CEF_BTEX_EDA

Phill Pham

2024-01-12

Exploratory Data Analysis of BTEX data collected near CEF

Background: Compared to other air sampling studies, this dataset is unique due to the high density of air samplers situated within and around the Central Experimental Farm in Ottawa, Ontario. Unlike other studies, this provides higher data resolution, and we would be able to pinpoint the source of a pollutant. In this case, we are interested in nearest distance to gas stations, since gasoline evaporation is a source of BTEX (Benzene, Toluene, Xylene isomers) VOCs (Volatile Organic Compounds).

We are interested in: BTEX VOC concentrations differences between Fall and Winter, and the relationship between VOC concentrations and distance to the nearest gas station. We hypothesize that generally, BTEX VOC concentration decreases with increased distance from nearest gas station, since

Loading Data and Libraries, Set Global Variables

```r
setwd("C:/Users/PPHAM/OneDrive - HC-SC PHAC-ASPC/R_environments/HC_2023/CEF_BTEX")

library(tidyverse)
library(stringr)
library(flextable)
library(moments)
library(rstatix)
library(ggpubr)
library(readxl)

# change this variable to TRUE to write tables
write_tables = FALSE

# load data & remove data where voc data is missing (can
# easily do this by filtering out any VOC species by na
# values)
voc = readxl::read_xlsx(path = "data/EGROW_passive_for_analysis.xlsx",
    sheet = 2, col_names = T) %>%
    dplyr::filter(!is.na(Dichloromethane))

# list of btex species
voc_species = c("Dichloromethane", "Hexane", "Chloroform", "__2_Dichloroethane",
    "Benzene", "Trichloroethylene", "Toluene", "Tetrachloroethylene",
    "Ethylbenzene", "_m_p__Xylene", "o_Xylene", "Styrene", "Cumene",
    "a_Pinene", "__1_2_2_Tetrchloroethane", "n_Decane", "__3_5_Trimethylbenzene",
```

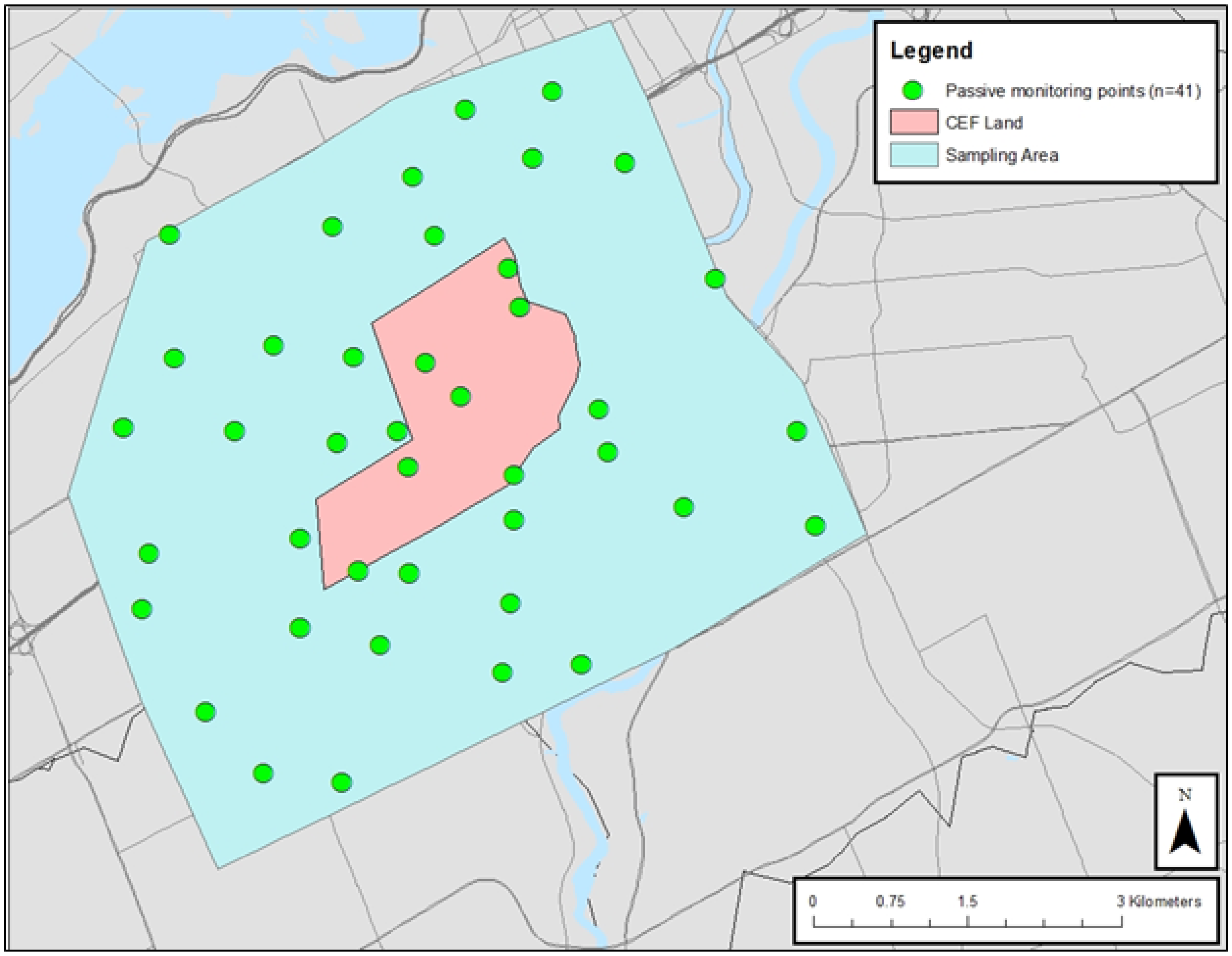


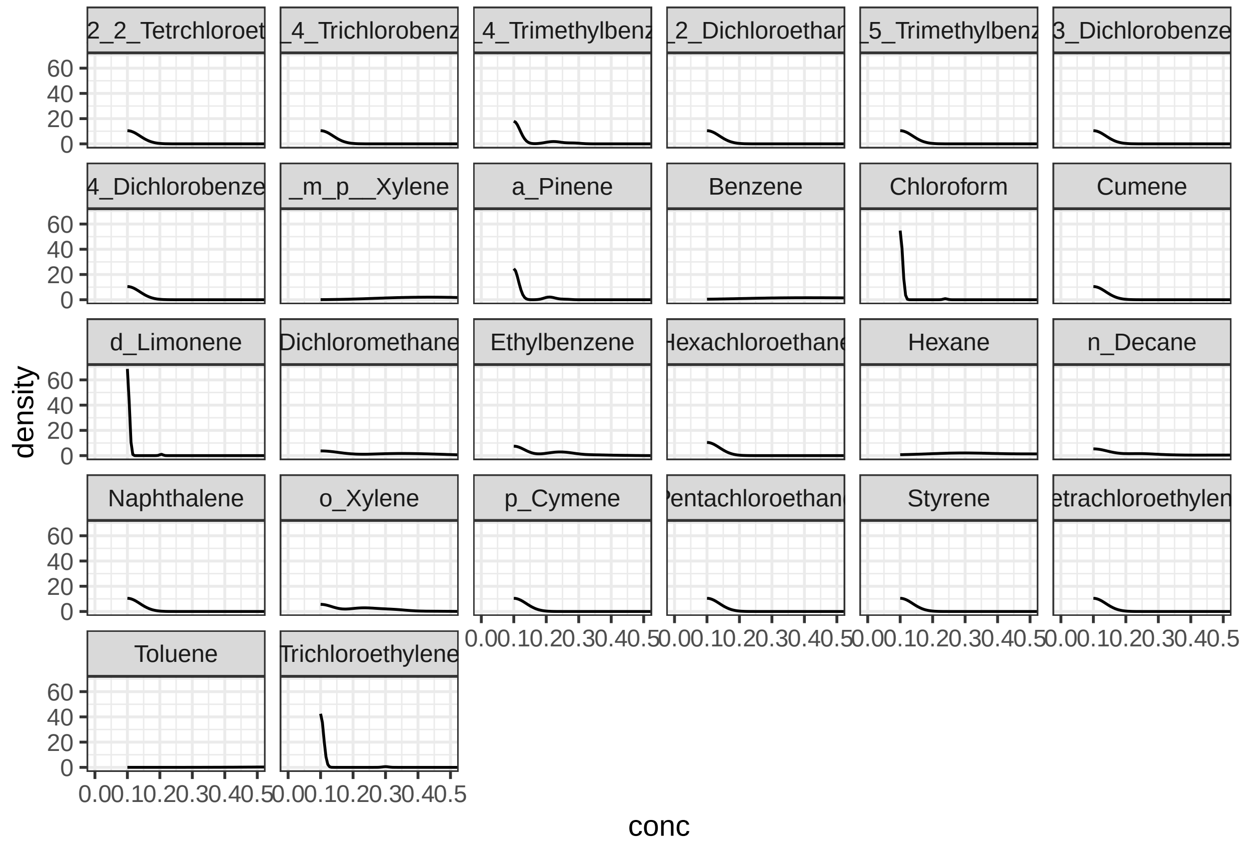
Figure 1: Locations of the 41 passive sampling points that collected measures for noise, NO2, and VOC in both fall and winter campaign.

```
    "__2_4_Trimethylbenzene", "Pentachloroethane", "d_Limonene",
    "p_Cymene", "__3_Dichlorobenzene", "__4_Dichlorobenzene",
    "Hexachloroethane", "__2_4_Trichlorobenzene", "Naphthalene")
```

Data wrangling + examination of data distribution

```
# convert data to long format - for calculations and
# plotting later
voc_long = voc %>%
    dplyr::select(season, all_of(voc_species)) %>%
    tidyr::pivot_longer(cols = all_of(voc_species), names_to = "species",
        values_to = "conc")

# examine data distribution - does not look normally
# distributed. Should use Dunn Test or other non parametric
# approaches
voc_long %>%
    ggplot(aes(x = conc, y = ..density..)) + geom_density() +
    facet_wrap(~species) + coord_cartesian(xlim = c(0, 0.5)) +
    theme_bw()
```



Generate Summary Statistics Table

```
# Calculate number of below detection limit (%BDL) using
# blank correction flags 1: observation < LDL 2: corrected
# observation < LDL 3: corrected observation > LDL but <
# FDL 4: corrected observation > FDL 5: uncorrected
# observation > LDL basically just looking for % of dataset
# being compared of _BDL = 1 or _BDL = 2

# Compute number of BDL (Below detection limit) by species
# and season
voc_BDL = voc %>%
    dplyr::select(season, contains("BDL")) %>%
    tidyr::pivot_longer(cols = all_of(contains("BDL")), names_to = "species",
        values_to = "flag") %>%
    dplyr::mutate(species = stringr::str_remove(string = species,
        pattern = "_BDL")) %>%
    dplyr::group_by(season, species) %>%
    dplyr::summarise(N = n(), n_BDL = sum(flag %in% c("1", "2")),
        `%_BDL` = (n_BDL/N) * 100) %>%
    dplyr::select(season, species, `%_BDL`)

# calculate P values for seasonality; P value adjustment
# not applicable since we're just looking at each compound
# individually
dunn_res = voc_long %>%
    dplyr::group_by(species) %>%
    rstatix::dunn_test(data = ., formula = as.formula("conc ~ season"),
        p.adjust.method = "none")

# format output dummy dataframe to join with summary stat
# table
df_temp = data.frame(matrix(nrow = nrow(dunn_res) * 2, ncol = 4))
colnames(df_temp) = c("species", "season", "seasonality_p", "significance")
for (i in 1:nrow(dunn_res)) {
    df_temp$species[i * 2 - 1] = dunn_res$species[i]
    df_temp$species[i * 2] = dunn_res$species[i]
    df_temp$season[i * 2 - 1] = "fall"
    df_temp$season[i * 2] = "winter"
    df_temp$seasonality_p[i * 2 - 1] = dunn_res$p.adj[i]
    df_temp$significance[i * 2 - 1] = dunn_res$p.adj.signif[i]
}

# compute summary stats for all species by season
voc_summary_stats = voc_long %>%
    dplyr::group_by(season, species) %>%
    dplyr::summarise(N = n(), mean = mean(conc), sd = sd(conc),
        min = min(conc), p25 = quantile(conc, probs = 0.25),
        p50 = quantile(conc, probs = 0.5), p75 = quantile(conc,
            probs = 0.75), max = max(conc)) %>%
    dplyr::left_join(., voc_BDL, by = c("species", "season")) %>%
    dplyr::left_join(., df_temp, by = c("species", "season")) %>%
    dplyr::arrange(species, season) %>%
    dplyr::select(species, season, N, `%_BDL`, min, max, mean,
        sd, p25, p50, p75, seasonality_p, significance)

# formatted table
table_out = voc_summary_stats %>%
    dplyr::mutate(dplyr::across(tidyselect::where(is.numeric),
        ~round(., digits = 4))) %>%
    dplyr::mutate_if(is.numeric, ~ifelse(. < 1e-04, "<0.0001",
        as.character(.))) %>%
    dplyr::mutate(`Mean (Std. Dev.)` = paste0(mean, " (", sd,
        ")"), `Min - Max` = paste0(min, " - ", max), `Q1/median/Q3` = paste0(p25,
        "/", p50, "/", p75)) %>%
    dplyr::rename(Species = species, Season = season, `% BDL` = `%_BDL`,
        `Seasonality P` = seasonality_p, Significance = significance) %>%
    dplyr::select(Species, Season, N, `% BDL`, `Min - Max`, `Mean (Std. Dev.)`,
        `Q1/median/Q3`, `Seasonality P`, Significance)

# optionally print these tables as xlsx - can edit/format
# manually
if (write_tables == T) {
    writexl::write_xlsx(x = voc_summary_stats, path = "preliminary/summary_stats_unformatted.xlsx")
    writexl::write_xlsx(x = table_out, path = "preliminary/summary_stats_formatted.xlsx")
}
rm(df_temp)
```

```
# Display Results as flextable
data_temp = table_out %>%
    tidyr::pivot_longer(cols = c("N", "% BDL", "Min - Max", "Mean (Std. Dev.)",
        "Q1/median/Q3", "Seasonality P", "Significance"), names_to = "stat",
        values_to = "value") %>%
    dplyr::mutate(stat = factor(stat, levels = c("N", "% BDL",
        "Min - Max", "Mean (Std. Dev.)", "Q1/median/Q3", "Seasonality P",
        "Significance")))

# stat_values
tab = flextable::tabulator(x = data_temp, rows = c("Species",
    "Season"), columns = c("stat"), stat_values = flextable::as_paragraph(flextable::as_chunk(value)))

ft = flextable::as_flextable(tab, separate_with = "Species")
print(ft)
```

```
## a flextable object.
## col_keys: `Species`, `Season`, `dummy1`, `N@stat_values`, `dummy2`, `% BDL@stat_values`, `dummy3`, `
## header has 1 row(s)
## body has 52 row(s)
## original dataset sample:
##      Species Season value@N value@% BDL value@Min - Max value@Mean (Std. Dev.)
## 1    Benzene   fall      39      7.6923       0.1 - 0.9        0.3667 (0.1603)
## 2    Benzene winter      36     <0.0001 0.2516 - 1.5834         0.7104 (0.285)
## 3 Chloroform   fall      39         100       0.1 - 0.1           0.1 (<0.0001)
## 4 Chloroform winter      36     97.2222     0.1 - 0.239        0.1039 (0.0232)
## 5     Cumene   fall      39         100       0.1 - 0.1           0.1 (<0.0001)
##    value@Q1/median/Q3 value@Seasonality P value@Significance stat_values@N
## 1     0.26/0.36/0.485            <0.0001              ****
## 2 0.542/0.6669/0.8158               <NA>              <NA>
## 3         0.1/0.1/0.1              0.298                ns
## 4         0.1/0.1/0.1               <NA>              <NA>
## 5         0.1/0.1/0.1               <NA>              <NA>
##   stat_values@% BDL stat_values@Min - Max stat_values@Mean (Std. Dev.)
## 1
## 2
## 3
## 4
## 5
##   stat_values@Q1/median/Q3 stat_values@Seasonality P stat_values@Significance
## 1
## 2
## 3
## 4
## 5
##   dummy1 N@stat_values dummy2 % BDL@stat_values dummy3 Min - Max@stat_values
## 1
## 2
## 3
## 4
## 5
##   dummy4 Mean (Std. Dev.)@stat_values dummy5 Q1/median/Q3@stat_values dummy6
## 1
## 2
## 3
## 4
## 5
##   Seasonality P@stat_values dummy7 Significance@stat_values
## 1
## 2
## 3
## 4
## 5
```