

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 $< 2 \times \text{LOD}$ and maximum value $> 100 \times \text{LOD}$)
3. Zero values are converted to a random value between 0 and $0.5 \times \text{LOD}$
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
hourly_vocs <- hourly_nona %>% select(any_of(vocs))

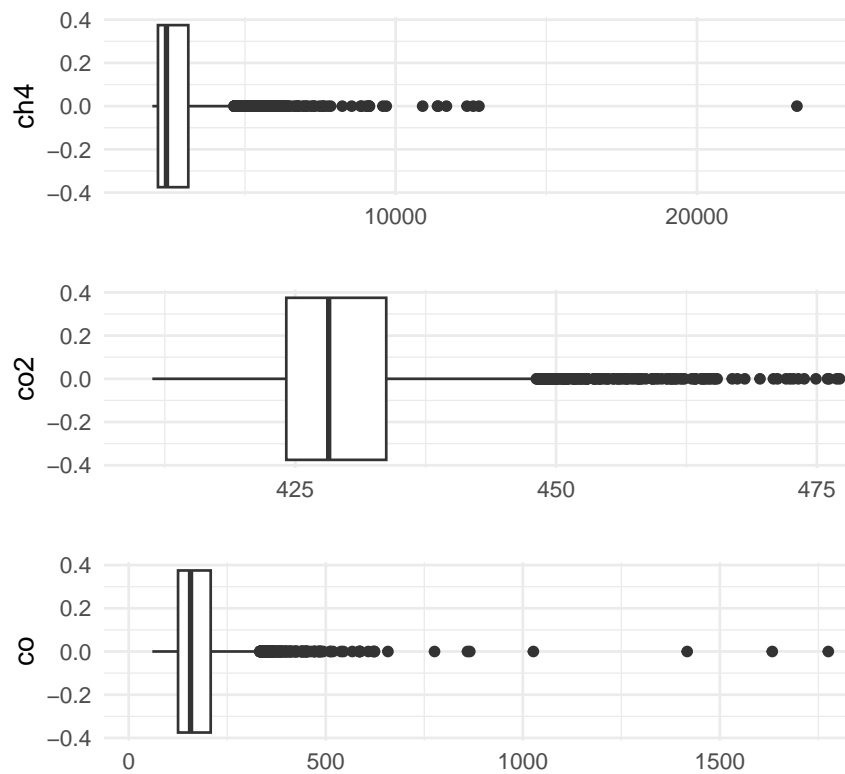
# retrieving the non-vocs: co2_ppm, nox, ch4, h2s, so2, o3
# double check this
hourly_non_vocs <- hourly_nona %>% select(any_of(non_vocs))

hourly_full_nona <- cbind(hourly_non_vocs, hourly_vocs)

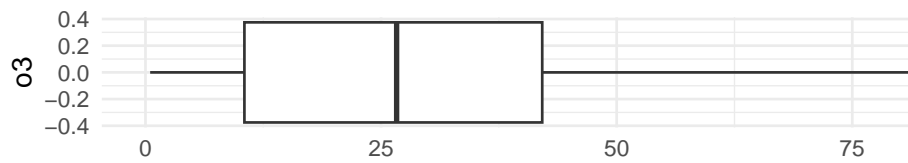
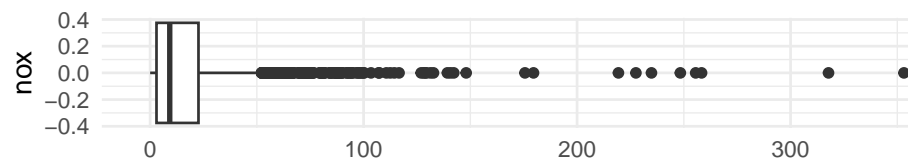
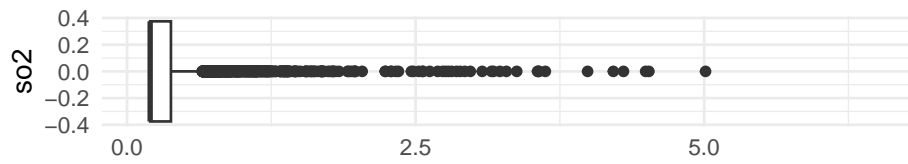
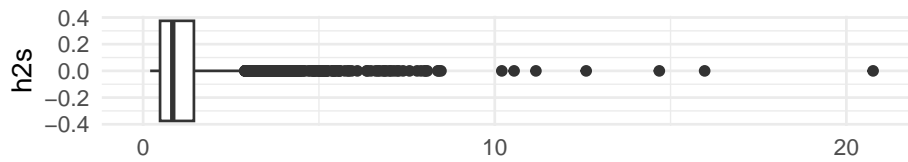
# retrieve a vector of yearmonth
hourly_dates <- hourly_nona %>%
  mutate(yearmonth = substring(day, 0, 7)) %>%
  pull(yearmonth)

```

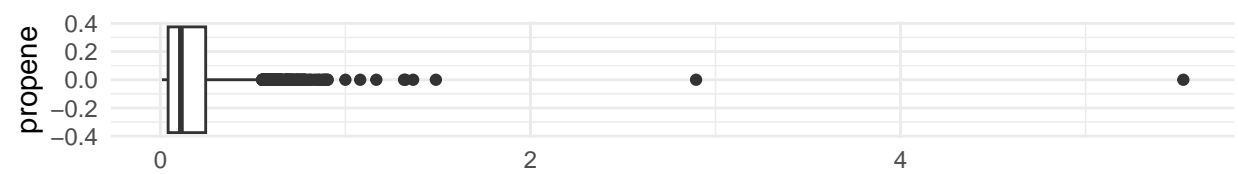
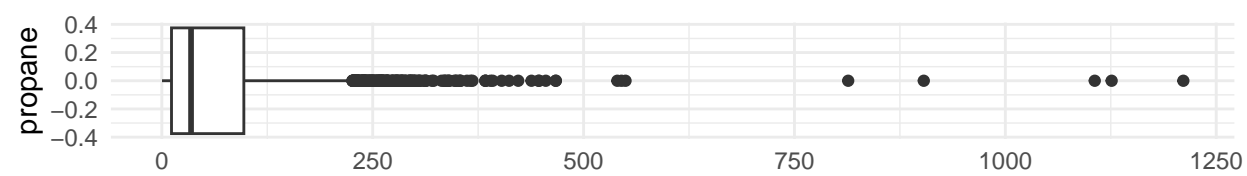
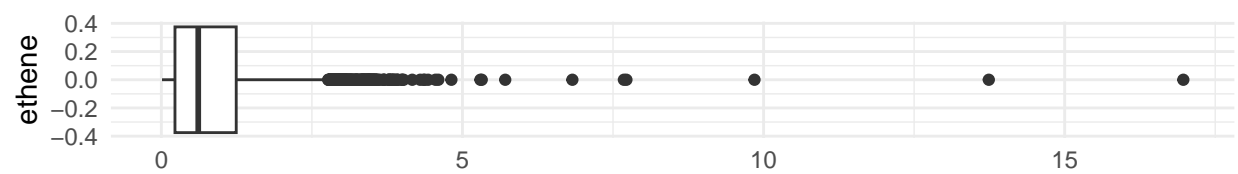
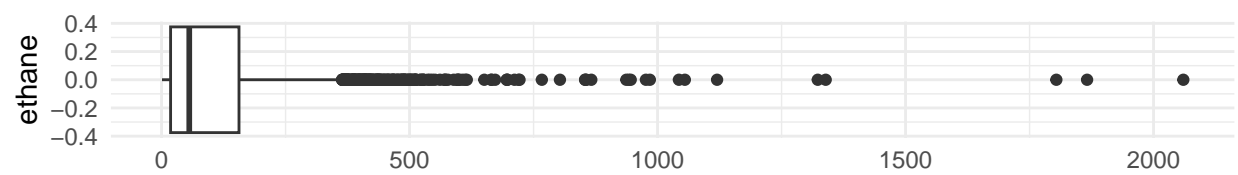
Data visualisation

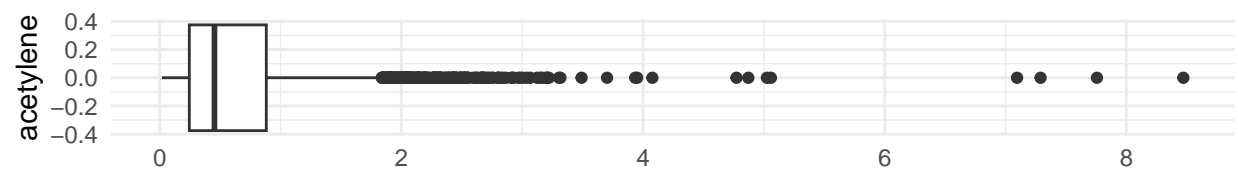
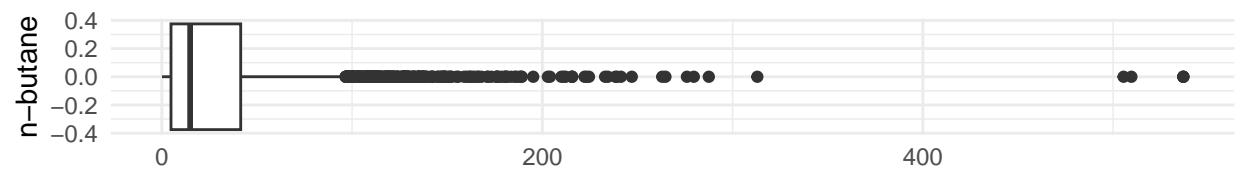
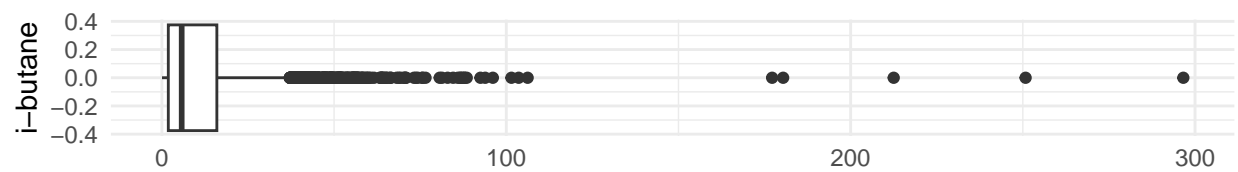
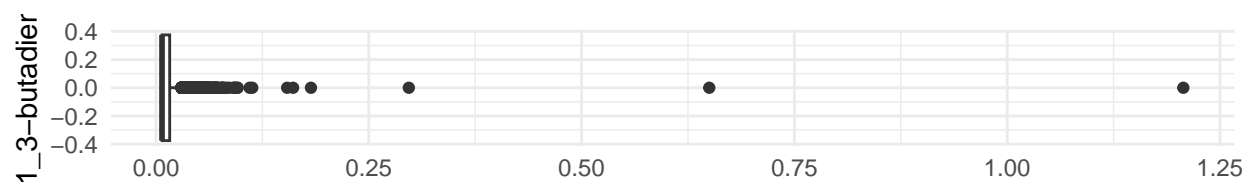


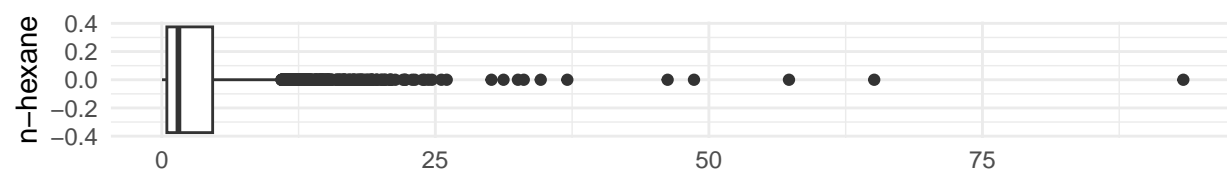
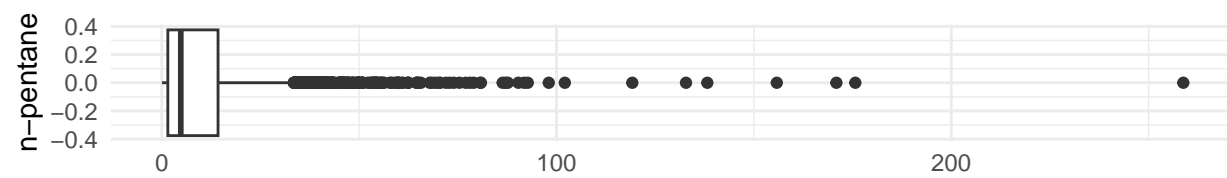
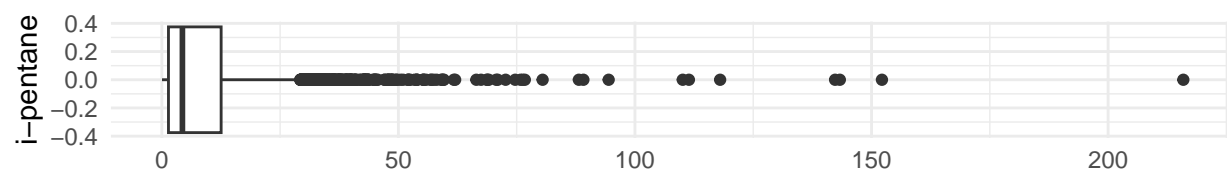
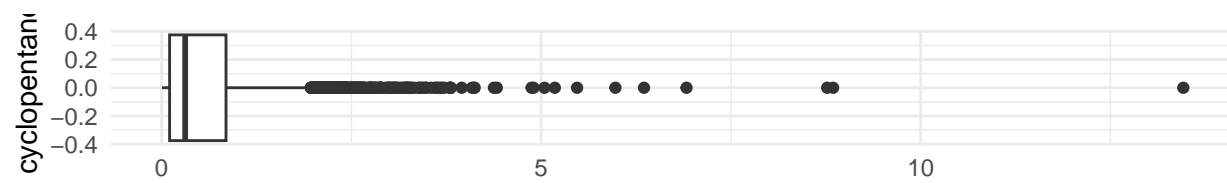
- Boxplots of the hourly concentrations non-voc

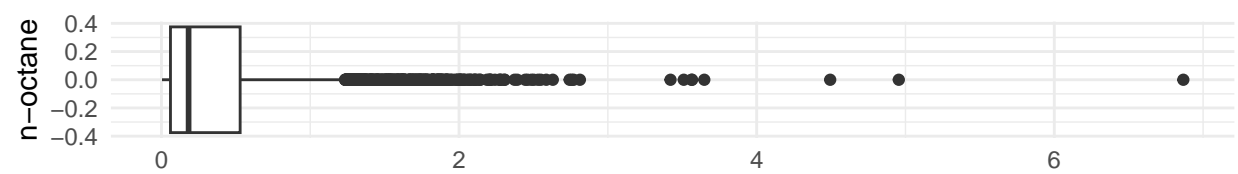
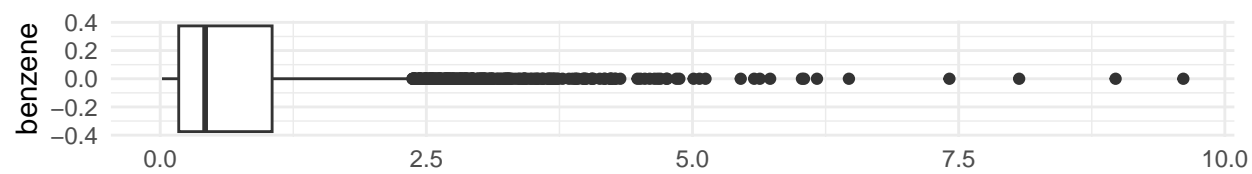
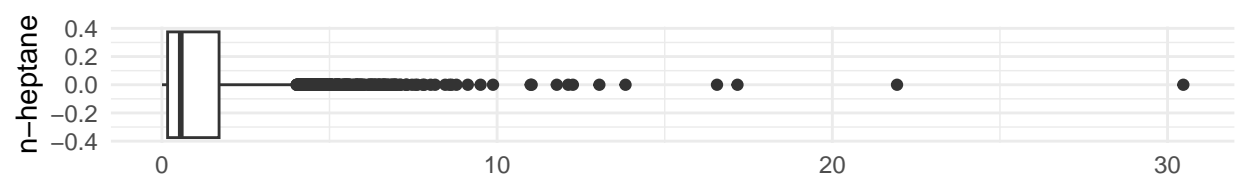
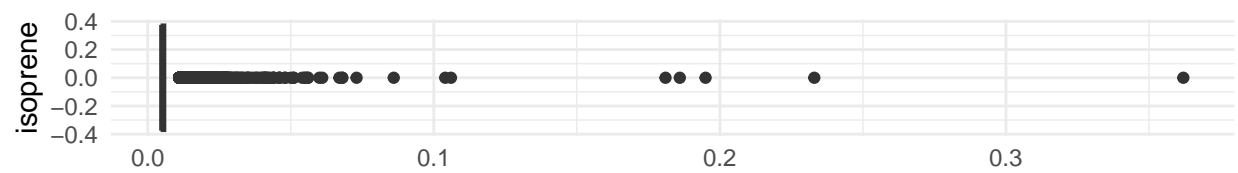


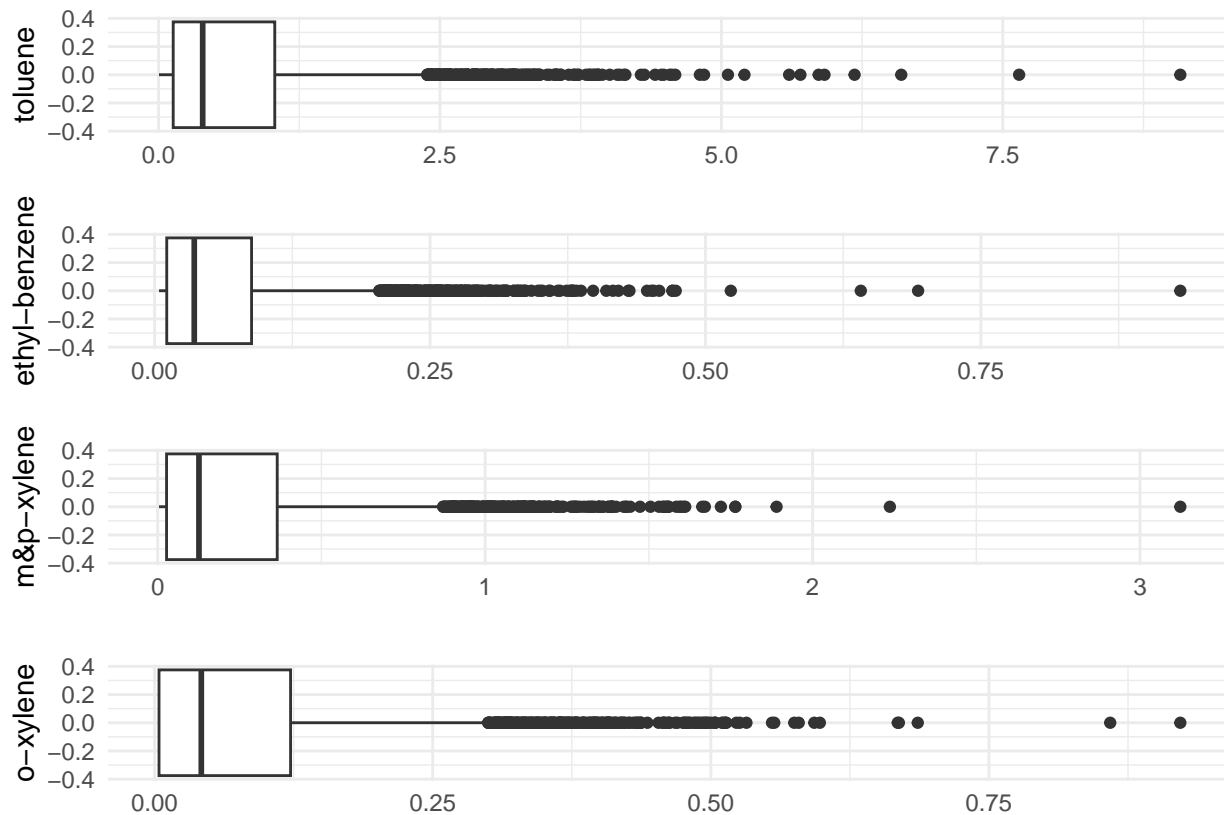
- Boxplots sulfur compounds, NOx, ozone
- Boxplots VOCs











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 = strptime(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

name	N	1st percentile	99th percentile	Max	Background	Background	#
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
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.   :0.000000 Min.   :0.0000 Min.   :0.00000 Min.   :0.00000
## 1st Qu.:0.005795 1st Qu.:0.1384 1st Qu.:0.02592 1st Qu.:0.01022
## Median :0.014603 Median :0.1823 Median :0.03884 Median :0.02335
## Mean   :0.026837 Mean   :0.2000 Mean   :0.04761 Mean   :0.03500
## 3rd Qu.:0.037200 3rd Qu.:0.2418 3rd Qu.:0.05970 3rd Qu.:0.04525
## Max.   :1.000000 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.:0.007997 1st Qu.:0.006534 1st Qu.:0.09747 1st Qu.:0.008386
## Median :0.016114 Median :0.020262 Median :0.25487 Median :0.026672
## Mean   :0.026320 Mean   :0.036440 Mean   :0.26676 Mean   :0.050993
## 3rd Qu.:0.023633 3rd Qu.:0.049978 3rd Qu.:0.40546 3rd Qu.:0.075376
```

##	Max. :1.000000	Max. :1.000000	Max. :1.000000	Max. :1.000000
## ethene		propane	propene	1_3-butadiene
## Min. :0.00000	Min. :0.000000	Min. :0.000000	Min. :0.000000	Min. :0.000000
## 1st Qu.:0.01268	1st Qu.:0.009285	1st Qu.:0.005979	1st Qu.:0.001667	
## Median :0.03547	Median :0.028411	Median :0.018482	Median :0.004167	
## Mean :0.05042	Mean :0.053805	Mean :0.028772	Mean :0.007368	
## 3rd Qu.:0.07266	3rd Qu.:0.080132	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. :0.000000	Min. :0.000000	Min. :0.000000	Min. :0.000000	
## 1st Qu.:0.006153	1st Qu.:0.008783	1st Qu.:0.02674	1st Qu.:0.007432	
## Median :0.019261	Median :0.027528	Median :0.05135	Median :0.022668	
## Mean :0.038384	Mean :0.054906	Mean :0.07436	Mean :0.043730	
## 3rd Qu.:0.053703	3rd Qu.:0.077047	3rd Qu.:0.10211	3rd Qu.:0.062653	
## Max. :1.000000	Max. :1.000000	Max. :1.000000	Max. :1.000000	
## i-pentane	n-pentane	n-hexane	isoprene	
## Min. :0.000000	Min. :0.000000	Min. :0.000000	Min. :0.000000	
## 1st Qu.:0.006293	1st Qu.:0.005681	1st Qu.:0.004725	1st Qu.:0.002801	
## Median :0.019932	Median :0.018371	Median :0.016060	Median :0.005602	
## Mean :0.041085	Mean :0.038859	Mean :0.035000	Mean :0.010304	
## 3rd Qu.:0.057848	3rd Qu.:0.054837	3rd Qu.:0.049564	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. :0.000000	Min. :0.000000	Min. :0.000000	
## 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. :1.000000	Max. :1.000000	Max. :1.000000	
## ethyl-benzene	m&p-xylene	o-xylene		
## Min. :0.000000	Min. :0.000000	Min. :0.000000		
## 1st Qu.:0.007551	1st Qu.:0.007374	1st Qu.:0.000000		
## 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. :1.000000		

- 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

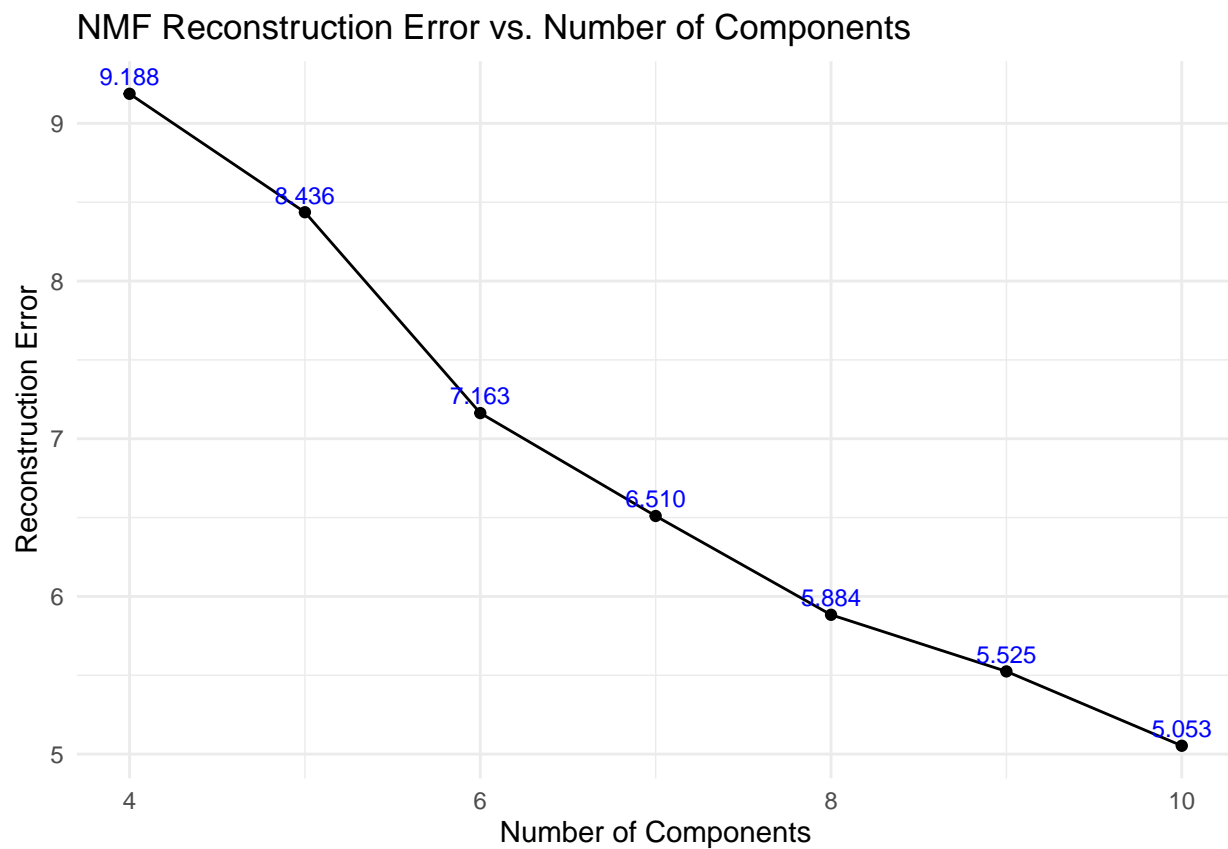
```
components <- 4:10
errors <- numeric(length(components) - 4)
```

```

# Loop over the number of components
# for (n in components) {
#   nmf_result <- nmf(normalized_matrix, rank = n, method = "KL", seed='nndsud')
#   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 deprecated
## Use c() or as.vector() instead.
```

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

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

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

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

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

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

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

```
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] 0.9212817
```

```
# 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_o3[, i, drop=FALSE])

# 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

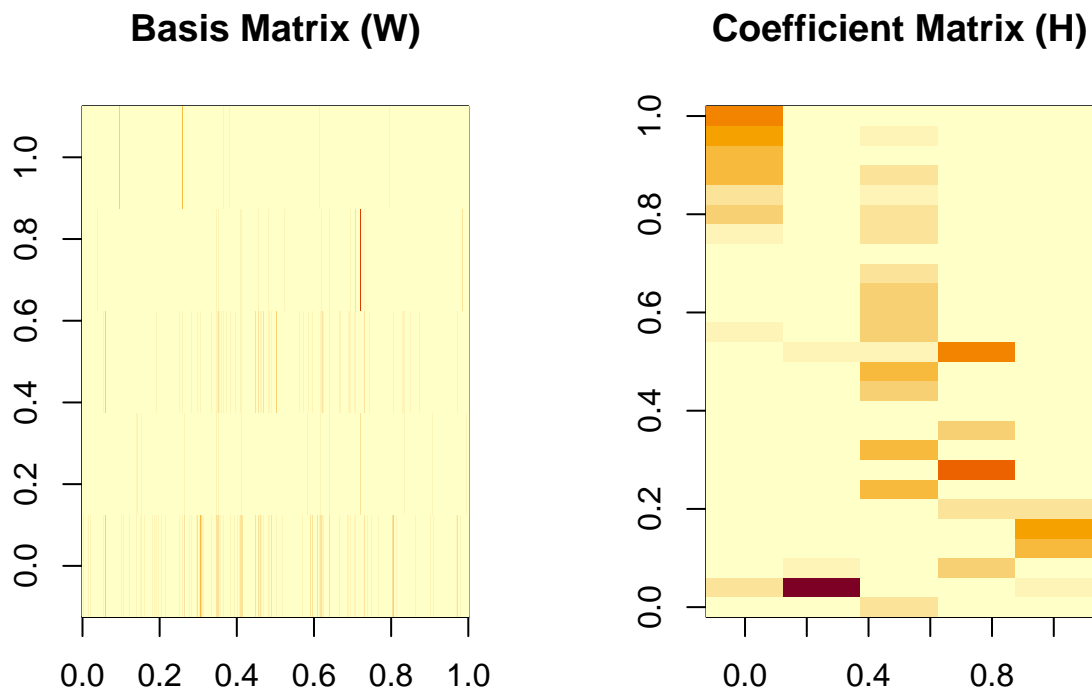
```

```
## [1] 0.2409233 0.5113765 0.8112270 0.8924548 0.9212817
```

```

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)

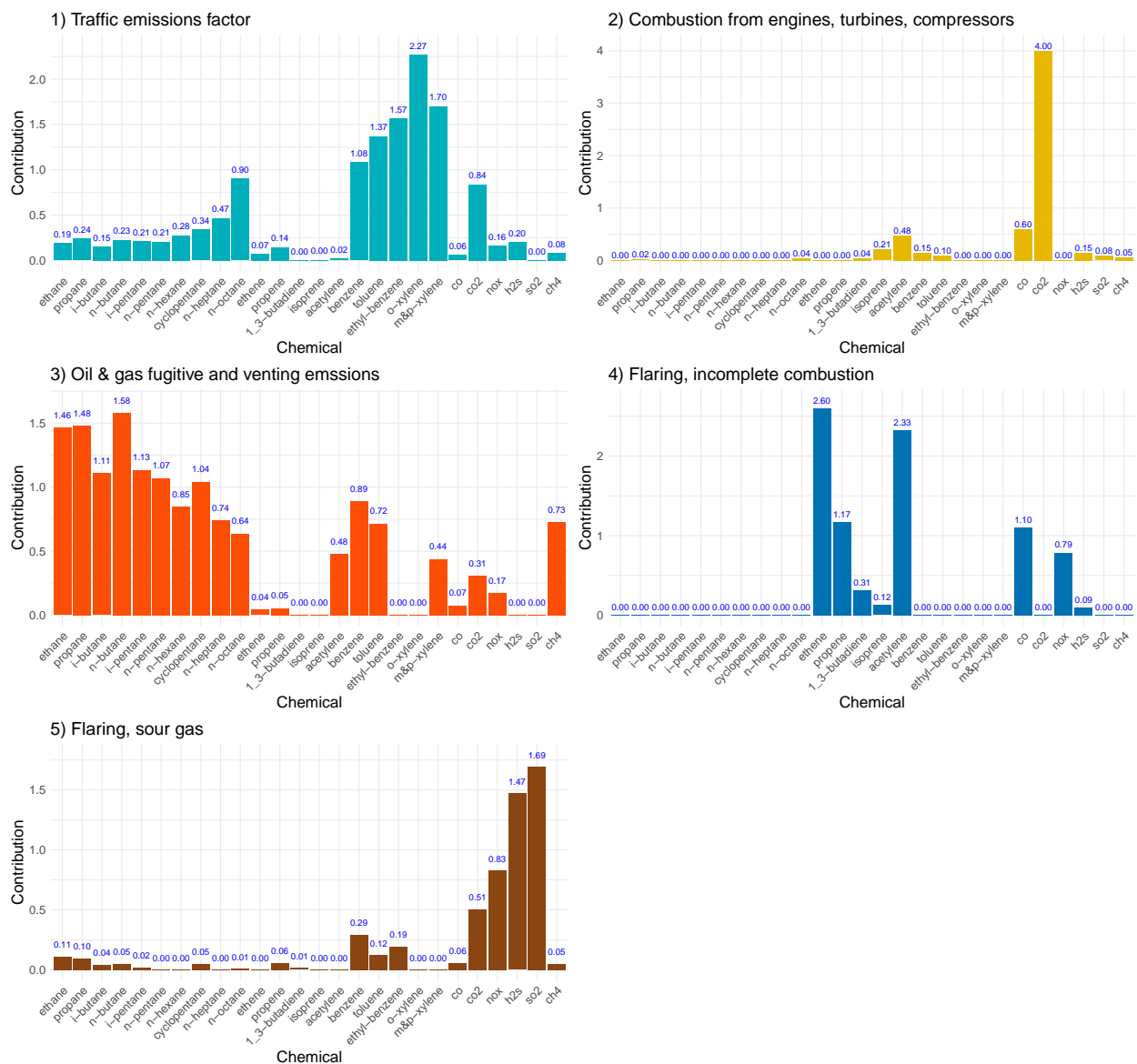
# Reshape data to long format

```



```
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')
```



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] 1060.414
```

```
# 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
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.
```

```
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')
```

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

```
# 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

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
## -5.7635 -3.0378 -0.4893  2.3031 16.8406
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.351      1.014   3.305 0.00108 **
## Factor1       -10.434     14.475  -0.721 0.47163
## Factor2         7.936     26.825   0.296 0.76756
## Factor3        36.265     20.638   1.757 0.08001 .
## Factor4       -28.511     33.444  -0.852 0.39469
## Factor5        37.042     28.700   1.291 0.19791
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.785 on 273 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.02441, Adjusted R-squared:  0.006544
## F-statistic: 1.366 on 5 and 273 DF, p-value: 0.2372
```

```
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
## -5.4658 -3.0946 -0.3795  2.2016 17.1266
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.592      0.456   7.878 7.71e-14 ***
## Factor4         6.625     20.357   0.325  0.745
## Factor5        42.500     27.706   1.534  0.126
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.787 on 276 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared: 0.01269, Adjusted R-squared: 0.005536
## F-statistic: 1.774 on 2 and 276 DF, p-value: 0.1716

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
## -10.369  -3.477  -0.572   2.114  117.655
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.165      2.232   4.107 5.3e-05 ***
## Factor1       -29.660     31.861  -0.931 0.35272
## Factor2      -121.718     59.043  -2.062 0.04020 *
## Factor3        20.457     45.425   0.450 0.65283
## Factor4       -43.619     73.613  -0.593 0.55398
## Factor5       188.812     63.171   2.989 0.00305 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.332 on 273 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared: 0.05585, Adjusted R-squared: 0.03856
## F-statistic: 3.23 on 5 and 273 DF, p-value: 0.007515

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
## -10.209  -3.167  -0.377   1.832  120.250
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.030      1.007   4.996 1.04e-06 ***
## Factor4       -103.752     44.944  -2.308 0.02171 *
## Factor5       193.540     61.168   3.164 0.00173 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 8.361 on 276 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared: 0.03869, Adjusted R-squared: 0.03173
## F-statistic: 5.554 on 2 and 276 DF, p-value: 0.004316

# 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
## -10.444  -3.183  -0.521   2.261  114.568
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10.3777      3.2634   3.180  0.00165 **
## Factor1         -28.7880     32.8916  -0.875  0.38223
## Factor2        -135.7806     60.7431  -2.235  0.02622 *
## Factor3           7.8931     47.1800   0.167  0.86726
## Factor4         -32.2435     77.0050  -0.419  0.67576
## Factor5         201.1532     73.2026   2.748  0.00641 **
## wsp_ms          -0.2740     0.4426  -0.619  0.53646
## wind_45_135TRUE    2.5387     1.7368   1.462  0.14498
## wind_135_180TRUE  -0.5723     1.2991  -0.441  0.65993
## wind_180_270TRUE   1.4875     2.1999   0.676  0.49953
## Factor5:wind_180_270TRUE -82.7076    126.3094  -0.655  0.51316
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.342 on 268 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared: 0.07083, Adjusted R-squared: 0.03616
## F-statistic: 2.043 on 10 and 268 DF, p-value: 0.02944
```

```
# 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
## -10.181  -3.098  -0.366   1.876  119.987
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.5004     2.1788   2.524  0.01216 *
## Factor4        -115.1950    50.6660  -2.274  0.02377 *
## Factor5         204.1482    71.2016   2.867  0.00446 **
## wsp_ms          -0.1641     0.4010  -0.409  0.68264
## wind_180_270TRUE    1.2950     2.0985   0.617  0.53769
## Factor5:wind_180_270TRUE -48.4824    124.5886  -0.389  0.69748
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.398 on 273 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.04064,    Adjusted R-squared:  0.02307
## F-statistic: 2.313 on 5 and 273 DF,  p-value: 0.04421
```

```
# 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
##  -9.978  -3.171  -0.290   1.841  120.163
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.7763     2.1217   2.723  0.00690 **
## Factor4        -120.3790    55.0563  -2.186  0.02963 *
## Factor5         190.5754    62.1541   3.066  0.00239 **
## wsp_ms          -0.1648     0.4021  -0.410  0.68227
## wind_180_270TRUE    0.2192     2.2108   0.099  0.92108
## Factor4:wind_180_270TRUE  21.2282    94.6402   0.224  0.82269
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.4 on 273 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.04028,    Adjusted R-squared:  0.0227
## F-statistic: 2.292 on 5 and 273 DF,  p-value: 0.046
```

```
# 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
	-22.708	-2.847	0.007	2.101	94.801

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.2565	1.9884	3.649	0.000315 ***
Factor4	-115.8172	46.1765	-2.508	0.012717 *
Factor5	88.8289	58.7340	1.512	0.131591
wsp_ms	-0.2622	0.3680	-0.713	0.476627
wind_45_135TRUE	-18.2763	3.3450	-5.464	1.05e-07 ***
Factor5:wind_45_135TRUE	1441.5995	206.5522	6.979	2.25e-11 ***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.704 on 273 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.1928, Adjusted R-squared:  0.178
## F-statistic: 13.04 on 5 and 273 DF, p-value: 2.174e-11
```

```
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
	-12.959	-3.123	-0.101	1.978	114.048

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.36016	2.12027	2.056	0.04069 *
Factor4	-62.16866	51.90733	-1.198	0.23208
Factor5	176.93325	61.20495	2.891	0.00415 **
wsp_ms	-0.04398	0.39320	-0.112	0.91102
wind_45_135TRUE	8.71066	2.74477	3.174	0.00168 **

```
## Factor4:wind_45_135TRUE -348.14280 128.95859 -2.700 0.00737 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.254 on 273 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.07347,    Adjusted R-squared:  0.0565
## F-statistic: 4.33 on 5 and 273 DF,  p-value: 0.0008292

# 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
	-12.378	-2.800	-0.039	1.756	117.438

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.4368	2.0215	3.184	0.001619 **
wsp_ms	-0.2062	0.3944	-0.523	0.601521
Factor4	-13.6951	49.7795	-0.275	0.783435
wind_45_135TRUE	9.2438	2.7751	3.331	0.000984 ***
Factor4:wind_45_135TRUE	-371.1890	130.4285	-2.846	0.004763 **

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.364 on 274 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.04511,    Adjusted R-squared:  0.03117
## F-statistic: 3.236 on 4 and 274 DF,  p-value: 0.01288
```

```
# 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
	-7.276	-2.666	-0.108	1.507	123.726

```
##
## Coefficients:
```



```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      8.4605     2.0284   4.171 4.07e-05 ***
## wsp_ms           -0.2906     0.3993  -0.728   0.467
## Factor4          -75.7741    55.0513  -1.376   0.170
## wind_135_180TRUE  -3.2162     2.2939  -1.402   0.162
## Factor4:wind_135_180TRUE 97.7109    97.3642   1.004   0.316
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.499 on 274 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.01397, Adjusted R-squared:  -0.0004239
## F-statistic: 0.9705 on 4 and 274 DF, p-value: 0.424

# 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
## -7.001  -2.990  -0.227   1.583  124.220
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      8.1106     2.0105   4.034 7.11e-05 ***
## wsp_ms           -0.3435     0.4039  -0.850   0.396
## Factor4          -71.2477    53.4743  -1.332   0.184
## wind_180_270TRUE   0.1415     2.2443   0.063   0.950
## Factor4:wind_180_270TRUE 22.2221    96.0796   0.231   0.817
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.528 on 274 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.007232, Adjusted R-squared:  -0.007261
## F-statistic: 0.499 on 4 and 274 DF, p-value: 0.7365
```

```
# 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
## -23.857  -2.640   0.193   1.842  97.377
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.3974     1.6449   2.673  0.00796 **
## wsp_ms           0.1450     0.3334   0.435  0.66396
## Factor5          44.2960    56.5239   0.784  0.43391
## wind_45_135TRUE  -17.7090     3.3694  -5.256 2.97e-07 ***
## Factor5:wind_45_135TRUE 1423.9509   208.4158   6.832 5.40e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.778 on 274 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1621
## F-statistic: 14.45 on 4 and 274 DF,  p-value: 1.022e-10

# 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
## -11.230  -3.033  -0.247   1.607  121.106
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.6160     1.8517   1.413  0.15887
## wsp_ms           0.2543     0.3583   0.710  0.47835
## Factor5          205.6894    70.6049   2.913  0.00387 **
## wind_135_180TRUE   0.9276     1.9887   0.466  0.64128
## Factor5:wind_135_180TRUE -178.3060   119.1866  -1.496  0.13580
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.402 on 274 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.03637, Adjusted R-squared:  0.0223
## F-statistic: 2.585 on 4 and 274 DF,  p-value: 0.03737

# 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
## -9.586  -2.920  -0.146   1.662  122.420
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.7653     1.8304   1.511   0.1320
## wsp_ms           0.2419     0.3618   0.669   0.5043
## Factor5          160.6788    69.1065   2.325   0.0208 *
## wind_180_270TRUE    0.9646     2.1094   0.457   0.6478
## Factor5:wind_180_270TRUE -53.5432   125.5129  -0.427   0.6700
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.462 on 274 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.02247,    Adjusted R-squared:  0.008201
## F-statistic: 1.575 on 4 and 274 DF,  p-value: 0.1812

# 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
## -17.8872  -4.0924  -0.6397   3.1281  15.8871
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.3055     0.8902  22.809  <2e-16 ***
## Factor4      78.3034    40.2421   1.946   0.053 .
## Factor5     -61.7593    51.8998  -1.190   0.235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.536 on 218 degrees of freedom
## (59 observations deleted due to missingness)
## Multiple R-squared:  0.01769,    Adjusted R-squared:  0.008681
## F-statistic: 1.963 on 2 and 218 DF,  p-value: 0.1429
```