # Smaller axis labels
      axis.text = element_text(size = 6),  # Smaller x and y tick labels
      axis.text.x = element_text(angle = 45, hjust = 1)
    )
})


grid.arrange(grobs = plots, ncol = 3)
```




```
# ggsave("factors-wind.png", w)
```

Factor analysis

- merge in factors 1-5 to dataset (hourly)

```
# First look at how well this approximates
fitted_5c_less_o3 <- fitted(nmf_result_5c_less_o3)
sum(abs(normalized_matrix_less_o3-fitted_5c_less_o3))
```

```
## [1] 1059.63
```

```
# NMF factorizes V = WH
# Store Basis matrix (W) and Coef Matrix (H)
saveRDS(basis_matrix_5c_less_o3, 'result_rfiles/nmf_norm_5c_less_o3_basis.rds')
saveRDS(coef_matrix_5c_less_o3, 'result_rfiles/nmf_norm_5c_less_o3_coef.rds')

# Merge basis matrix into hourly observations
```

```r
basis_matrix_5c_less_o3 <- as_tibble(basis_matrix_5c_less_o3) %>%
  setNames(c('Factor1', 'Factor2', 'Factor3', 'Factor4', 'Factor5'))
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
## `.name_repair` is omitted as of tibble 2.0.0.
## i Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
normalized_hourly_data_5c_less_o3 <- hourly_nona[,c('day', 'time_utc')] %>%
  cbind(normalized_matrix_less_o3) %>%
  cbind(basis_matrix_5c_less_o3) %>%
  right_join(hourly_data %>% select(-'day'), join_by(time_utc), suffix = c('_norm', ''))

# saveRDS(normalized_hourly_data_5c_less_o3,
#  'result_rfiles/normalized_hourly_data_5c_less_o3.rds')
normalized_hourly_data_5c_less_o3 <- readRDS('result_rfiles/normalized_hourly_data_5c_less_o3.rds')
```

- make daily dataset for VNF analysis
- compute wind directions from plots

```r
# Also compute a daily dataset
normalized_daily_data_5c_less_o3 <- normalized_hourly_data_5c_less_o3 %>%
  group_by(day) %>%
  summarise(across(where(is.numeric) & !any_of('wdr_deg'), ~ mean(.x, na.rm = T)),
            wdr_deg = as.numeric(mean(circular(wdr_deg, units = "degrees"), na.rm = T))) %>%
  mutate(wdr_deg = if_else(wdr_deg < 0, wdr_deg+360, wdr_deg)) %>%
  mutate(wind_45_135 = wdr_deg >= 45 & wdr_deg < 135,
         wind_135_180 = wdr_deg >= 135 & wdr_deg < 180,
         wind_180_270 = wdr_deg >= 180 & wdr_deg < 270,
         wind_270_45 = wdr_deg >= 270 & wdr_deg < 45)

# saveRDS(normalized_daily_data_5c_less_o3,
#  'result_rfiles/normalized_daily_data_5c_less_o3.rds')

normalized_daily_data_5c_less_o3 <-
  readRDS('result_rfiles/normalized_daily_data_5c_less_o3.rds')
```

- 1) number of flares in 100km of trailer associated with NMF
- 2) weighted count based on distance to trailer

```r
# Check if relationship between # flares and flare factor (4 & 5)
# Linear model
flare_factor <- lm(n_flare_100 ~ Factor1 + Factor2 + Factor3 + Factor4 + Factor5,
                   data = normalized_daily_data_5c_less_o3)
summary(flare_factor)
```

```
##
## Call:
## lm(formula = n_flare_100 ~ Factor1 + Factor2 + Factor3 + Factor4 +
##     Factor5, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.638 -22.160   4.205  18.488  76.270
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   52.438      7.455   7.034 1.61e-11 ***
## Factor1      -27.402    106.172  -0.258   0.7965
## Factor2     -338.560    196.573  -1.722   0.0861 .
## Factor3      286.534    151.310   1.894   0.0593 .
## Factor4     -287.536    244.717  -1.175   0.2410
## Factor5      231.978    212.510   1.092   0.2760
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.79 on 273 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.03877,    Adjusted R-squared:  0.02117
## F-statistic: 2.202 on 5 and 273 DF,  p-value: 0.05434
```

```
flare_factor45 <- lm(n_flare_100 ~ Factor4 + Factor5, data = normalized_daily_data_5c_less_o3)
summary(flare_factor45)
```

```
##
## Call:
## lm(formula = n_flare_100 ~ Factor4 + Factor5, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.409 -23.830   5.588  18.235  77.131
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   40.780      3.386  12.044   <2e-16 ***
## Factor4      -48.431    150.383  -0.322   0.7477
## Factor5      360.559    206.393   1.747   0.0818 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.02 on 276 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.01171,    Adjusted R-squared:  0.004548
## F-statistic: 1.635 on 2 and 276 DF,  p-value: 0.1968
```

```
flare_factor_weighted <- lm(weighted.count ~ Factor1 + Factor2 + Factor3 + Factor4 + Factor5,
                            data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted)
```

```
##
## Call:
## lm(formula = weighted.count ~ Factor1 + Factor2 + Factor3 + Factor4 +
##     Factor5, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3295 -0.2180  0.0546  0.3809  3.9848
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2572     0.2278   9.907   <2e-16 ***
## Factor1       0.1244     3.2450   0.038   0.9694
## Factor2      -4.7740     6.0080  -0.795   0.4275
## Factor3       7.4339     4.6246   1.607   0.1091
## Factor4     -12.9155     7.4794  -1.727   0.0853 .
## Factor5       4.0762     6.4951   0.628   0.5308
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8492 on 273 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.02221,    Adjusted R-squared:  0.0043
## F-statistic:  1.24 on 5 and 273 DF,  p-value: 0.2905
```

```
flare_factor_weighted45 <- lm(weighted.count ~ Factor4 + Factor5,
                              data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted45)
```

```
##
## Call:
## lm(formula = weighted.count ~ Factor4 + Factor5, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2558 -0.1821  0.0775  0.3622  3.9366
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0910     0.1029  20.315   <2e-16 ***
## Factor4      -4.0144     4.5712  -0.878    0.381
## Factor5       7.3663     6.2738   1.174    0.241
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8518 on 276 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.005595,   Adjusted R-squared:  -0.001611
## F-statistic: 0.7765 on 2 and 276 DF,  p-value: 0.461
```

```
# All factors + wind speed + wind direction + factor5:sw wind.
# Wind direction from 270 to 45 is left as reference group.
flare_factor_weighted_2 <- lm(weighted.count ~ Factor1 + Factor2 + Factor3 +
                                Factor4 + Factor5 + wsp_ms + wind_45_135 +
                                wind_135_180 + Factor5*wind_180_270,
                              data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_2)
```

```
##
## Call:
## lm(formula = weighted.count ~ Factor1 + Factor2 + Factor3 + Factor4 +
##     Factor5 + wsp_ms + wind_45_135 + wind_135_180 + Factor5 *
##     wind_180_270, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.4382 -0.2036  0.0782  0.3578  3.9528
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.14418    0.33390   6.422 6.11e-10 ***
## Factor1                     0.07365    3.35780   0.022   0.9825
## Factor2                    -4.92773    6.19462  -0.795   0.4270
## Factor3                     8.99603    4.80767   1.871   0.0624 .
## Factor4                   -10.66714    7.82960  -1.362   0.1742
## Factor5                     3.28556    7.52447   0.437   0.6627
## wsp_ms                      0.04430    0.04523   0.980   0.3282
## wind_45_135TRUE            -0.15557    0.17743  -0.877   0.3814
## wind_135_180TRUE           -0.15153    0.13270  -1.142   0.2545
## wind_180_270TRUE           -0.21667    0.22628  -0.958   0.3392
## Factor5:wind_180_270TRUE    2.51821   13.08348   0.192   0.8475
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8524 on 268 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.03302,    Adjusted R-squared:  -0.003061
## F-statistic: 0.9152 on 10 and 268 DF,  p-value: 0.5196
```

```r
# Same as above but only factor 4 and 5
flare_factor_weighted_3 <- lm(weighted.count ~ Factor4 + Factor5 + wsp_ms +
                              Factor5*wind_180_270,
                            data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_3)
```

```
##
## Call:
## lm(formula = weighted.count ~ Factor4 + Factor5 + wsp_ms + Factor5 *
##     wind_180_270, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2706 -0.1961  0.0714  0.3721  3.9358
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.057728   0.222033   9.268   <2e-16 ***
## Factor4                   -3.273445   5.155494  -0.635    0.526
## Factor5                    7.499668   7.297235   1.028    0.305
## wsp_ms                     0.009441   0.040855   0.231    0.817
## wind_180_270TRUE          -0.059345   0.215638  -0.275    0.783
## Factor5:wind_180_270TRUE -0.078349  12.897092  -0.006    0.995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.856 on 273 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.006554,   Adjusted R-squared:  -0.01164
## F-statistic: 0.3602 on 5 and 273 DF,  p-value: 0.8754
```

```r
# Same as above but interaction between factor 4 and SW wind
flare_factor_weighted_3b <- lm(weighted.count ~ Factor4 + Factor5 + wsp_ms +
                                 Factor4*wind_180_270,
                               data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_3b)
```

```
##
## Call:
## lm(formula = weighted.count ~ Factor4 + Factor5 + wsp_ms + Factor4 *
##     wind_180_270, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3058 -0.2123  0.0650  0.3774  3.9523
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  2.030311   0.215853   9.406   <2e-16 ***
## Factor4                     -0.841684   5.604781  -0.150    0.881
## Factor5                      7.401203   6.365967   1.163    0.246
## wsp_ms                       0.005478   0.040870   0.134    0.893
## wind_180_270TRUE             0.143565   0.224432   0.640    0.523
## Factor4:wind_180_270TRUE   -10.510472   9.618074  -1.093    0.275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8541 on 273 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.01088,    Adjusted R-squared:  -0.007235
## F-statistic: 0.6006 on 5 and 273 DF,  p-value: 0.6995
```

```r
# Same as above but with East wind
flare_factor_weighted_3c <- lm(weighted.count ~ Factor4 + Factor5 + wsp_ms +
                                 Factor5*wind_45_135,
                               data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_3c)
```

```
##
## Call:
## lm(formula = weighted.count ~ Factor4 + Factor5 + wsp_ms + Factor5 *
##     wind_45_135, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2355 -0.1832  0.0761  0.3768  3.9129
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.1177298  0.2207356   9.594   <2e-16 ***
## Factor4                    -4.0986004  5.1120629  -0.802    0.423
## Factor5                     6.1846170  6.5540446   0.944    0.346
## wsp_ms                      0.0008777  0.0407876   0.022    0.983
## wind_45_135TRUE            -0.4275903  0.3752245  -1.140    0.255
## Factor5:wind_45_135TRUE 22.0187173 23.3341439   0.944    0.346
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8543 on 273 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.01053,    Adjusted R-squared:  -0.007588
## F-statistic: 0.5813 on 5 and 273 DF,  p-value: 0.7143
```

```
flare_factor_weighted_3d <- lm(weighted.count ~ Factor4 + Factor5 + wsp_ms +
                                 Factor4*wind_45_135,
                               data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_3d)
```

```
##
## Call:
## lm(formula = weighted.count ~ Factor4 + Factor5 + wsp_ms + Factor4 *
##     wind_45_135, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2680 -0.1822  0.0707  0.3665  3.9260
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.09473    0.21997   9.523   <2e-16 ***
## Factor4                  -4.50558    5.36882  -0.839    0.402
## Factor5                   7.78444    6.38818   1.219    0.224
## wsp_ms                    0.00343    0.04074   0.084    0.933
## wind_45_135TRUE          -0.19324    0.28449  -0.679    0.498
## Factor4:wind_45_135TRUE   4.84677   13.38860   0.362    0.718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8555 on 273 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.007783,   Adjusted R-squared:  -0.01039
## F-statistic: 0.4283 on 5 and 273 DF,  p-value: 0.8288
```

```
# Wind speed + factor 4 and interaction with East wind
flare_factor_weighted_4a <- lm(weighted.count ~ wsp_ms + Factor4*wind_45_135,
                               data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_4a)
```

```
##
## Call:
## lm(formula = weighted.count ~ wsp_ms + Factor4 * wind_45_135,
##     data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1668 -0.1983  0.0661  0.3882  3.8663
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.187226   0.206642  10.585   <2e-16 ***
## wsp_ms                  -0.003785   0.040343  -0.094    0.925
```

```
## Factor4                 -2.428778   5.095632  -0.477    0.634
## wind_45_135TRUE         -0.171010   0.284161  -0.602    0.548
## Factor4:wind_45_135TRUE  3.879541  13.376874   0.290    0.772
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8562 on 274 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.002386,   Adjusted R-squared:  -0.01218
## F-statistic: 0.1638 on 4 and 274 DF,  p-value: 0.9565
```

```
# Wind speed + factor 4 and interaction with SE wind
flare_factor_weighted_4b <- lm(weighted.count ~ wsp_ms + Factor4*wind_135_180,
                               data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_4b)
```

```
##
## Call:
## lm(formula = weighted.count ~ wsp_ms + Factor4 * wind_135_180,
##     data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2269 -0.2186  0.0794  0.3693  3.8404
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.229232   0.202818  10.991   <2e-16 ***
## wsp_ms                     0.001297   0.039954   0.032   0.9741
## Factor4                   -5.716640   5.512131  -1.037   0.3006
## wind_135_180TRUE          -0.440258   0.229503  -1.918   0.0561 .
## Factor4:wind_135_180TRUE 18.354002   9.757795   1.881   0.0610 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.851 on 274 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.01456,    Adjusted R-squared:  0.0001782
## F-statistic: 1.012 on 4 and 274 DF,  p-value: 0.4014
```

```
# Wind speed + factor 4 and interaction with SW wind
flare_factor_weighted_4c <- lm(weighted.count ~ wsp_ms + Factor4*wind_180_270,
                               data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_4c)
```

```
##
## Call:
## lm(formula = weighted.count ~ wsp_ms + Factor4 * wind_180_270,
##     data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1911 -0.1952  0.0539  0.4087  3.8937
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                2.121372   0.201270  10.540   <2e-16 ***
## wsp_ms                    -0.001567   0.040444  -0.039    0.969
## Factor4                    1.058349   5.364665   0.197    0.844
## wind_180_270TRUE           0.143445   0.224576   0.639    0.524
## Factor4:wind_180_270TRUE -10.634150   9.623656  -1.105    0.270
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8547 on 274 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.005983,   Adjusted R-squared:  -0.008528
## F-statistic: 0.4123 on 4 and 274 DF,  p-value: 0.7997
```

```
# Wind speed + factor 5 and interaction with East wind
flare_factor_weighted_5a <- lm(weighted.count ~ wsp_ms + Factor5*wind_45_135,
                               data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_5a)
```

```
##
## Call:
## lm(formula = weighted.count ~ wsp_ms + Factor5 * wind_45_135,
##     data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2299 -0.1901  0.0789  0.3725  3.9421
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.01637    0.18083  11.151   <2e-16 ***
## wsp_ms                    0.01530    0.03658   0.418    0.676
## Factor5                   4.62551    6.25483   0.740    0.460
## wind_45_135TRUE          -0.40644    0.37405  -1.087    0.278
## Factor5:wind_45_135TRUE 21.30900   23.30214   0.914    0.361
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8537 on 274 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.008204,   Adjusted R-squared:  -0.006275
## F-statistic: 0.5666 on 4 and 274 DF,  p-value: 0.6871
```

```
# Wind speed + factor 5 and interaction with SE wind
flare_factor_weighted_5b <- lm(weighted.count ~ wsp_ms + Factor5*wind_135_180,
                               data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_5b)
```

```
##
## Call:
## lm(formula = weighted.count ~ wsp_ms + Factor5 * wind_135_180,
##     data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2259 -0.1876  0.0693  0.3775  3.9290
##
```

```
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.02295    0.18889  10.710   <2e-16 ***
## wsp_ms                     0.01822    0.03643   0.500    0.618
## Factor5                    4.19811    7.25463   0.579    0.563
## wind_135_180TRUE          -0.15866    0.20317  -0.781    0.436
## Factor5:wind_135_180TRUE   6.06208   12.22399   0.496    0.620
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8545 on 274 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.006345,   Adjusted R-squared:  -0.008161
## F-statistic: 0.4374 on 4 and 274 DF,  p-value: 0.7816
```

```
# Wind speed + factor 5 and interaction with SW wind
flare_factor_weighted_5c <- lm(weighted.count ~ wsp_ms + Factor5*wind_180_270,
                               data = normalized_daily_data_5c_less_o3)
summary(flare_factor_weighted_5c)
```

```
##
## Call:
## lm(formula = weighted.count ~ wsp_ms + Factor5 * wind_180_270,
##     data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2696 -0.2009  0.0671  0.3748  3.9556
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.98021    0.18525  10.689   <2e-16 ***
## wsp_ms                     0.02098    0.03655   0.574    0.566
## Factor5                    6.25646    7.02199   0.891    0.374
## wind_180_270TRUE          -0.06960    0.21480  -0.324    0.746
## Factor5:wind_180_270TRUE -0.16687   12.88229  -0.013    0.990
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8551 on 274 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.005087,   Adjusted R-squared:  -0.009438
## F-statistic: 0.3502 on 4 and 274 DF,  p-value: 0.8438
```

```
# Check relationship between avg flare distance and flare factor (4 & 5)
# Linear model
flare_factor_dist <- lm(distToLovi ~ Factor4 + Factor5, data = normalized_daily_data_5c_less_o3)
summary(flare_factor_dist)
```

```
##
## Call:
## lm(formula = distToLovi ~ Factor4 + Factor5, data = normalized_daily_data_5c_less_o3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -34.449  -2.443  -0.139   2.266  31.399
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  54.7821     0.8756  62.564   <2e-16 ***
## Factor4       7.2656    39.7289   0.183    0.855
## Factor5      64.5797    52.2590   1.236    0.218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.883 on 252 degrees of freedom
##   (25 observations deleted due to missingness)
## Multiple R-squared:  0.008425,   Adjusted R-squared:  0.0005557
## F-statistic: 1.071 on 2 and 252 DF,  p-value: 0.3444
```