R Markdown Code for “Take-Off Engine Particle Emission Indices for In-Service Aircraft at Los Angeles International Airport” Data Descriptor

Richard H. Moore, Michael A. Shook, Luke D. Ziemba, Joshua P. DiGangi, Edward L. Winstead, Bastian Rauch, Tina Jurkat, Kenneth L. Thornhill, Ewan C. Crosbie, Claire Robinson, Taylor J. Shingler, and Bruce E. Anderson. Correspondence to: richard.h.moore@nasa.gov

Revision:

R00 Final processing code

R01 Updated to reflect summary statistics for R01 test point data

Read in Data Record

```
library(readxl)
LAX_Data <- read_excel("D:/LAX-Ground-Summary_TP_20140518_R01_thru20140525.xlsx",
    sheet = "LAX_Data", na = "NA")
```

Functions to compute geometric mean and geometric standard deviation

```
# Compute geometric mean of a vector
geomean <- function(x, na.rm = FALSE, trim = 0, ...) {
  exp(mean(log(x, ...), na.rm = na.rm, trim = trim, ...))
}

# Compute geometric standard deviation of a vector
geosd <- function(x, na.rm = FALSE, ...) {
  exp(sd(log(x, ...), na.rm = na.rm, ...))
}

# Compute geometric mean for each column in a 2-D matrix
geocolmeans <- function(x, na.rm = FALSE, ...){
  exp(apply(log(x), 2, mean, na.rm = na.rm))
}

# Compute geometric standard deviation for each column in a 2-D matrix
geocolsd <- function(x, na.rm = FALSE, ...){
  exp(apply(log(x), 2, sd, na.rm = na.rm))
}
```

Code to compute geometric mean and standard deviations for different engine types.

Data are used to generate Table 4 in the data descriptor.

```
# Initialize variables
EngineModel <- c()
BC_mu <- c()
Num_Tot_mu <- c()
Num_NV_mu <- c()
CCN_mu <- c()
EEPSn_mu <- c()
EEPSv_mu <- c()

BC_sd <- c()
Num_Tot_sd <- c()
Num_NV_sd <- c()
CCN_sd <- c()
EEPSn_sd <- c()
EEPSv_sd <- c()

BC_N <- c()
Num_Tot_N <- c()
```

```
Num_NV_N <- c()
CCN_N <- c()
EEPSn_N <- c()
EEPSv_N <- c()



MM <- LAX_Data$Engine_Master_Model # Assign shorthand variable



# Loop over all engine master models and compute geometric means and standard deviations
for (StrV in c("CFM56-3B",
               "CFM56-3C",
               "CFM56-5A",
               "CFM56-5B",
               "CFM56-5C",
               "CFM56-7B",
               "CF34-3",
               "CF34-8",
               "CF6-80C2",
               "GE90-94B",
               "GE90-115B",
               "GEnx-2B67",
               "GP7270",
               "JT8D",
               "PT6A",
               "PW118",
               "PW150A",
               "PW2000",
               "PW4000",
               "AE3007",
               "RB211",
               "Trent 556",
               "Trent 772",
               "Trent 892",
               "Trent 970",
               "V2522",
               "V2524",
               "V2527",
               "V2533")){

                        EngineModel <- c(EngineModel, StrV)
                        BC_mu <- c(BC_mu, geomean(LAX_Data$EI_BC[MM==StrV], na.rm=TRUE))
                        BC_sd <- c(BC_sd, geosd(LAX_Data$EI_BC[MM==StrV], na.rm=TRUE))
                        BC_N <- c(BC_N, sum(LAX_Data$EI_BC[MM==StrV]>-1, na.rm=TRUE))

                        Num_Tot_mu <- c(Num_Tot_mu, geomean(LAX_Data$EI_Number_Total[MM==StrV], na.rm=T
RUE))
                        Num_Tot_sd <- c(Num_Tot_sd, geosd(LAX_Data$EI_Number_Total[MM==StrV], na.rm=TRU
E))
                        Num_Tot_N <- c(Num_Tot_N, sum(LAX_Data$EI_Number_Total[MM==StrV]>-1, na.rm=TRUE
))

                        Num_NV_mu <- c(Num_NV_mu, geomean(LAX_Data$EI_Number_Nonvol[MM==StrV], na.rm=TR
UE))
                        Num_NV_sd <- c(Num_NV_sd, geosd(LAX_Data$EI_Number_Nonvol[MM==StrV], na.rm=TRUE
))
                        Num_NV_N <- c(Num_NV_N, sum(LAX_Data$EI_Number_Nonvol[MM==StrV]>-1, na.rm=TRUE)
)

                        CCN_mu <- c(CCN_mu, geomean(LAX_Data$EI_CCN[MM==StrV], na.rm=TRUE))
                        CCN_sd <- c(CCN_sd, geosd(LAX_Data$EI_CCN[MM==StrV], na.rm=TRUE))
                        CCN_N <- c(CCN_N, sum(LAX_Data$EI_CCN[MM==StrV]>-1, na.rm=TRUE))

                        EEPSn_mu <- c(EEPSn_mu, geomean(LAX_Data$EI_Number_EEPS[MM==StrV], na.rm=TRUE))
                        EEPSn_sd <- c(EEPSn_sd, geosd(LAX_Data$EI_Number_EEPS[MM==StrV], na.rm=TRUE))
                        EEPSn_N <- c(EEPSn_N, sum(LAX_Data$EI_Number_EEPS[MM==StrV]>-1, na.rm=TRUE))

                        EEPSv_mu <- c(EEPSv_mu, geomean(LAX_Data$EI_Volume_EEPS[MM==StrV], na.rm=TRUE))
                        EEPSv_sd <- c(EEPSv_sd, geosd(LAX_Data$EI_Volume_EEPS[MM==StrV], na.rm=TRUE))
                        EEPSv_N <- c(EEPSv_N, sum(LAX_Data$EI_Volume_EEPS[MM==StrV]>-1, na.rm=TRUE))
```

```
                        }

# Assemble output data into a data frame
Table1 <- data.frame(EngineModel, BC_mu, BC_sd, Num_Tot_mu, Num_Tot_sd, Num_NV_mu, Num_NV_sd)
Table2 <- data.frame(EngineModel, CCN_mu, CCN_sd, EEPSn_mu, EEPSn_sd, EEPSv_mu, EEPSv_sd)
Table3 <- data.frame(EngineModel, BC_N, Num_Tot_N, Num_NV_N, CCN_N, EEPSn_N, EEPSv_N)

Table1[,-1] <-format(Table1[,-1], digits=4) # Format the precision of the data frame numbers
Table2[,-1] <-format(Table2[,-1], digits=4) # Format the precision of the data frame numbers
```

Summary Statistics (Table 1 of 3)

```
print(Table1)
```

```
##    EngineModel  BC_mu BC_sd Num_Tot_mu Num_Tot_sd Num_NV_mu Num_NV_sd
## 1     CFM56-3B 563.96 1.410  3.557e+16      1.411 2.543e+15     2.345
## 2     CFM56-3C 792.14    NA  1.647e+16         NA 2.307e+15        NA
## 3     CFM56-5A 419.17 1.373  4.618e+16      1.248 1.848e+15     2.496
## 4     CFM56-5B 276.12 1.512  5.089e+16      1.387 1.783e+15     2.200
## 5     CFM56-5C 415.59 1.075  5.057e+16      1.024 1.107e+15     1.554
## 6     CFM56-7B 348.07 1.553  5.217e+16      1.312 2.211e+15     2.334
## 7       CF34-3 487.76 1.588  3.618e+16      1.501 4.989e+15     1.339
## 8       CF34-8 435.25 1.486  2.975e+16      1.744 2.668e+15     1.788
## 9     CF6-80C2 189.93 1.485  3.829e+16      1.445 1.062e+15     2.536
## 10    GE90-94B  92.83 1.116  3.952e+16      1.160 5.116e+14     1.132
## 11   GE90-115B 175.24 1.298  3.392e+16      1.198 7.952e+14     1.960
## 12   GEnx-2B67  46.06 1.475  8.727e+16      1.414 1.948e+15     4.070
## 13      GP7270 129.74 1.327  2.855e+16      1.245 1.891e+15     1.178
## 14        JT8D 940.79 1.064  1.727e+16      2.883 2.832e+15     2.526
## 15        PT6A 440.91 2.048  3.769e+16      1.841 6.504e+15     1.793
## 16       PW118 649.34 1.616  5.798e+16      1.329 2.975e+15     1.693
## 17      PW150A 252.18 1.505  5.869e+16      1.255 4.487e+15     3.563
## 18      PW2000 816.96    NA  2.459e+16         NA 2.554e+15        NA
## 19      PW4000 439.41 1.653  8.799e+15      1.055 1.704e+15     1.223
## 20      AE3007 591.66    NA  2.607e+16      3.113 1.447e+15     1.024
## 21       RB211 196.29 1.246  4.115e+16      1.388 1.066e+15     2.079
## 22   Trent 556 319.49 1.487  2.831e+16      1.278 1.061e+15     3.993
## 23   Trent 772 177.42 1.011  2.487e+16      1.322 1.058e+15     1.371
## 24   Trent 892 303.56    NA  3.040e+16         NA 8.087e+14        NA
## 25   Trent 970 355.63 1.231  5.131e+16      1.169 8.439e+14     1.601
## 26       V2522 583.92    NA  2.576e+16         NA 2.861e+15        NA
## 27       V2524 785.64 1.207  1.137e+16      1.152 3.422e+15     2.096
## 28       V2527 413.87 1.331  1.934e+16      1.746 2.363e+15     1.733
## 29       V2533 343.41 1.292  2.091e+16      1.981 1.899e+15     3.147
```

Summary Statistics (Table 2 of 3)

```
print(Table2)
```

```
##    EngineModel    CCN_mu CCN_sd  EEPSn_mu EEPSn_sd EEPSv_mu EEPSv_sd
## 1     CFM56-3B 7.279e+14  1.385 2.705e+16    1.527    406.4    1.247
## 2     CFM56-3C 6.849e+14     NA 1.252e+16       NA    699.7       NA
## 3     CFM56-5A 5.407e+14  1.443 3.538e+16    1.262    326.9    1.337
## 4     CFM56-5B 4.507e+14  1.525 3.794e+16    1.375    274.1    1.303
## 5     CFM56-5C 4.535e+14  1.068 3.784e+16    1.184    316.7    1.176
## 6     CFM56-7B 4.554e+14  1.908 3.853e+16    1.290    305.5    1.378
## 7       CF34-3 9.129e+14  1.233 2.608e+16    1.423    361.1    1.394
## 8       CF34-8 6.974e+14  1.895 2.591e+16    1.701    295.1    1.496
## 9     CF6-80C2 2.860e+14  1.658 3.509e+16    1.460    203.2    1.571
## 10    GE90-94B 1.802e+14  1.273 2.945e+16    1.310    123.8    1.397
## 11   GE90-115B 3.422e+14  1.582 2.875e+16    1.256    191.0    1.334
## 12   GEnx-2B67 3.772e+14  1.497 4.824e+16    1.049    557.1    1.339
## 13      GP7270 3.228e+14  1.164 3.490e+16    1.419    171.4    1.041
## 14        JT8D 1.266e+15  1.552 1.233e+16    2.890    616.5    1.228
## 15        PT6A 6.454e+14  2.073 2.799e+16    1.564    457.8    2.016
## 16       PW118 8.795e+14  1.301 4.736e+16    1.307    478.4    1.519
## 17      PW150A 5.933e+14  1.020 4.266e+16    1.292    355.9    1.324
## 18      PW2000 1.160e+15     NA 2.730e+16       NA    650.0       NA
## 19      PW4000 5.110e+14  1.478 7.946e+15    1.946    340.8    1.548
## 20      AE3007 1.895e+14  3.270 3.532e+16    1.034    138.0    1.321
## 21       RB211 2.246e+14  1.012 3.029e+16    1.392    241.7    1.697
## 22   Trent 556 3.181e+14  1.313 1.986e+16    1.042    211.8    1.284
## 23   Trent 772 3.092e+14  1.206 2.528e+16    1.373    170.6    1.385
## 24   Trent 892 4.364e+14     NA 3.755e+16       NA    248.6       NA
## 25   Trent 970 3.119e+14  1.232 3.813e+16    1.127    340.0    1.097
## 26       V2522 1.019e+15     NA 2.073e+16       NA    392.3       NA
## 27       V2524 1.435e+15  1.429 1.038e+16    1.399    501.2    1.163
## 28       V2527 8.288e+14  1.506 2.215e+16    1.593    375.7    1.360
## 29       V2533 5.681e+14  1.508 2.565e+16    1.772    288.7    1.495
```

Summary Statistics (Table 3 of 3)

```
print(Table3)
```

```
##    EngineModel BC_N Num_Tot_N Num_NV_N CCN_N EEPSn_N EEPSv_N
## 1     CFM56-3B   20        20       20    18      17      17
## 2     CFM56-3C    1         1        1     1       1       1
## 3     CFM56-5A   11        11       11    10      10      10
## 4     CFM56-5B   40        40       40    36      37      37
## 5     CFM56-5C    2         2        2     2       2       2
## 6     CFM56-7B   83        86       86    76      80      80
## 7       CF34-3    6         6        6     6       6       6
## 8       CF34-8   21        23       23    21      21      21
## 9     CF6-80C2    6         6        6     6       5       5
## 10    GE90-94B    2         2        2     2       2       2
## 11   GE90-115B    6         6        6     6       5       5
## 12   GEnx-2B67    2         2        2     2       2       2
## 13      GP7270    3         3        3     3       3       3
## 14        JT8D    2         2        2     2       2       2
## 15        PT6A    7         7        7     5       5       5
## 16       PW118   10        10       10    10      10      10
## 17      PW150A    3         3        3     2       2       2
## 18      PW2000    1         1        1     1       1       1
## 19      PW4000    2         2        2     2       2       2
## 20      AE3007    1         2        2     2       2       2
## 21       RB211    2         2        2     2       2       2
## 22   Trent 556    2         2        2     2       2       2
## 23   Trent 772    2         2        2     2       2       2
## 24   Trent 892    1         1        1     1       1       1
## 25   Trent 970    3         3        3     3       3       3
## 26       V2522    1         1        1     1       1       1
## 27       V2524    2         2        2     2       2       2
## 28       V2527   17        17       17    17      16      16
## 29       V2533   10        10       10     9       9       9
```

Code to compute geometric mean and standard deviation size distributions for different engine types.

Data are fit to log-normal functions to generate Table 5 in the data descriptor.

```
SizeDistrData <- as.matrix(LAX_Data[,38:69])


# Initialize Variables
EngineModel <- c()
SizeDistr_mu <- data.frame()
SizeDistr_sd <- data.frame()
N <- data.frame()

# Loop over all engine master models and compute column geometric means and standard deviations
for (StrV in c("CFM56-3B",
               "CFM56-3C",
               "CFM56-5A",
               "CFM56-5B",
               "CFM56-5C",
               "CFM56-7B",
               "CF34-3",
               "CF34-8",
               "CF6-80C2",
               "GE90-94B",
               "GE90-115B",
               "GEnx-2B67",
               "GP7270",
               "JT8D",
               "PT6A",
               "PW118",
               "PW150A",
               "PW2000",
               "PW4000",
               "AE3007",
               "RB211",
               "Trent 556",
               "Trent 772",
               "Trent 892",
               "Trent 970",
               "V2522",
               "V2524",
               "V2527",
               "V2533")){

                        EngineModel <- c(EngineModel, StrV)

                        SubsetData <- subset(SizeDistrData, LAX_Data$Engine_Master_Model==StrV)

                        SizeDistr_mu <- rbind(SizeDistr_mu, t(geocolmeans(SubsetData, na.rm=TRUE)))
                        SizeDistr_sd <- rbind(SizeDistr_sd, t(geocolsd(SubsetData, na.rm=TRUE)))
                        N <- rbind(N, (NROW(SubsetData)))
                      }


row.names(SizeDistr_mu) <- EngineModel
row.names(SizeDistr_sd) <- EngineModel
row.names(N) <- EngineModel
```

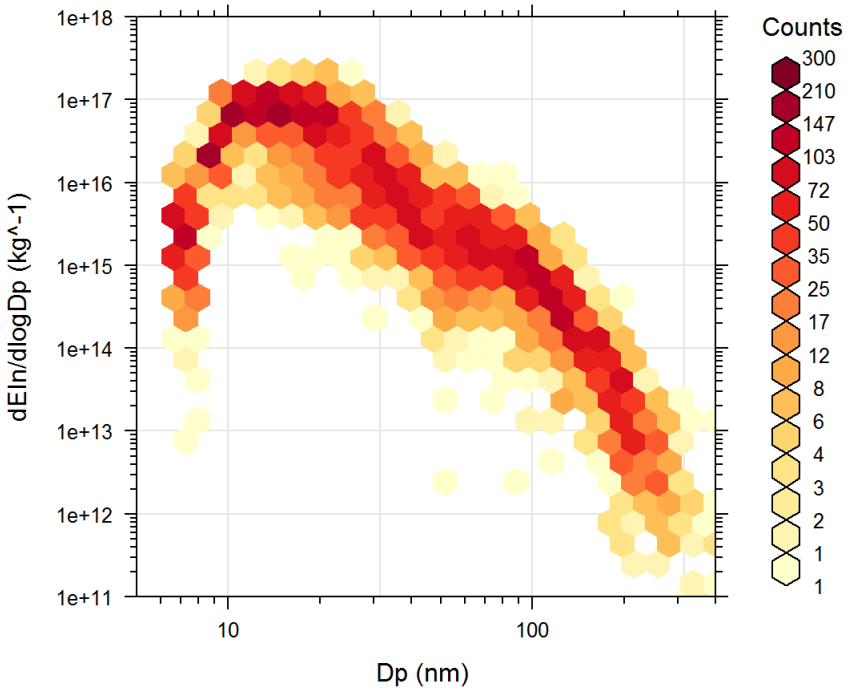
Code to generate size distribution density plots.

Figure 5 in the data descriptor.

```
# Load external packages
library(openair)
library(hexbin)

# Input vector containing EEPS bin midpint diameters (as corrected, see data descriptor)
Dp = c(6.64, 7.68, 8.87, 10.2, 11.9, 13.6, 15.7, 18.2, 21.0, 24.3, 28.1,
       32.3, 37.4, 43.1, 49.8, 57.5, 66.4, 76.8, 88.7, 102.4, 118.3, 136.5,
       157.6, 182.1, 210.2, 242.8, 280.3, 323.7, 373.8, 431.6, 498.5, 575.6)

dEIn_dlogDp_Vector <- as.vector(t(SizeDistrData)) # Unwrap number EI size distribution matrix into a vect
or

Dp_Vector <- as.vector(t(matrix(rep(Dp,275), nrow = 275, ncol = length(Dp), byrow=T))) # Create a vector
corresponding to size distribution data

dEIv_dlogDp_Vector <- dEIn_dlogDp_Vector*pi/6*(Dp_Vector*1e-6)^3 # Compute volume EI size distribution vec
tor

mydata = data.frame(Dp_Vector, dEIn_dlogDp_Vector, dEIv_dlogDp_Vector) # Combine vectors into a data fram
e.
```

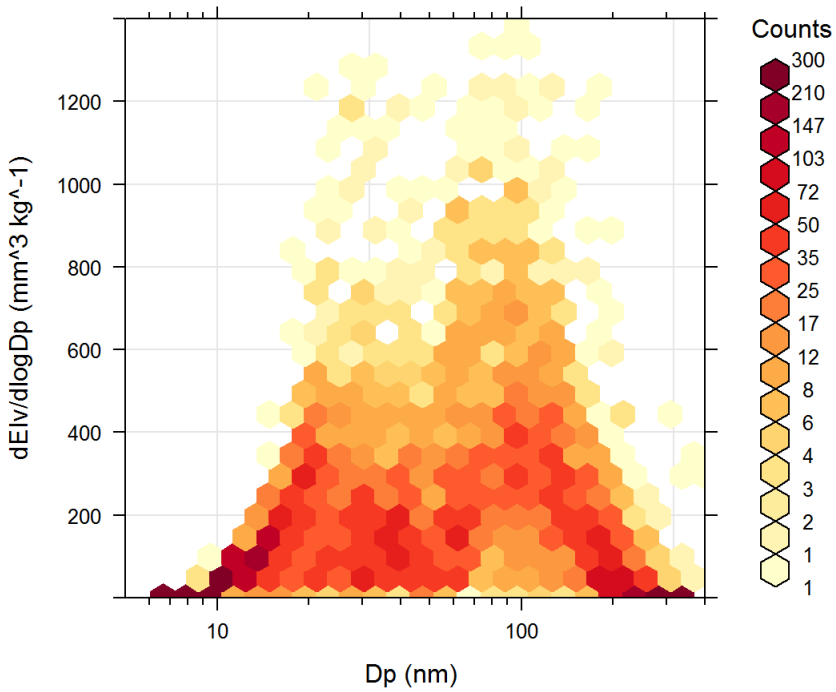
Aerosol number size distribution density plot.

```
plt <- scatterPlot(mydata, x="Dp_Vector", y="dEIn_dlogDp_Vector", method="hexbin", log.x=TRUE, log.y=TRUE
, mincnt=1, maxcnt=300, col="heat", xlim=c(5,400), ylim=c(1e11,1e18), xbin=25, xlab="Dp (nm)", ylab="dEIn
/dlogDp (kg^-1)")
```



Aerosol volume size distribution density plot.

```
pltv <- scatterPlot(mydata, x="Dp_Vector", y="dEIv_dlogDp_Vector", method="hexbin", log.x=TRUE, mincnt=1,
maxcnt=300, col="heat", xlim=c(5,400), ylim=c(0,1400), xbin=25, xlab="Dp (nm)", ylab="dEIv/dlogDp (mm^3 k
g^-1)")
```


dElv/dlogDp (mm^3 kg^-1)
1200
1000
800
600
400
200
10
100
Dp (nm)
Counts
300
210
147
103
72
50
35
25
17
12
8
6
4
3
2
1
1