Pollutant Exposure Data

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(sp)
library(geosphere)
library(openxlsx)

pm_2016 <- read.csv("daily_88101_2016.csv")
year <- 2016

county_list <- read.xlsx("CountyList.xlsx")
combined_county_list <- paste(county_list[,1],county_list[,2])

pm_2016_clean <- pm_2016 %>% filter(Observation.Percent >= 50)

CBSA_list <- unique(pm_2016_clean$CBSA.Name)
CBSA_ind <- c(19,29,39,47,49,62,77,79,82,86,96,114,174,179,192,196,227,319,321,342)

pm_2016_clean <- mutate(pm_2016_clean, combined_county_state = paste(County.Name, State.Name))
```

```
# pm_2016_clean <- pm_2016_clean %>% filter((combined_county_state %in% combined_county_list) | CBSA.Na

pm_2016_clean <- pm_2016_clean %>% filter( CBSA.Name %in% CBSA_list[CBSA_ind])


unique(pm_2016_clean$combined_county_state) %in% combined_county_list
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [37]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [49]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [61]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [73]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [85]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
lon_list <- unique(pm_2016_clean$Longitude)
lat_list <- unique(pm_2016_clean$Latitude)
samdur_list <- unique(pm_2016_clean$Sample.Duration)

pm_2016.df <- data.frame()
for (lat_ind in 1:length(lat_list)){
  date_list <- seq.Date(as.Date(paste0(year,"-01-01")),as.Date(paste0(year,"-12-31")),"day")
  working_pm_2016 <- filter(pm_2016_clean,Latitude == lat_list[lat_ind])

  method1 <- filter(working_pm_2016,Sample.Duration == samdur_list[1])
  method1 <- select(method1,Date.Local, Arithmetic.Mean)
  method1 <- as.data.frame(method1) %>%
    mutate(Date = as.Date(Date.Local)) %>%
    complete(Date = date_list)

  method2 <- filter(working_pm_2016,Sample.Duration == samdur_list[2])
  method2 <- select(method2,Date.Local, Arithmetic.Mean)
  method2 <- as.data.frame(method2) %>%
    mutate(Date = as.Date(Date.Local)) %>%
    complete(Date = date_list)

  method3 <- filter(working_pm_2016,Sample.Duration == samdur_list[3])
  method3 <- select(method3,Date.Local, Arithmetic.Mean)
  method3 <- as.data.frame(method3) %>%
    mutate(Date = as.Date(Date.Local)) %>%
    complete(Date = date_list)


  missing_inds <- which(is.na(method3$Arithmetic.Mean))
  method3$Arithmetic.Mean[missing_inds] <- method2$Arithmetic.Mean[missing_inds]

  missing_inds <- which(is.na(method3$Arithmetic.Mean))
  method3$Arithmetic.Mean[missing_inds] <- method1$Arithmetic.Mean[missing_inds]


  to_add.df <- data.frame(Latitude = lat_list[lat_ind],
                          Longitude = lon_list[lat_ind],
                          Date = method3$Date,
                          armean = method3$Arithmetic.Mean)

  pm_2016.df <- rbind(pm_2016.df,to_add.df)
}
```

```
lon_list <- unique(pm_2016.df$Longitude)
lat_list <- unique(pm_2016.df$Latitude)


la_lon <- -118.2437
la_lat <- 34.0522

la_lon <- -118.239390
la_lat <- 33.950949

det_lon <- -112.0740
det_lat <- 33.4484

det_lon <- -83.0458
det_lat <- 42.3314



distHaversine(c(la_lon,la_lat),c(det_lon,det_lat), r=6378137) * 0.001
```

```
## [1] 3193.763
```

```
GetMonitorPM <- function(lon_list,lat_list,lon,lat, return_mult, radius){
  combined_lonlat <- cbind(lon_list,lat_list)
  dist <- apply(combined_lonlat,1,distHaversine,p1 = c(lon,lat)) * .001
  if (min(dist) > radius){
    print("Error: No measurements within radius")
    return(NA)
  }
  if (return_mult){
    within_rad <- which(dist<radius)
    return(cbind(lon_list[within_rad],lat_list[within_rad]))
  } else {
   return(cbind(lon_list[which.min(dist)],lat_list[which.min(dist)]))
  }
}

GetMonitorPM(lon_list,lat_list,la_lon,la_lat,F,8)
```

```
##          [,1]     [,2]
## [1,] -118.205 33.90139
```

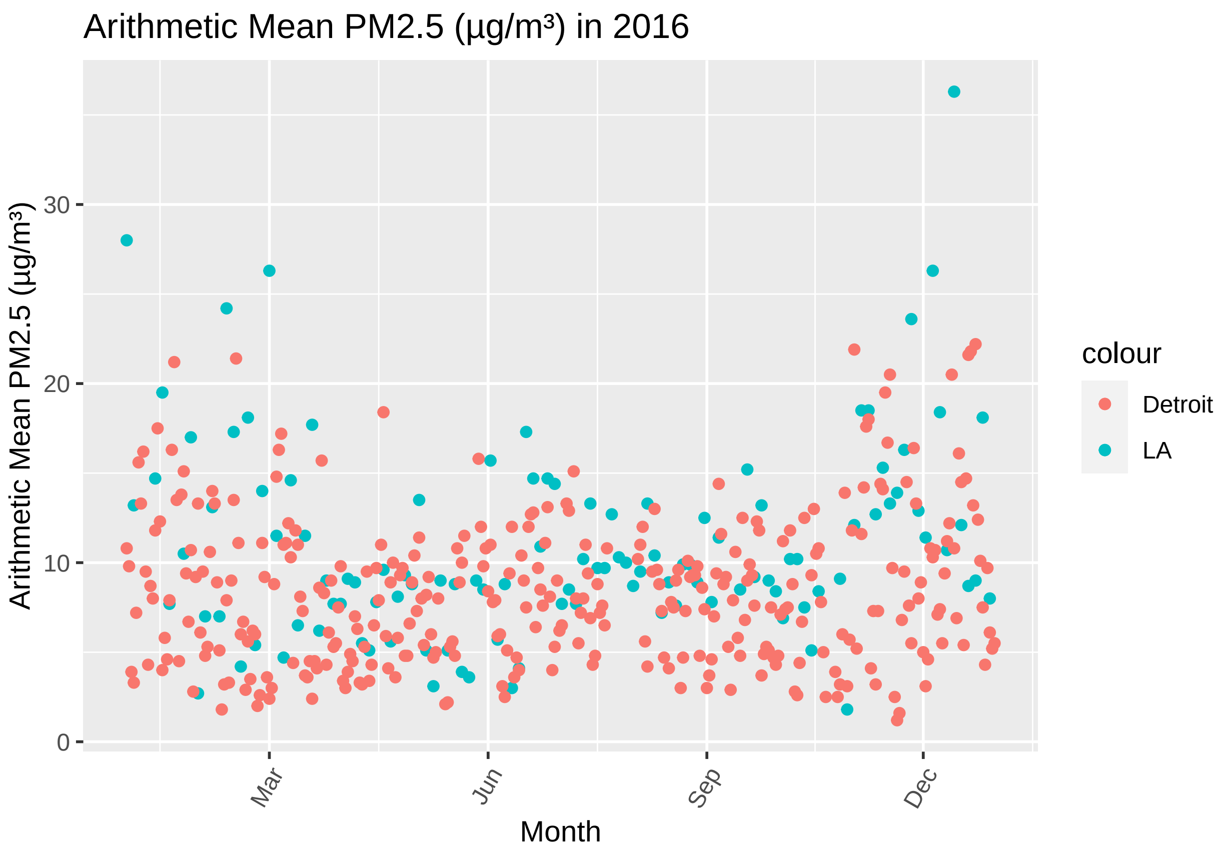
```
which.min(abs(lon_list-la_lon) + (abs(lat_list-la_lat)))
```

```
## [1] 25
```

```r
GetAP <- function(pm_data, lon_list, lat_list, lon, lat, return_mult, radius){

  lonlat <- GetMonitorPM(lon_list,lat_list, lon,lat,return_mult,radius)

  ap_df <- data.frame()

  for (i in 1:dim(lonlat)[1]){
    lon_adj <- lonlat[i,1]
    lat_adj <- lonlat[i,2]

    dist <- distHaversine(c(lon,lat),c(lon_adj,lat_adj), r=6378137) * 0.001
    ap_series <- filter(pm_data, Latitude == lat_adj & Longitude == lon_adj)

    to_add <- select(ap_series,armean, Date)
    to_add$AirMonDist <- dist

    ap_df <- rbind(ap_df, to_add)
  }

  return(ap_df)
}


la_ap <- GetAP(pm_2016.df, lon_list, lat_list, la_lon, la_lat,F,10)
det_ap <- GetAP(pm_2016.df, lon_list, lat_list, det_lon, det_lat,F,10)

ggplot() +
  geom_point(data=la_ap, aes(x=as.Date(Date), y=armean, color="LA ")) +
  geom_point(data=det_ap, aes(x=as.Date(Date), y=armean, color="Detroit")) +
  scale_x_date(date_breaks = "3 month", date_labels =  "%b") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  labs(x = "Month", y = "Arithmetic Mean PM2.5 (µg/m³)", title = "Arithmetic Mean PM2.5 (µg/m³) in 2016
```

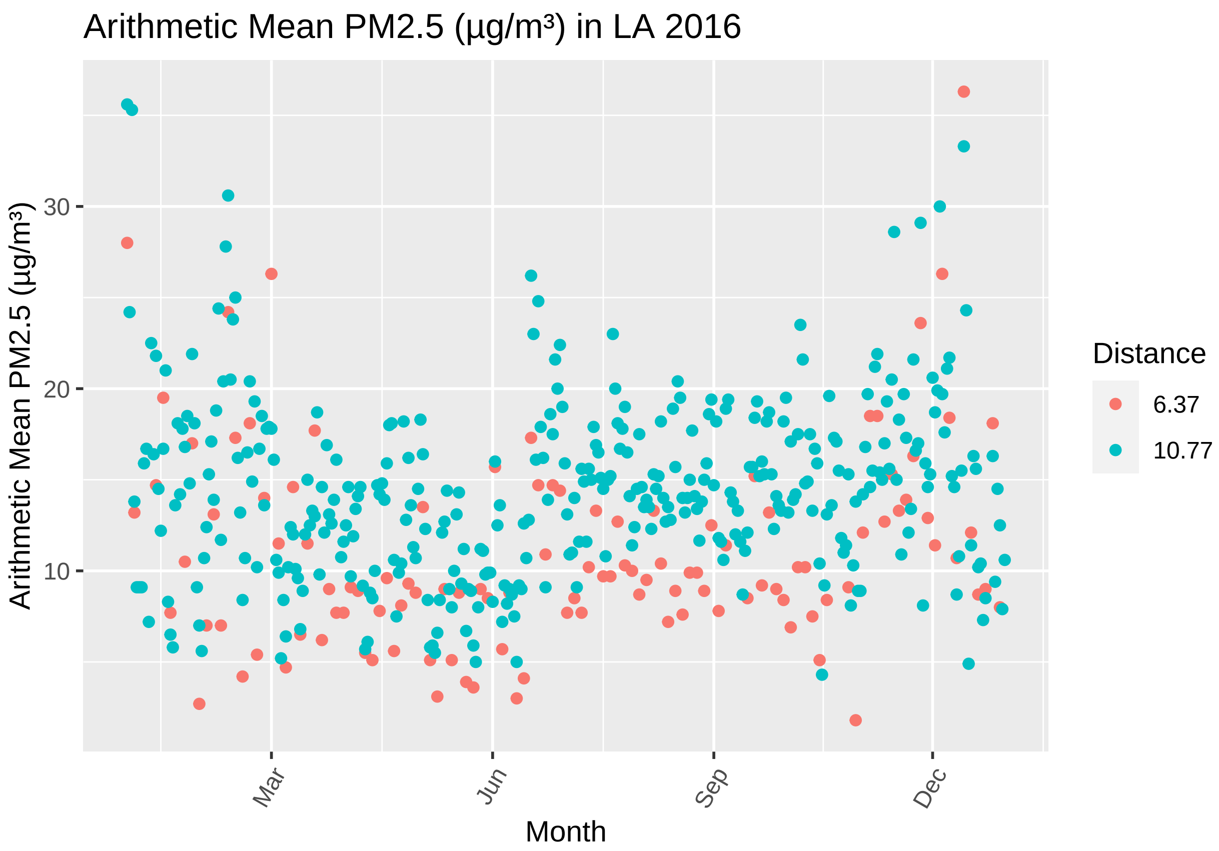
```
## Warning: Removed 251 rows containing missing values (geom_point).
```

```
## Warning: Removed 29 rows containing missing values (geom_point).
```

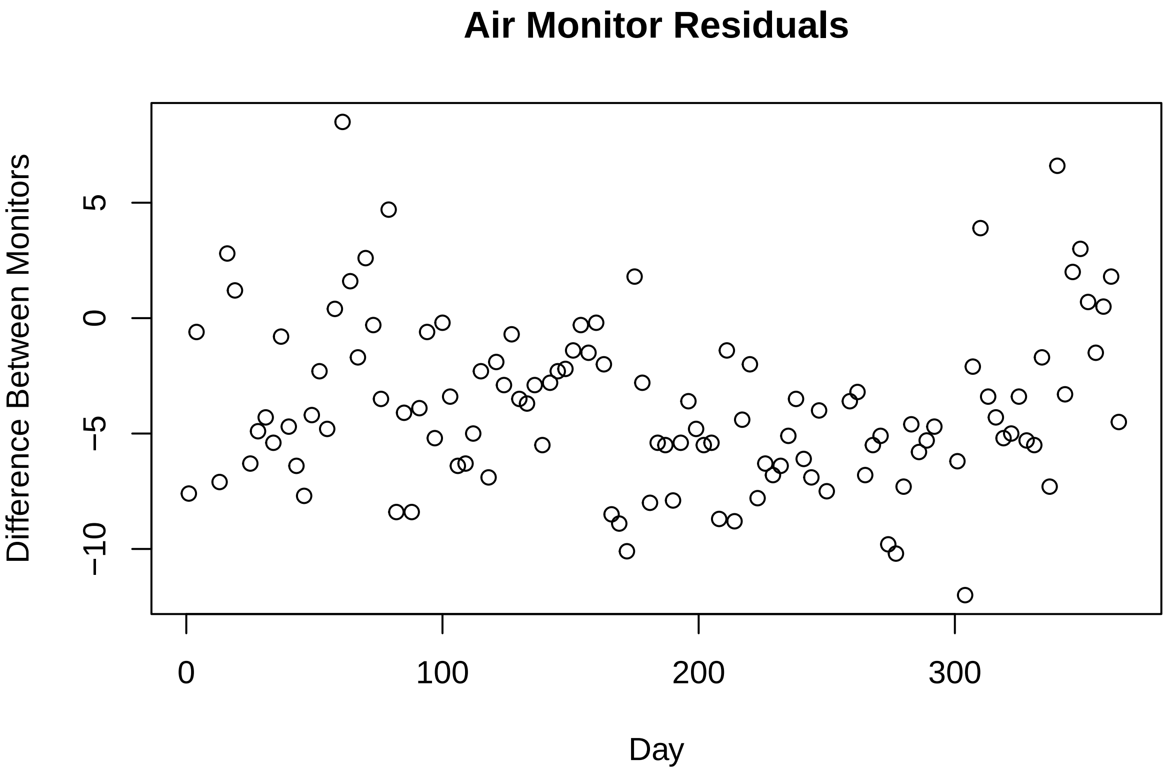


```
MultiAM.df <- GetAP(pm_2016.df, lon_list, lat_list, la_lon, la_lat,T,11)

ggplot() +
  geom_point(data=MultiAM.df, aes(x=as.Date(Date), y=armean, color=factor(round(AirMonDist,2)))) +
  scale_x_date(date_breaks = "3 month", date_labels =  "%b") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  labs(x = "Month", y = "Arithmetic Mean PM2.5 (µg/m³)", title = "Arithmetic Mean PM2.5 (µg/m³) in LA 2
```

```
## Warning: Removed 254 rows containing missing values (geom_point).
```



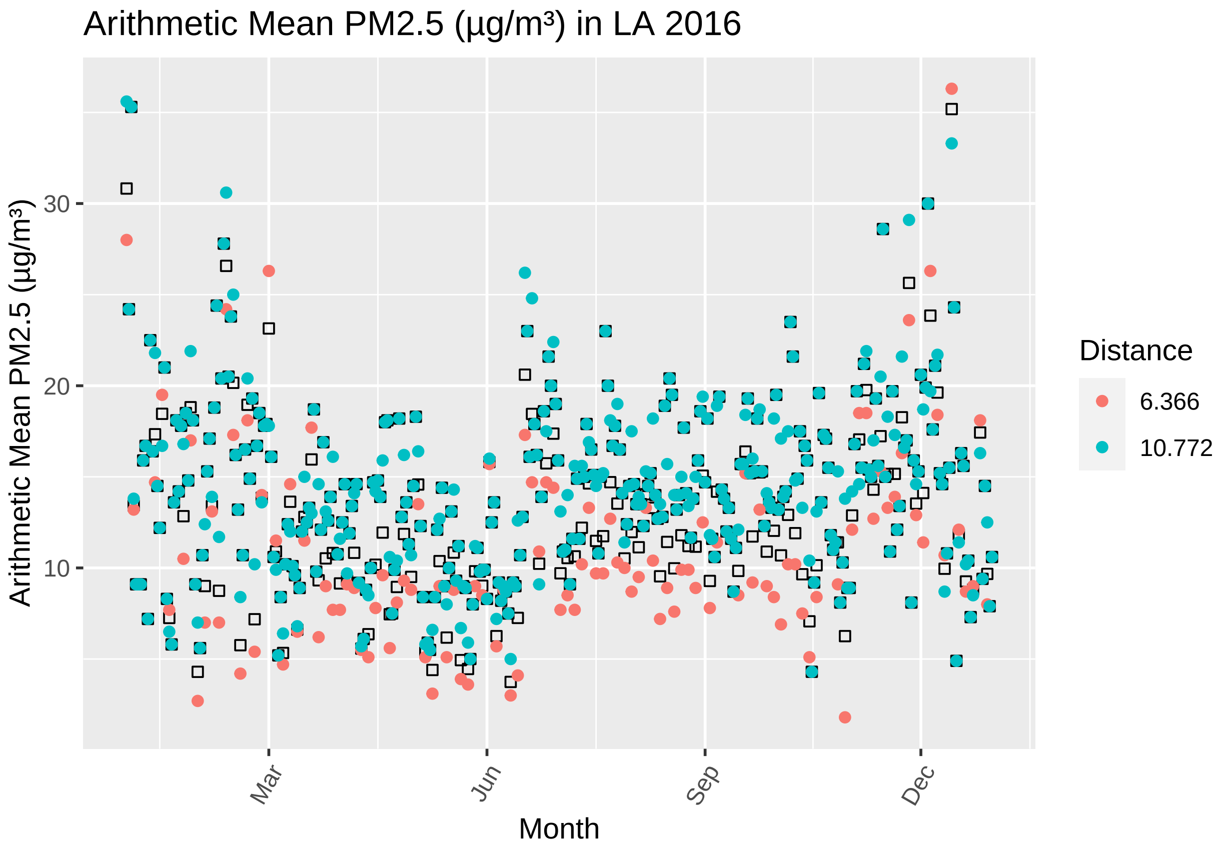
```
plot(filter(MultiAM.df,AirMonDist < 7)$armean - filter(MultiAM.df,AirMonDist > 7)$armean,
     xlab = "Day",
     ylab = "Difference Between Monitors",
     main = "Air Monitor Residuals")
```



```
InverseWeightedAvg <- function(MultiAM.df){
  spread_ap <- spread(MultiAM.df,AirMonDist,armean)
  dist_vec <- as.double(colnames(spread_ap)[2:dim(spread_ap)[2]])
  inv_weighted_vec <- numeric(dim(spread_ap)[1])
  for(i in 1:dim(spread_ap)[1]){

    armeans <- spread_ap[i,2:dim(spread_ap)[2]]

    if (sum(is.na(armeans)) == 3){
      inv_weighted_vec[i] <- NA
    } else {
      total_dist <- as.integer(!is.na(armeans)) %*% (1/dist_vec)
      armeans * (1/dist_vec)
      inv_weighted_armean <- sum((armeans / dist_vec)/total_dist, na.rm = T)

      inv_weighted_vec[i] <- inv_weighted_armean
    }
  }
  inv_weighted.df <- mutate(select(spread_ap,Date),armean = inv_weighted_vec)

  missing_days <- which(inv_weighted.df$armean == 0)
  inv_weighted.df$armean[missing_days] <- NA

  return(inv_weighted.df)
}
```

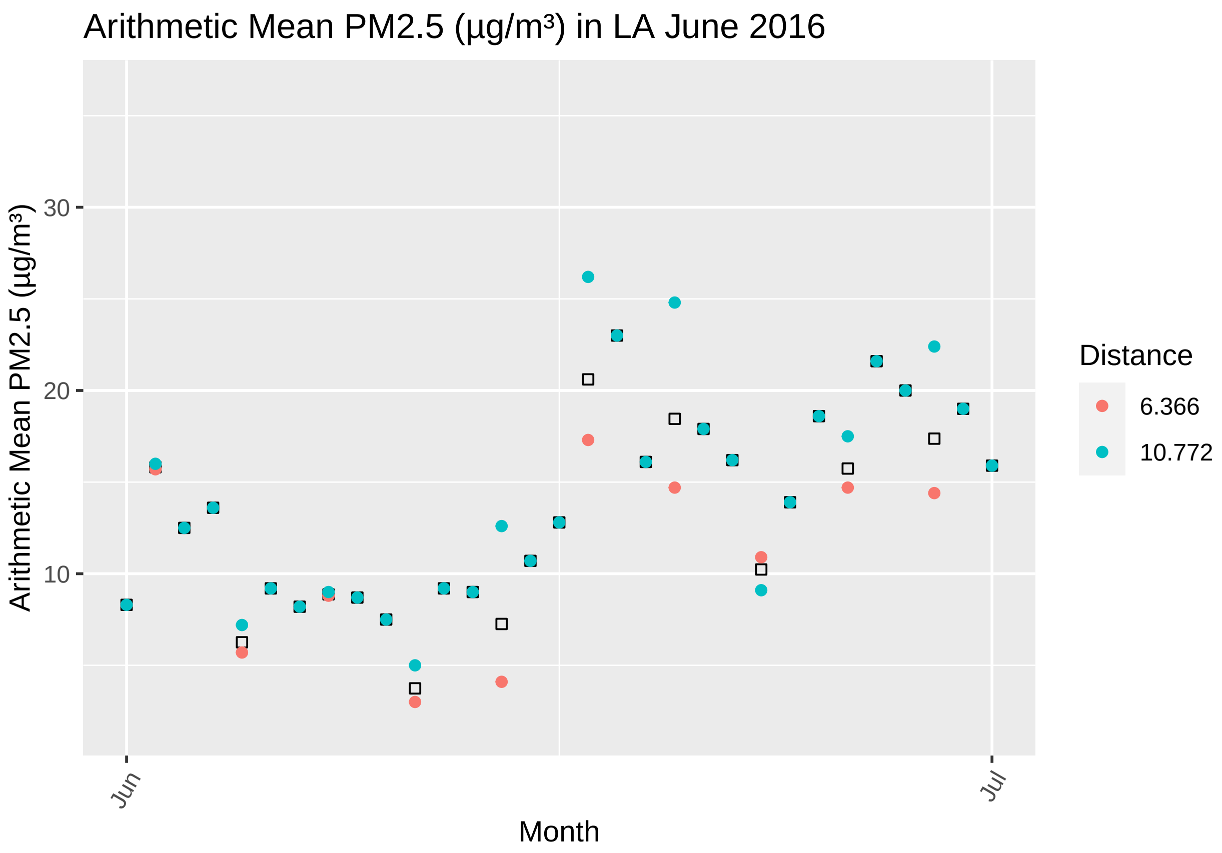
```r
inv_weighted.df <- InverseWeightedAvg(MultiAM.df)

ggplot() +
  geom_point(data=inv_weighted.df, aes(x=as.Date(Date), y=armean), shape = 0) +
  geom_point(data=MultiAM.df, aes(x=as.Date(Date), y=armean, color=factor(round(AirMonDist,3)))) +
  scale_x_date(date_breaks = "3 month", date_labels =  "%b") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  labs(x = "Month", y = "Arithmetic Mean PM2.5 (µg/m³)", title = "Arithmetic Mean PM2.5 (µg/m³) in LA 2
```

```
## Warning: Removed 3 rows containing missing values (geom_point).

## Warning: Removed 254 rows containing missing values (geom_point).
```



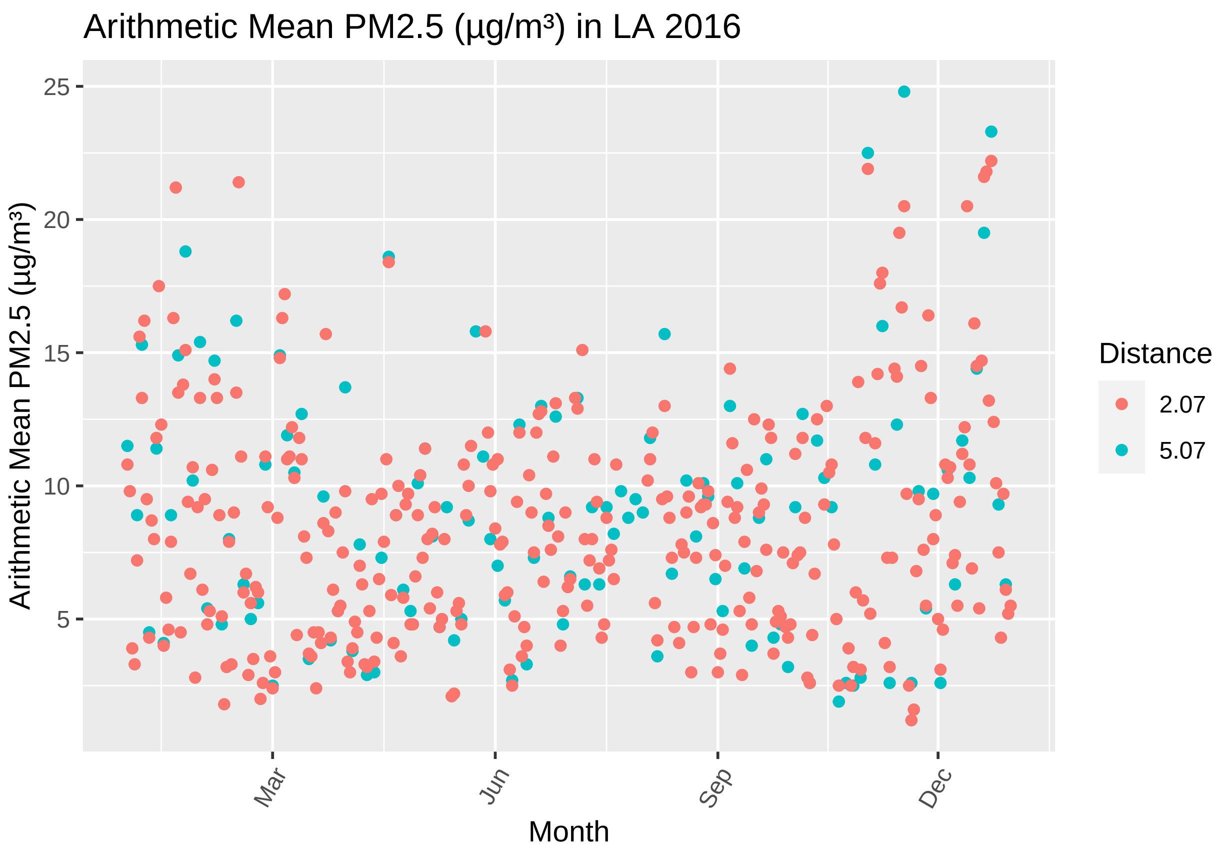
```r
ggplot() +
  geom_point(data=inv_weighted.df, aes(x=as.Date(Date), y=armean), shape = 0) +
  geom_point(data=MultiAM.df, aes(x=as.Date(Date), y=armean, color=factor(round(AirMonDist,3)))) +
  scale_x_date(date_breaks = "1 month", date_labels =  "%b",
              limits = as.Date(c('2016-06-01','2016-7-01'))) +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  labs(x = "Month", y = "Arithmetic Mean PM2.5 (µg/m³)", title = "Arithmetic Mean PM2.5 (µg/m³) in LA J
```

```
## Warning: Removed 335 rows containing missing values (geom_point).

## Warning: Removed 691 rows containing missing values (geom_point).
```

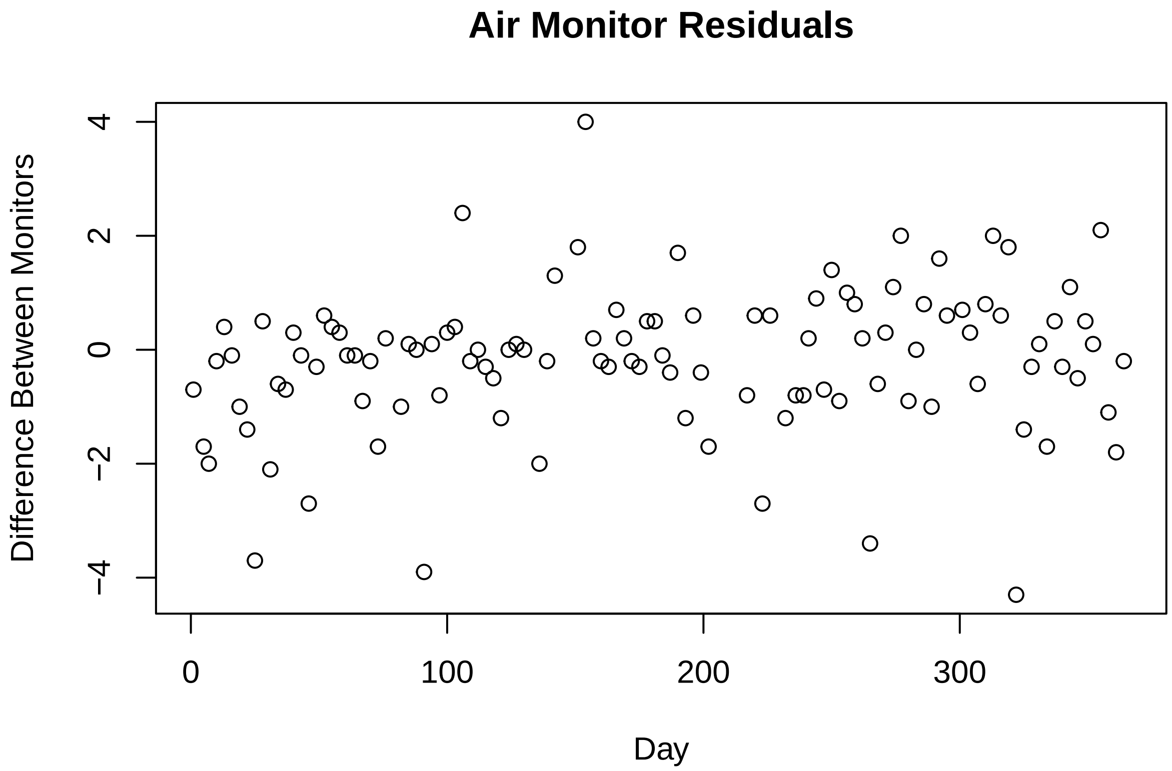


```
MultiAM.df <- GetAP(pm_2016.df, lon_list, lat_list, det_lon, det_lat,T,5.5)

ggplot() +
  geom_point(data=MultiAM.df, aes(x=as.Date(Date), y=armean, color=factor(round(AirMonDist,2)))) +
  scale_x_date(date_breaks = "3 month", date_labels =  "%b") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  labs(x = "Month", y = "Arithmetic Mean PM2.5 (µg/m³)", title = "Arithmetic Mean PM2.5 (µg/m³) in LA 2
```

```
## Warning: Removed 275 rows containing missing values (geom_point).
```



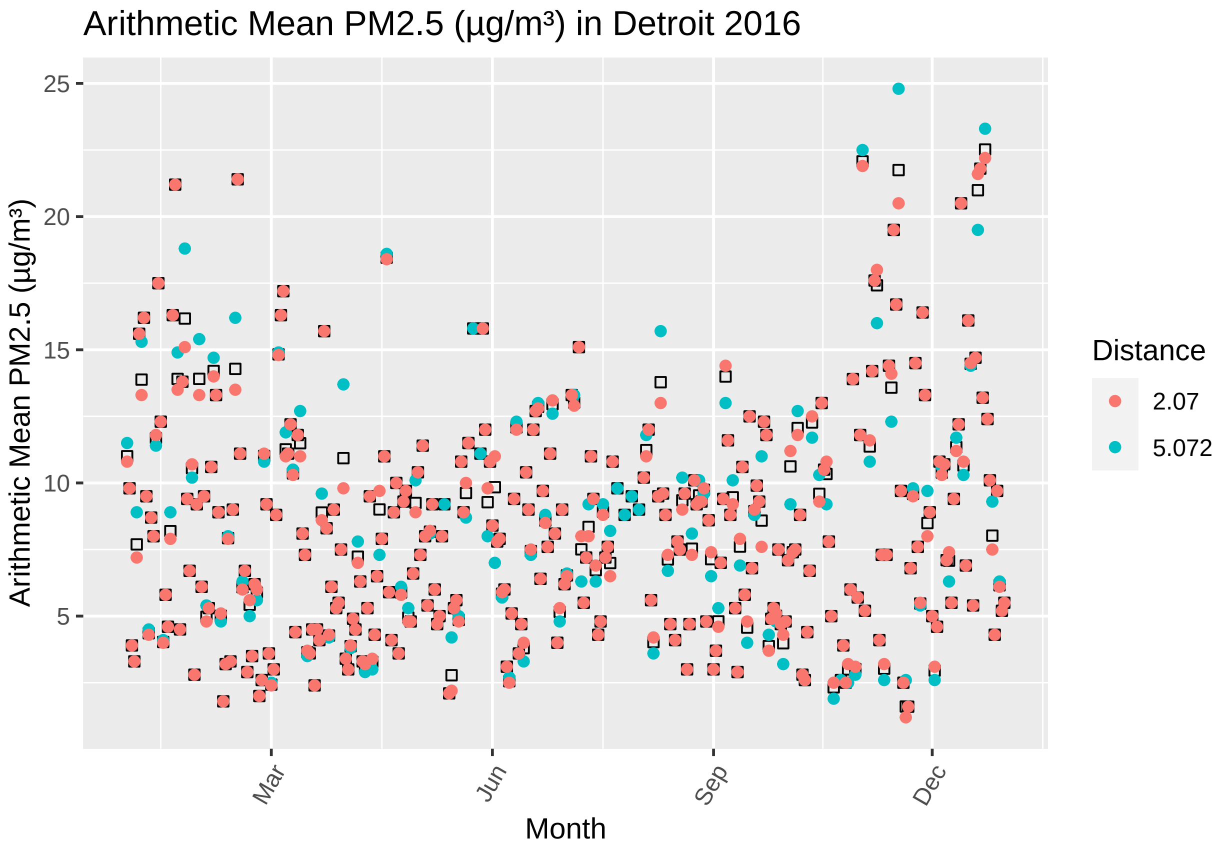
```
plot(filter(MultiAM.df,AirMonDist < 5)$armean - filter(MultiAM.df,AirMonDist > 5)$armean,
     xlab = "Day",
     ylab = "Difference Between Monitors",
     main = "Air Monitor Residuals")
```



```
InverseWeightedAvg <- function(MultiAM.df){
  spread_ap <- spread(MultiAM.df,AirMonDist,armean)
  dist_vec <- as.double(colnames(spread_ap)[2:dim(spread_ap)[2]])
  inv_weighted_vec <- numeric(dim(spread_ap)[1])
  for(i in 1:dim(spread_ap)[1]){

    armeans <- spread_ap[i,2:dim(spread_ap)[2]]

    if (sum(is.na(armeans)) == 3){
      inv_weighted_vec[i] <- NA
    } else {
      total_dist <- as.integer(!is.na(armeans)) %*% (1/dist_vec)
      armeans * (1/dist_vec)
      inv_weighted_armean <- sum((armeans / dist_vec)/total_dist, na.rm = T)

      inv_weighted_vec[i] <- inv_weighted_armean
    }
  }
  inv_weighted.df <- mutate(select(spread_ap,Date),armean = inv_weighted_vec)

  missing_days <- which(inv_weighted.df$armean == 0)
  inv_weighted.df$armean[missing_days] <- NA

  return(inv_weighted.df)
}
```

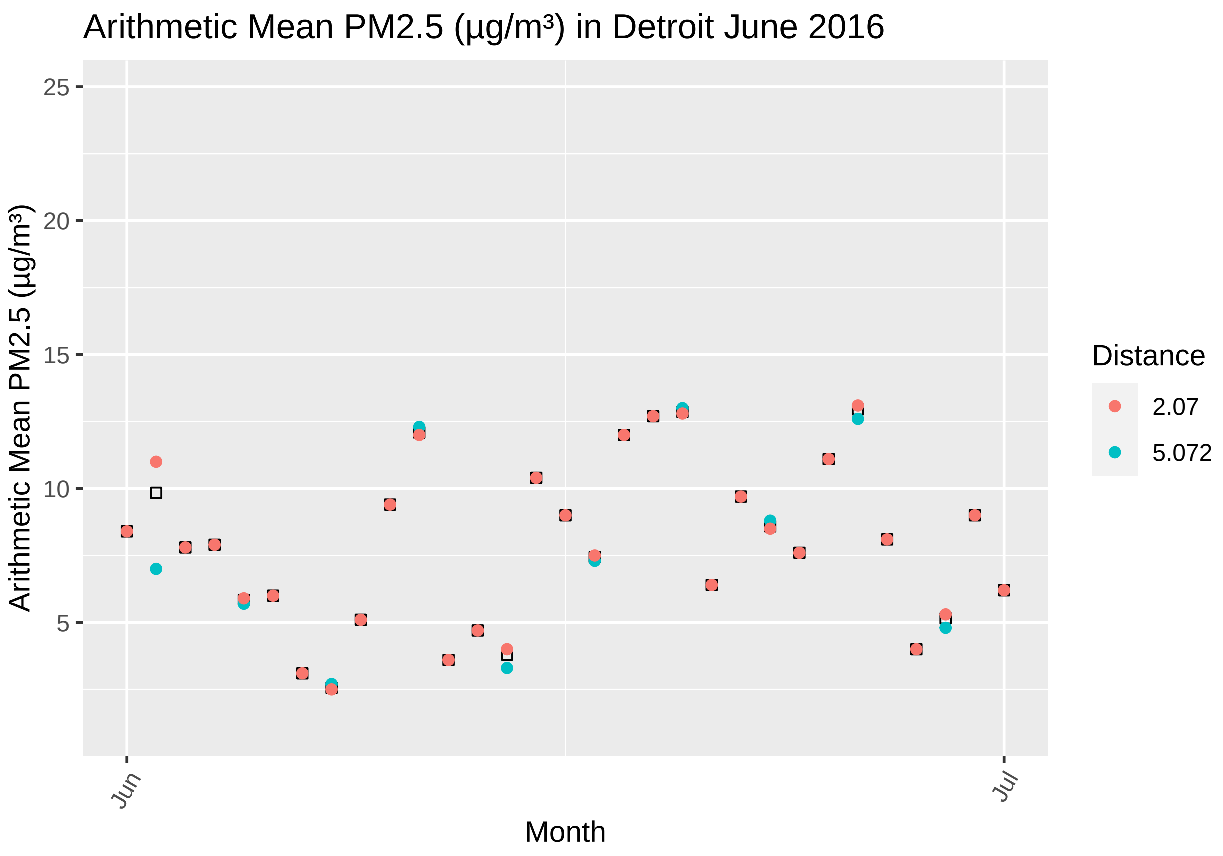
```
inv_weighted.df <- InverseWeightedAvg(MultiAM.df)

ggplot() +
  geom_point(data=inv_weighted.df, aes(x=as.Date(Date), y=armean), shape = 0) +
  geom_point(data=MultiAM.df, aes(x=as.Date(Date), y=armean, color=factor(round(AirMonDist,3)))) +
  scale_x_date(date_breaks = "3 month", date_labels =  "%b") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  labs(x = "Month", y = "Arithmetic Mean PM2.5 (µg/m³)", title = "Arithmetic Mean PM2.5 (µg/m³) in Detr
```

```
## Warning: Removed 21 rows containing missing values (geom_point).

## Warning: Removed 275 rows containing missing values (geom_point).
```



```
ggplot() +
  geom_point(data=inv_weighted.df, aes(x=as.Date(Date), y=armean), shape = 0) +
  geom_point(data=MultiAM.df, aes(x=as.Date(Date), y=armean, color=factor(round(AirMonDist,3)))) +
  scale_x_date(date_breaks = "1 month", date_labels =  "%b",
               limits = as.Date(c('2016-06-01','2016-7-01'))) +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  labs(x = "Month", y = "Arithmetic Mean PM2.5 (µg/m³)", title = "Arithmetic Mean PM2.5 (µg/m³) in Detr
```

```
## Warning: Removed 335 rows containing missing values (geom_point).

## Warning: Removed 691 rows containing missing values (geom_point).
```


Arithmetic Mean PM2.5 (µg/m³) in Detroit June 2016
Arithmetic Mean PM2.5 (µg/m³)
25
20
15
10
5
Jun
Jul
Month
Distance
2.07
5.072