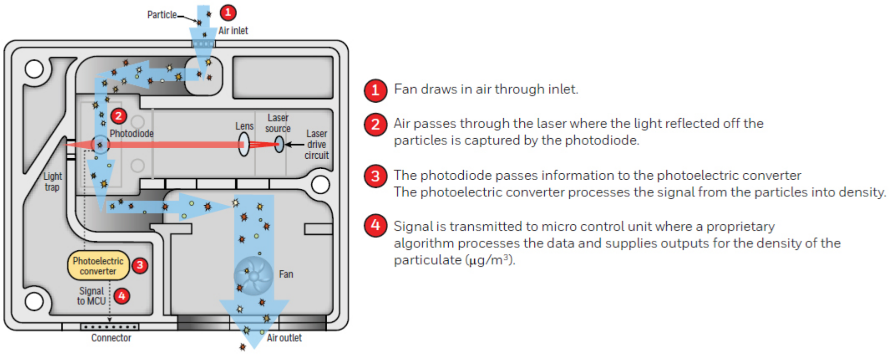


1 Fan draws in air through inlet.

2 Air passes through the laser where the light reflected off the particles is captured by the photodiode.

3 The photodiode passes information to the photoelectric converter The photoelectric converter processes the signal from the particles into density.

4 Signal is transmitted to micro control unit where a proprietary algorithm processes the data and supplies outputs for the density of the particulate ($\mu g/m^3$).

Figure 1: Schematic of Honeywell Particulate Matter Sensor. A laser light source illuminates a particle as it is pulled through the detection chamber. As particles pass through the laser beam, the light reflects off the particles and is recorded on the photo or light detector. The light is then analyzed and converted to an electrical signal to calculate particle concentration.

1 Introduction

1.1 How is particulate matter analyzed?

The PM sensor has several components to measure the particles in the air – remember, we don't know what the make up of these particles might be – but we can get an estimate of their size distribution. We have written the code to collect particles smaller than 1μm, 2.5μm and 10μm.

Figure 1 is a schematic of the particle sensor made by Honeywell (HPM Series — Particulate Matter Sensors 32322550HP). The one we are using is made by Plantower. As a Chinese company, most of their literature is in Chinese, so I didn't find something that I could interpret for us.

Our sensors can generate several categories of PM data. Using a laser and light defraction, the number and size of the particles can be estimated (Figure 2).

2 Data Collection

2.1 What data is collected by the PMS5003?

The PMS5003 generates the following data:

Data 1 refers to PM1.0 concentration unit $\mu g/m^3$CF=1standard particle)[1]

Data 2 refers to PM2.5 concentration unit $\mu g/m^3$CF=1standard particle

Data 3 refers to PM10 concentration unit $\mu g/m^3$CF=1standard particle

[1] The Asterisk on Data 1 and Data 4 is defined on the same page as "Note: CF=1 should be used in the factory environment"

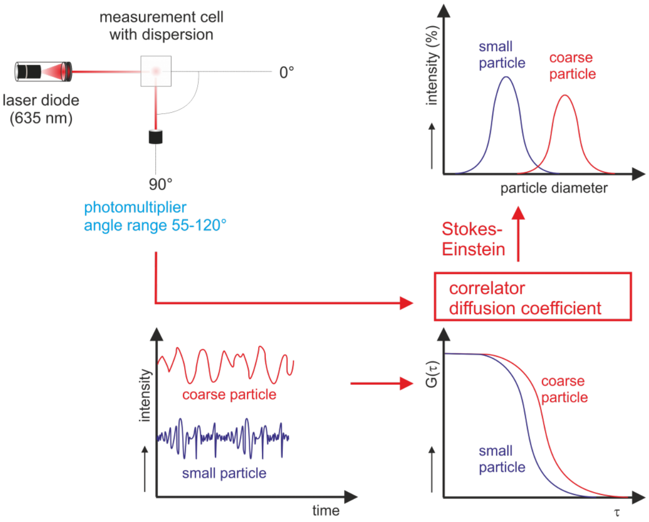


Figure 2: In particle size measurements using light scattering (DLS), a laser beam is scattered on very small, finely dispersed particles in a highly diluted matrix. The scattered light of each particle will then interfere with each other. Since the particles constantly change locations, the position of the scattering centers changes with respect to each other and the interferences lead to small fluctuations in scattering intensity (this explains the name "dynamic" light scattering).

Data 4 refers to PM1.0 concentration unit * μg/m^{3}under atmospheric environment

Data 5 refers to PM2.5 concentration unit μg/m^{3}under atmospheric environment

Data 6 refers to concentration unit (under atmospheric environment) μg/m^3

Data 7 indicates the number of particles with diameter beyond 0.3 μm in 0.1 L of air.

Data 8 indicates the number of particles with diameter beyond 0.5 μm in 0.1 L of air.

Data 9 indicates the number of particles with diameter beyond 1.0 μm in 0.1 L of air.

Data 10 indicates the number of particles with diameter beyond 2.5 μm in 0.1 L of air.

Data 11 indicates the number of particles with diameter beyond 5.0 μm in 0.1 L of air.

Data 12 indicates the number of particles with diameter beyond 10 μm in 0.1 L of air

The first set are labeled “standard”, while the second set are labeled “atmospheric environment”. What do these measure and how do the “standard” ones differ from the “atmospheric environment” ones?

It has to do with the density of the air used for calculations. ”Standard” refers to the concentration “corrected” to the “standard atmosphere” which in the US is DEFINED as “having a temperature of 288.15 K at the sea level 0 km geo-potential height and 1013.25 hPa”. NOTE: 288.15 K equals 25 C and Pa are “pascals” which is a meausure of pressure.

On the other hand, the “ambient conditions” are just as the air is “now” (whatever temperature and pressure there is). Now what does that mean. . .

Air being a gas, it is compressible which means that it changes its volume when the pressure changes so when you report concentrations as mass per volume of air it is relevant at what pressure that volume is calculated. For example, if you have a bunch of particles rising in the air in a bubble (no loss of particles, no addition, they’re just riding a bubble up in the air) then, as they rise, the pressure drops so what was 1cc at the ground it is now 2cc so the concentration of what was in the bubble is now half without anything actually changing other than the ambient pressure. So, it is common to report concentrations (of anything) as “x mg per standard m3” and because we scientist don’t like to write much (current example excluded) you’ll usually see the “standard” being dropped because it is “implicit”.

For gases it is also common to report concentrations as “parts per million” or “ppm” and that metric is independent of the volume of air as it represents the number of molecules of the gas in a million molecules of air (including the target gas).

In conclusion, I use the "standard" readings for reporting but keep the "ambient conditions" for analysis. I haven't used these things in high altitudes so for all my deployments the standard and ambient are very similar. I have written code to extract the standard values only – but you can certain add the ambient if you want to see if there is a difference – could be important if you are above 1000 ft in elevation.

3 Collecting the data

3.1 Read SD Card

Once the data have been collected, you can extract the data from the SD card and upland to Rstudio for processing. Shut down the Pi and read the SD card to collect all of the Air_Quality.csv files created. It might be useful to put them together using excel, but we can do that easily enough using R.

3.2 Processing the data

We create a script to process the data and allow you to create a reasonably user-friendly dataframe to analyze the data.

```
filepath.csv = "/home/CAMPUS/mwl04747/github/EJnPi/data/Air_Quality.csv"
rawdata = read.csv(filepath.csv)

names(rawdata)= c("X1", "X2", "Month", "Day", "Hour",
    "Minute", "Second", "X3", "X4", "pm1_cf", "X5", "pm25_cf", "X6",
    "pm10_cf", "X7", "pm1", "X8", "pm25", "pm10.", "X9")

rawdata$pm1_cf = as.numeric(gsub('[)]', '', rawdata$pm1_cf))
rawdata$pm25_cf = as.numeric(gsub('[)]', '', rawdata$pm25_cf))
rawdata$pm10_cf = as.numeric(gsub('[)]', '', rawdata$pm10_cf))
as.Date(with(rawdata, paste("2020", Month, Day,sep="-")), "%Y-%m-%d")
```

```
## [1] "2020-10-18" "2020-10-18" "2020-10-18" "2020-10-18" "2020-10-18"
## [6] "2020-10-18" "2020-10-18" "2020-10-18"
```

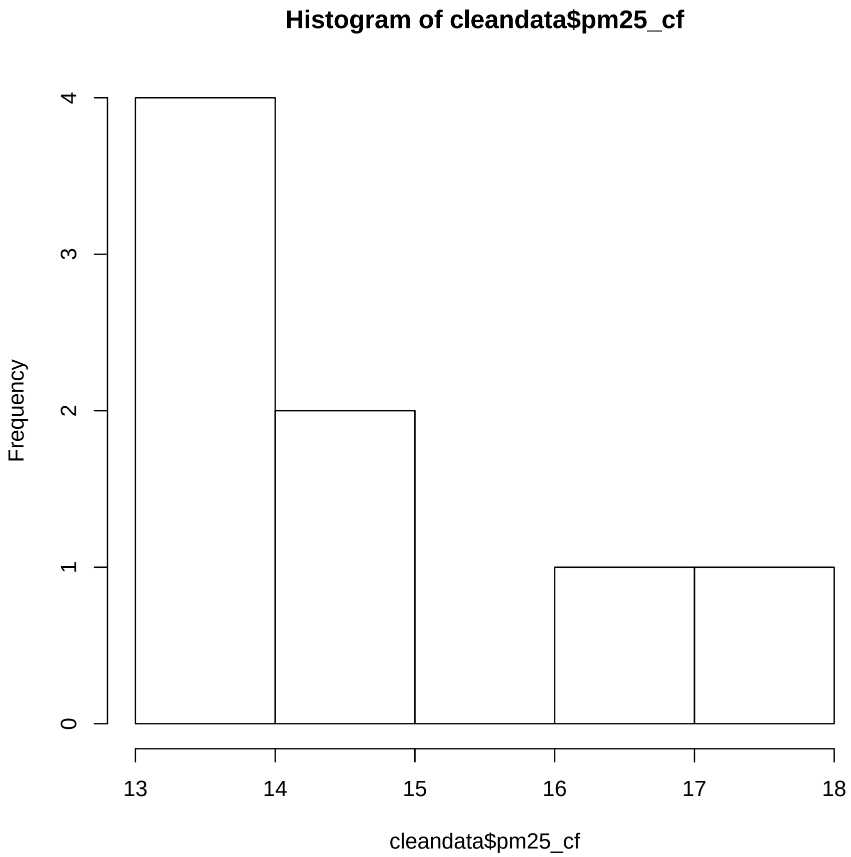
```
library(lubridate)
```

```
##
## Attaching package:  'lubridate'
## The following object is masked from 'package:base':
##
##     date
```

```
rawdata$DateTime = with(rawdata, ymd_hms(paste("2020", Month, Day, Hour, Minute, Second, sep
## Warning:  4 failed to parse.

head(rawdata)

##                          X1                      X2 Month Day Hour Minute Second
## 1 OrderedDict([('DateTime'  datetime.datetime(2020    10   18   21     27     31
## 2 OrderedDict([('DateTime'  datetime.datetime(2020    10   18   22     27     41
## 3 OrderedDict([('DateTime'  datetime.datetime(2020    10   18   23     27     21
## 4 OrderedDict([('DateTime'  datetime.datetime(2020    10   18   24     27     31
## 5 OrderedDict([('DateTime'  datetime.datetime(2020    10   18   25     27     41
## 6 OrderedDict([('DateTime'  datetime.datetime(2020    10   18   26     27     21
##          X3         X4 pm1_cf          X5 pm25_cf          X6 pm10_cf      X7
## 1  546422))  ('pm1_cf'      9  ('pm25_cf'      15  ('pm10_cf'      17  ('pm1'
## 2  910120))  ('pm1_cf'     11  ('pm25_cf'      14  ('pm10_cf'      12  ('pm1'
## 3  185161))  ('pm1_cf'     13  ('pm25_cf'      14  ('pm10_cf'      16  ('pm1'
## 4  546422))  ('pm1_cf'     12  ('pm25_cf'      15  ('pm10_cf'      19  ('pm1'
## 5  910120))  ('pm1_cf'      9  ('pm25_cf'      14  ('pm10_cf'      16  ('pm1'
## 6  185161))  ('pm1_cf'     11  ('pm25_cf'      18  ('pm10_cf'      16  ('pm1'
##   pm1       X8 pm25    pm10.     X9            DateTime
## 1  9)  ('pm25'  15)  ('pm10'  17)]) 2020-10-18 21:27:31
## 2  9)  ('pm25'  14)  ('pm10'  16)]) 2020-10-18 22:27:41
## 3  9)  ('pm25'  14)  ('pm10'  16)]) 2020-10-18 23:27:21
## 4  9)  ('pm25'  15)  ('pm10'  17)]) 2020-10-19 00:27:31
## 5  9)  ('pm25'  14)  ('pm10'  16)])                <NA>
## 6  9)  ('pm25'  14)  ('pm10'  16)])                <NA>

# Remove Variables
cleandata = subset(rawdata, select=c(DateTime, pm1_cf, pm25_cf, pm10_cf))
```

3.3 Plot Data

```
hist(cleandata$pm25_cf)
```



4 Comparing Data to EPA Stations

4.1 Using previous collected data

4.2 Comparing μ to EPA Data

The population mean (monthly average from the EPA data) is a parameter estimate. We can determine if the values you collected fall within the parameter estimate and it's 95% confidence intervals.

4.3 More Intricate Analysis

You could look to see if the there is a different during the time of the day or day of the week. You could move you sensor around in and outside your house and see if air quality is better / worse or the same. . . .

We can help you with r code if you'd like to follow up with these additional questions.