SUPERCOMPUTING CHALLENGE

ROBOTIC AIR QUALITY MONITOR:

SNOOPY

Team 12 members:

Joshua Mari Tamarra

Zachariah Burch

Isel Aragon

Britny Marquez

Teacher Sponsor:

Barbara Teterycz

Mentors:

David Ritter

Chris Karr

March, 27 2022

Table of Contents

Abstract/Executive Summary

Due to current events, such as global warming causing the increase of carbon dioxide in the air as well as pandemic and the risk of catching deadly viruses, we were curious if there was a way to monitor air quality in their surrounding areas, such as school and homes. We strongly believed that knowing what's in the air could then lead to the best possible solutions and a much safer and healthier environment.

For this reason, we were meeting with a retired professional from Silicon Valley, and together we developed a robotic air quality monitoring system, called Snoopy. In order to build such a robot, we needed the following hardware: Arduino Mega microcontroller board, two stepper motors, two drive wheels, pivot wheel, BlueFruit drive for Bluetooth, Real Time Clock, SGP30 chemical sniffer/sensor, PMSA003 dust & pollen sensor, SD driver, SD card, battery holder with connector and plug, batteries, and wall-mounted power supply. After connecting all these parts together, we then worked with our mentor on programming them in an Arduino coding environment using C++ language.

Thanks to all these efforts, intense learning, and wonderful mentorship, each of us took our own Snoopy home in order to measure the quality of the air and to collect data, which we then analyzed and compared with each other as well as presented graphically using another programming language, Python. In order to better understand our data, we also did research about the particles in the air, and we learned that increased amounts of some of them may have very harmful effects on human's health.

As a result of this project, we found out that there is very good air quality at school, but not at our homes. Because of this, we propose the following solutions to improve the quality of air: open the window to allow CO2 and TVOC escape from the house, replace carpet with tiles, use ecological paints, use air purifiers with HEPA filters, and invest in a good ventilation system.

Hypothesis

Due to current events, such as global warming causing the increase of atmospheric carbon dioxide as well as pandemic and its constantly increasing COVID cases, there is a big chance that the air quality isn't sufficient, and should be improved. In order to confirm this hypothesis, a programmable air quality monitoring system should be used to identify the circumstances that contribute to either worsening or improving the quality of air. We predict that simply opening windows and doors as well as investing in a good air purification and ventilation systems should contribute to decreasing the level of carbon dioxide and other toxic gasses in the air, but at the same time these remedies (especially opening the window) might increase the level of pollen.

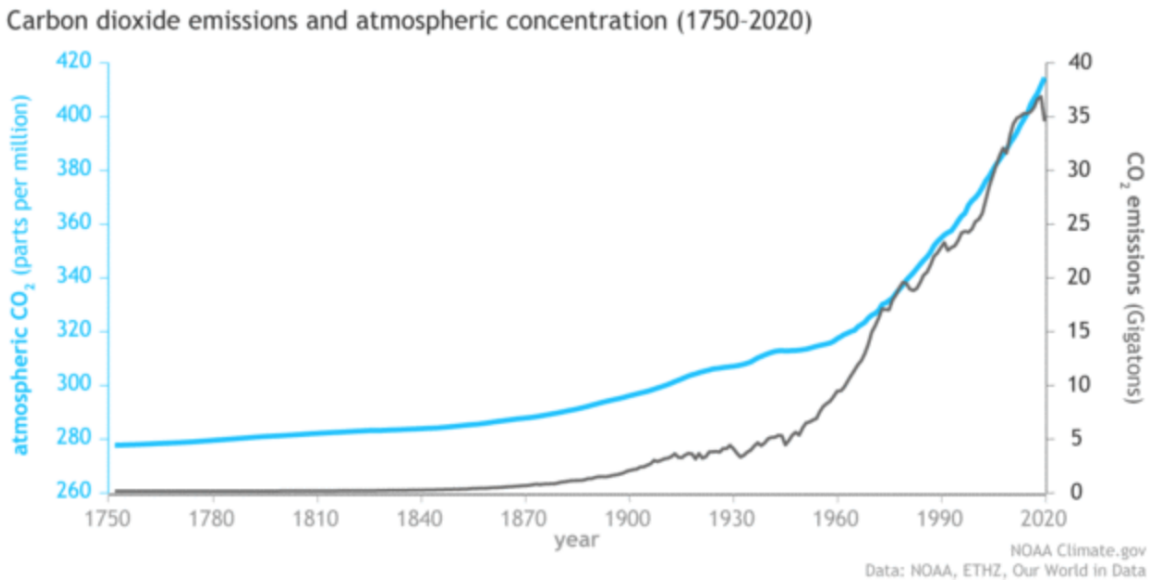
Identify the Problem

Problem Background and Research

Carbon Dioxide (CO2)

Carbon dioxide is made of one carbon atom and two oxygen atoms, and is a naturally occurring gas in the atmosphere, without which plants would not be able to survive, and our planet would be too cold after the sunset. Carbon dioxide, as a greenhouse gas, traps heat in the atmosphere keeping our planet warm (Smith, 2019). However, due to anthropogenic CO2 emissions, caused by human activities, such as burning fossil fuels for energy, recent outdoor carbon dioxide levels are the highest they've ever been.

According to the article published at climate.gov, “fossil fuels like coal and oil contain carbon that plants pulled out of the atmosphere through photosynthesis over many millions of years; and we are returning that carbon to the atmosphere in just a few hundred years” (Lindsey, 2020). As shown on the chart below, since the start of the Industrial Revolution in 1750, the amount of carbon dioxide in the atmosphere (blue line) has increased along with human emissions (gray line) and currently it has already exceeded 400 ppm (particles per million parts of air). That is why the lowest possible level of carbon dioxide that has been detected by Snoopy never dropped below 400 ppm.



In addition to atmospheric carbon dioxide, there is also indoor CO2, which is produced by our own bodies as a byproduct of respiration when we exhale as well as during cooking, burning candles, incense, wood in fireplace, and smoking. And if there is poor ventilation in our houses that are tightly sealed to save energy, CO2 is trapped and builds up to unhealthy levels. At around 1000 ppm, the residents may start to experience fatigue, sleepiness, and problems with concentration. With prolonged exposure to higher concentration of CO2, people will experience additional and more severe health issues, such as headache, drowsiness, tiredness, dizziness, sweating, difficulty breathing, and even seizures and loss of consciousness when the level of

indoor CO2 is very high. The table below shows the examples of health effects caused by the specific levels of CO2 concentration (Smith, 2019).

CO2 Concentration	**Health Effects**
< 1000 ppm	Limited or no health effects
1000 ppm - 2500 ppm	Fatigue, loss of focus and concentration, uncomfortable 'stuffy' feeling in the air
2500 ppm - 5000 ppm	Headache, drowsiness, tiredness
5000 ppm - 40000 ppm	Violates OSHA requirements, severe headaches, slight intoxication depending on the exposure time
40000 ppm - 100000 ppm	IDLH (Immediately dangerous to life or health), dizziness, increased heart rate, sweating, difficulty breathing; seizures and loss of consciousness after prolonged exposure
> 100000 ppm	Loss of consciousness within minutes, coma, risk of death

Total Volatile Organic Compounds (TVOC)

Total Volatile Organic Compounds are multiple organic chemicals that become a gas at room temperature. Many VOCs come from cleaners and disinfectants, air fresheners, fragrances, paints and solvents, glue, plywood, candles and fires, cooking fumes, new furniture and carpets, electronic devices, etc. TVOC can be measured in micrograms per cubic meter (µg/m3) of air (or milligrams per cubic meter (mg/m3), parts per million (ppm) or parts per billion (ppb)). Past recommendations from the USEPA (United States Environmental Protection Agency, 2021) and indoor environment rating schemes is that a recommended limit of 500 ppb TVOC and less than 250 ppb of any one VOC is appropriate in average office environments (The World Green Building Council, 2020). The table below shows the recommended actions that should be taken for a given ppb level of TVOC:

0 - 250 ppb	The VOC contents in the air are low.
250 - 2000 ppb	Look for VOC sources if this average level persists for a month.
> 2000 ppb	The VOC contents are very high - consider taking action/ventilating right now.

Some VOCs are bad for health, especially during long-term exposure in large doses. Immediate symptoms that some people have experienced soon after exposure to VOCs are eye and respiratory tract irritation, headaches, dizziness, visual disorders and memory impairment. Some VOCs as formaldehyde (used in making building materials) can cause cancer (Advanced Solutions Nederland B.V., 2020).

Pollen

Some plants, including various kinds of trees, grasses, and weeds, make a fine powder called pollen that's light enough to travel through the air in order to reproduce. More than 25 million Americans are allergic to pollen, which means that their white blood cells release a chemical, called histamine, when their immune system is defending against allergen. This can result in allergic reactions, such as itchy throat; red, itchy, watery eyes; runny or stuffy nose; sneezing; wheezing or coughing (Fields and DerSarkissian, 2021).

In addition to getting medical help, many people try to avoid going out and stay indoors instead. However, as our Snoopy found out, keeping the window always closed, traps CO2 and TVOC inside and makes the indoor air even more dangerous and toxic.

The Size of Particles

Due to the microscopic size of the particles in the air, in order for Snoopy to detect and correctly classify them, it was necessary to install two kinds of sensors, one for CO2 and TVOC

and the other one for pollen and dust.

As shown here, pollen, salt, and sand are significantly larger than viruses or bacteria. Because of their higher relative sizes, our body is usually able to block them out—a particle needs to be smaller than 10 microns before it can be inhaled into our respiratory tract. Because of this, pollen or sand typically get trapped in the eyes, nose, and throat, before they enter our lungs. The smaller particles (e.g. viruses or wildfire smoke) however, are able to slip through more easily and cause even more dangerous health risks.

While allergies to pollen worsen the quality of life and are very inconvenient (sometimes even making it difficult to see), the smaller particles, like viruses, are even more dangerous for human health. However, air pollution caused by particulate matter (such as dust, dirt, soot, and smoke particles) has even more chances to enter human lungs. For example, wildfire smoke at just a fraction of the size between 0.4-0.7 microns, has been identified as the key factor in not just respiratory issues, but also cardiovascular and neurological problems (Ang et al., 2020).

Because of all these harmful air particles, it's very important especially now, during the time of pandemic and global warming resulting in frequent wildfires, to constantly care for the

best air quality possible in both public and private places.

Work Done By Others

Since building the robot and programming all its components required thorough knowledge of C++ language and even some electrical engineering, such as designing electrical circuits and connectors, we were not able to build it on our own. That is why we were working with a professional mentor with almost 40 years of experience working at Silicon Valley as an electrical engineer and also C/C++ programmer.

After collecting and analyzing data, we then worked with another mentor, a professional computer programmer, who helped us apply linear regression in a computational model for the purpose of further statistical analysis.

Constraints

Because of the risk with COVID, we were working with our engineering mentor remotely, which wasn't always easy, especially when we had to connect the physical parts together. In addition, there were many electrical concepts, which we were not familiar with, and which were hard for us to comprehend without professional experience.

Another limitation, which we had, was caused by the COVID-related guidelines, which required us to always have air purifiers on and windows open during school hours. Because of this, we were unable to measure how air quality was affected by the presence of students in class when the air purifier was turned off and windows were closed.

Goal

The goal of this project is to have a correctly working air quality monitoring system that can detect and record increased levels of particulates, such as pollutants and allergens in the air. This would let us know how safe/unsafe the environment, in which we live, work, study, and sleep, is. For example, if there is a low level of CO2, it means that there is very good ventilation, which is especially important and beneficial now, during the pandemic. Knowing what's in the air might also help people suffering from headaches, dizziness, sleepiness, problems with concentration and allergies identify the source(s) of their health issues.

Brainstorming

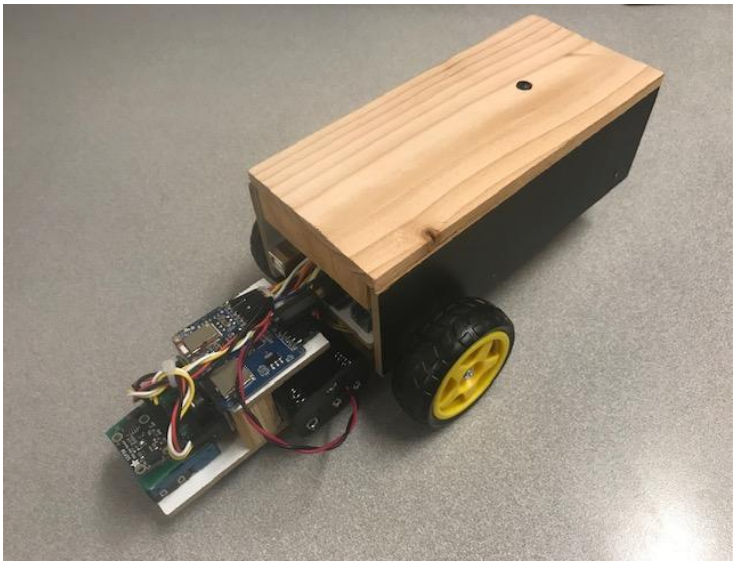
Idea Generation / Selected Approach

As high school students with no prerequisites in computer science, we first wanted to build a robot. We then realized that for this kind of a challenge, our robot should be able to do something practical and useful rather than just only move around. Since our teacher-sponsor was suffering from allergies to pollen, we came up with the idea to have our robot detect it. Our mentor also advised us to add another sensor that would detect the level of carbon dioxide and other chemicals, such as volatile organic compounds. This way, our robot would measure the overall air quality, not just pollen. Thanks to all these ideas, we strongly believed that knowing what's in the air could lead to a much safer and healthier environment.

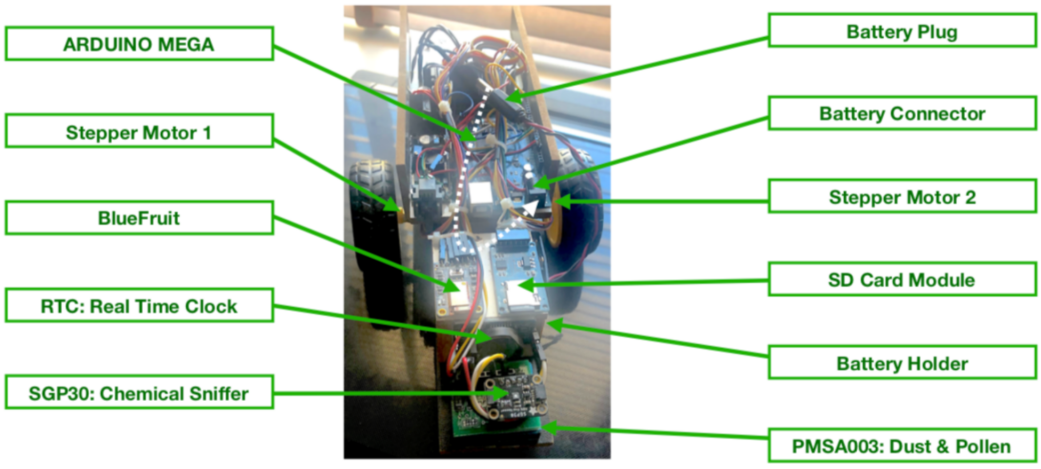
Model/Prototype

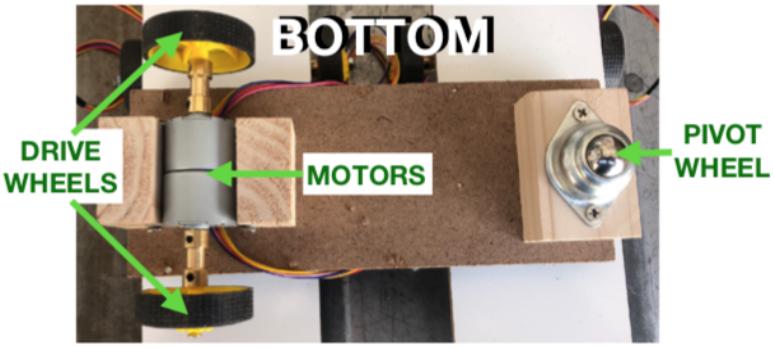
Description of Model

While working with our mentor from Silicon Valley, we have developed a robotic air quality monitoring system, called Snoopy:



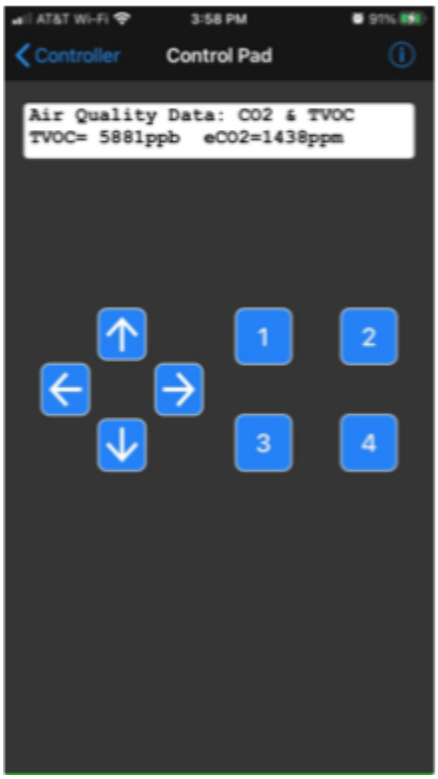
In order to build such a robot, we needed the following hardware:



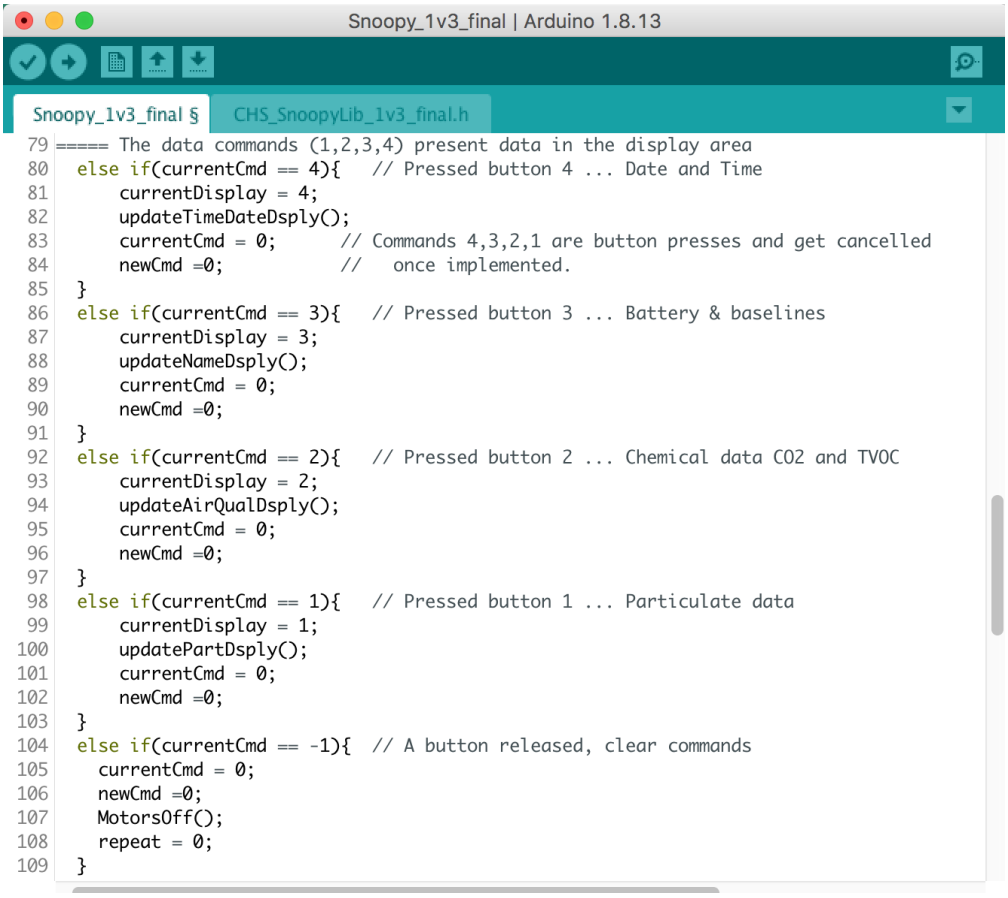


After connecting all the parts of Snoopy together, some of us worked in create.arduino.cc using our school chromebooks, and some others installed the Arduino coding environment on their computers. We were participating in many lectures about using C++ language to program all the parts of Snoopy. Many of the concepts were too advanced for us, but in general, what we've learned was that in order to complete the program, almost each part of Snoopy required a special library that had to be imported at the top of the program using #include command with the name of the library and .h extension enclosed by the less and greater than symbols. Next, we've learned that all the variables had to be defined together with their kind and data type just under the libraries and before the main part of the program, like this:

We also learned how to connect Snoopy to the Bluefruit app installed on our cell phones via Bluetooth, and how to control its movement when each of the following arrow buttons was pressed:

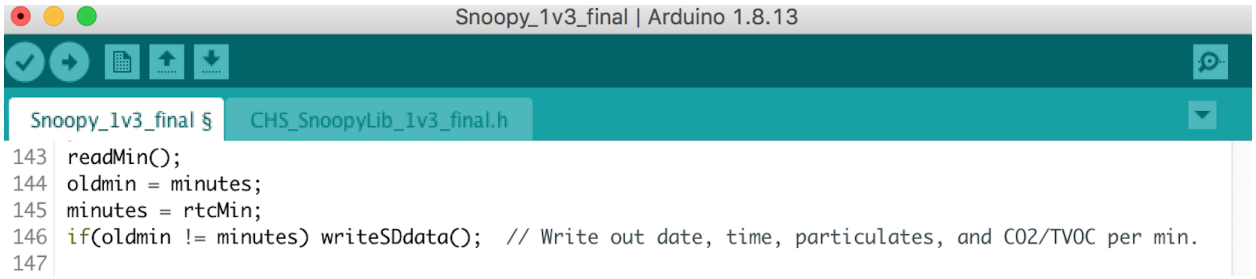


In addition, we've learned how to program each of the four buttons to display data on the screen of the Bluefruit's app, such as button 1, which when pressed, displays the level of pollen and dust in the air; button 2 displays the level of chemicals, such as CO2 and TVOC, button 3 displays the level of battery, and button 4 displays the date and time, as shown on the screenshot below:




```
79 ===== The data commands (1,2,3,4) present data in the display area
80   else if(currentCmd == 4){   // Pressed button 4 ... Date and Time
81       currentDisplay = 4;
82       updateTimeDateDsply();
83       currentCmd = 0;      // Commands 4,3,2,1 are button presses and get cancelled
84       newCmd =0;           //   once implemented.
85   }
86   else if(currentCmd == 3){   // Pressed button 3 ... Battery & baselines
87       currentDisplay = 3;
88       updateNameDsply();
89       currentCmd = 0;
90       newCmd =0;
91   }
92   else if(currentCmd == 2){   // Pressed button 2 ... Chemical data CO2 and TVOC
93       currentDisplay = 2;
94       updateAirQualDsply();
95       currentCmd = 0;
96       newCmd =0;
97   }
98   else if(currentCmd == 1){   // Pressed button 1 ... Particulate data
99       currentDisplay = 1;
100      updatePartDsply();
101      currentCmd = 0;
102      newCmd =0;
103  }
104  else if(currentCmd == -1){  // A button released, clear commands
105    currentCmd = 0;
106    newCmd =0;
107    MotorsOff();
108    repeat = 0;
109  }
```

And finally, we've also learned how to write data gathered by Snoopy every minute to the SD card for the future analysis and graphical representation:

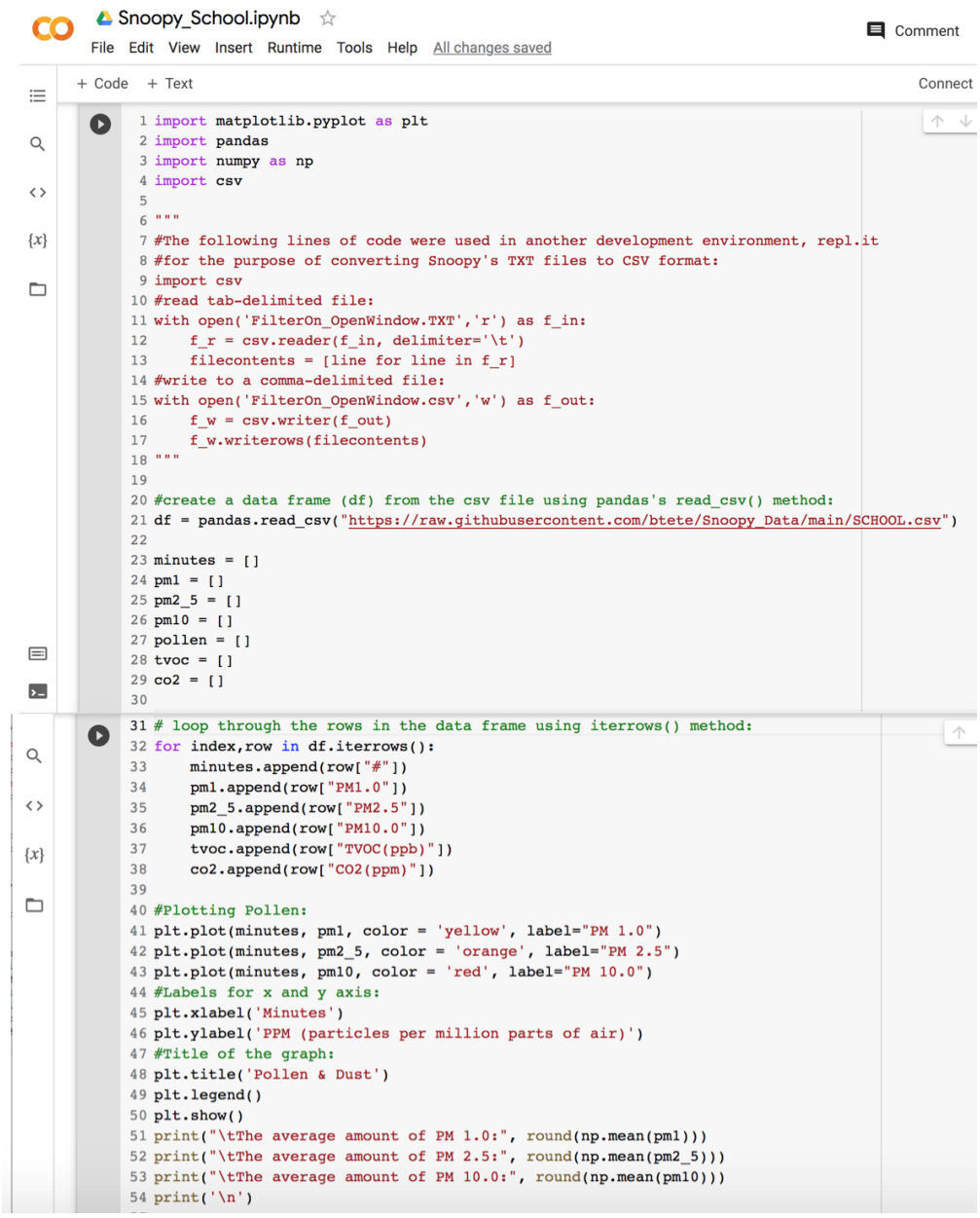



```
143 readMin();
144 oldmin = minutes;
145 minutes = rtcMin;
146 if(oldmin != minutes) writeSDdata();  // Write out date, time, particulates, and CO2/TVOC per min.
147
```

In this part of the program, the time was recorded by rtc (real time clock) and constantly compared to its previous record. When there was a difference in time, data detected by the sensors were then written to the SD card.

Thanks to all these efforts, intense learning, and wonderful support, which we've received from our mentor, all of us took one copy of Snoopy home in order to measure the

quality of their air and to collect data, which we then analyzed and compared with each other as well as presented graphically using another programming language, Python:



As a result of this collaborative work, we are able to measure, collect, analyze, present, research and explain the sources of air pollutants, and even propose some solutions to improve the quality of air.

Test model and evaluate

Troubleshooting, Testing & Redesigning

Snoopy let us investigate the situation in both school and home. However, what we've noticed after collecting data was that Snoopy almost always started with indicating the lowest level of CO2. We then learned that it was because it was the default base level, and that it usually took a while until Snoopy started sniffing real values. We've also noticed sometimes some sudden outliers in data, which could be caused by either sudden air movement or some kind of recalibration happening inside Snoopy.

In addition, once we gathered our data, we found out that it was hard to balance the level of CO2 and TVOC with the level of pollen and dust in the indoor air. When we were able to lower CO2 and TVOC by simply opening the window, then the level of pollen went drastically up. When we closed the window, we got the opposite effect: the level of CO2 and TVOC went up, whereas the level of pollen and dust went down. It forced us to look for some kind of a compromise between the amounts of these two major air pollutants and allergens.

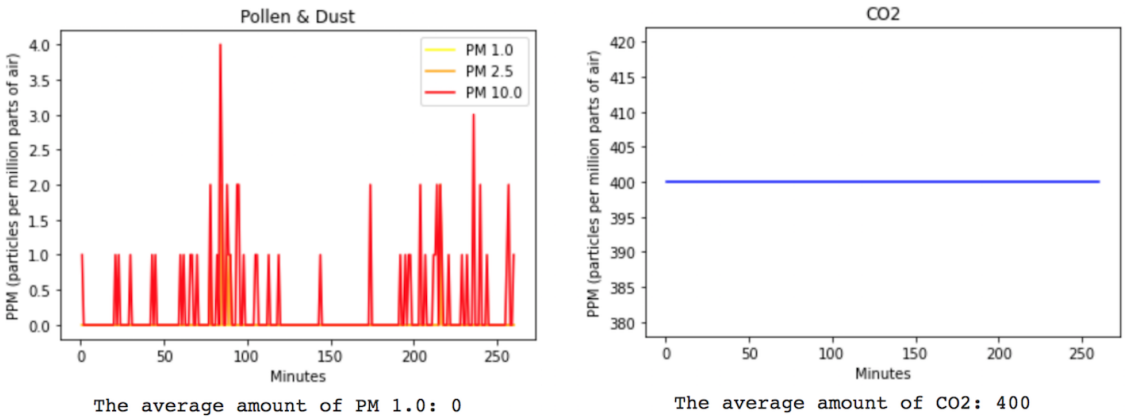
Another difficulty, which we faced while gathering data, was their inconsistency caused by different seasons of the year (e.g. there was more pollen during Fall than during Winter time). We also noticed a discrepancy between daily data collected at school that might depend on whether or not the students wore perfumes and/or deodorants, used laundry detergents and softeners with fragrances, hair spray, etc.

And finally, due to COVID-related guidelines, we were only able to collect data when students were present in class while the air purifier was on and windows were open. Thus, our

data lack information about the influence of people on air quality when the air purifier was off and windows were closed.

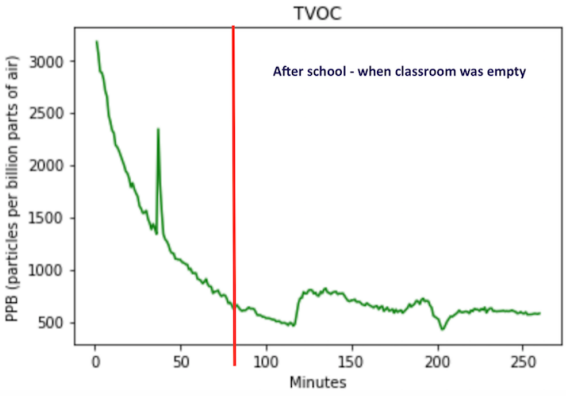
Because of the above issues, we decided to start with troubleshooting the performance of Snoopy by simply excluding the outliers and the first default, but not yet accurate, values from our data. According to the following data, which Snoopy gathered for us at school, both the level of pollen and dust as well as the level of CO2 were very low. However, the level of TVOC was first increased, especially when students were in class, even though the window was open and air purifier was turned on:

#	MM:DD:YYYY	HH:MM:SS	PM1.0	PM2.5	PM10.0	TVOC(ppb)	CO2(ppm)
1	11/10/2021	2:38: 0 PM	0	0	1	3173	400
2	11/10/2021	2:39: 0 PM	0	0	0	3052	400
3	11/10/2021	2:40: 0 PM	0	0	0	2888	400
4	11/10/2021	2:41: 0 PM	0	0	0	2879	400
5	11/10/2021	2:42: 0 PM	0	0	0	2819	400
6	11/10/2021	2:43: 0 PM	0	0	0	2704	400
7	11/10/2021	2:44: 0 PM	0	0	0	2651	400
8	11/10/2021	2:45: 0 PM	0	0	0	2471	400
9	11/10/2021	2:46: 0 PM	0	0	0	2408	400
10	11/10/2021	2:47: 0 PM	0	0	0	2327	400
11	11/10/2021	2:48: 0 PM	0	0	0	2304	400
12	11/10/2021	2:49: 0 PM	0	0	0	2187	400
13	11/10/2021	2:50: 0 PM	0	0	0	2168	400
14	11/10/2021	2:51: 0 PM	0	0	0	2133	400
15	11/10/2021	2:52: 0 PM	0	0	0	2087	400




```
The average amount of PM 1.0: 0
The average amount of PM 2.5: 0
The average amount of PM 10.0: 0
```

```
The average amount of CO2: 400
```

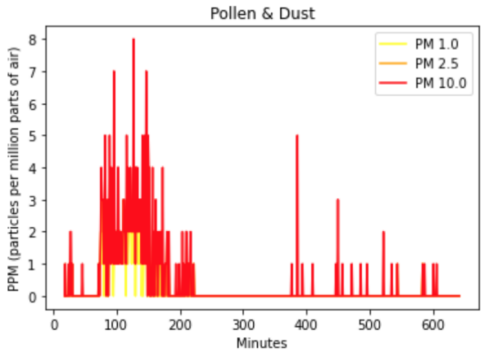



```
The average amount of TVOC: 885
```

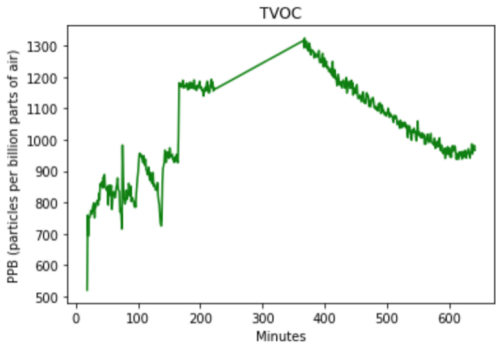
We suspect that this could be caused by the usage of deodorants, perfumes, hair sprays, clothes dried with sheets coated in lubricants and fragrances, etc. As shown on the graphs above, when students left the school and the classroom was empty, the level of both TVOC and CO2 remained at a lower level, even though the window was closed. It proves that our school has a very good ventilation system and was prepared for the pandemic very well.

We then used Snoopy at our houses. As we found out, the situation in all our homes was much worse. We suspect that the reason could be a much smaller area of our rooms than the area of our classrooms, the presence of kitchen, pets, and carpet in our houses, worse or no ventilation system, the lack of air purifiers (except for one of us), and close presence of neighbors and cars, as opposed to our classroom, which is far from the parking lot. The following results represent the level of pollen and dust, as well as CO2 and TVOC when the window was closed:

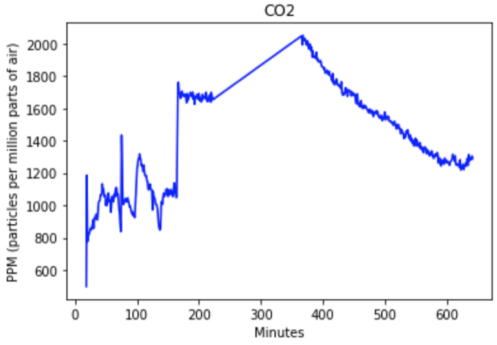
#	MM:DD:YYYY	HH:MM:SS	PM1.0	PM2.5	PM10.0	TVOC(ppb)	CO2(ppm)
18	9/11/2021	9:33: 0 PM	0	0	0	520	498
19	9/11/2021	9:34: 0 PM	0	1	1	760	1186
20	9/11/2021	9:35: 0 PM	0	0	0	693	775
21	9/11/2021	9:36: 0 PM	0	0	0	752	815
22	9/11/2021	9:37: 0 PM	0	0	0	751	815
23	9/11/2021	9:38: 0 PM	0	0	0	757	829
24	9/11/2021	9:39: 0 PM	0	0	0	774	844
25	9/11/2021	9:40: 0 PM	0	0	1	764	857
26	9/11/2021	9:41: 0 PM	0	0	0	764	848
27	9/11/2021	9:42: 0 PM	0	0	2	784	861
28	9/11/2021	9:43: 0 PM	0	0	2	793	881
29	9/11/2021	9:44: 0 PM	0	0	0	798	909
30	9/11/2021	9:45: 0 PM	1	1	1	751	857
31	9/11/2021	9:46: 0 PM	0	0	0	792	922
32	9/11/2021	9:47: 0 PM	0	0	0	785	924
33	9/11/2021	9:48: 0 PM	0	0	0	797	919
34	9/11/2021	9:49: 0 PM	0	0	0	804	945
35	9/11/2021	9:50: 0 PM	0	0	0	806	913
36	9/11/2021	9:51: 0 PM	0	0	0	792	910
37	9/11/2021	9:52: 0 PM	0	0	0	829	969
38	9/11/2021	9:53: 0 PM	0	0	0	809	1020
39	9/11/2021	9:54: 0 PM	0	0	0	859	1027




```
The average amount of PM 1.0: 0
The average amount of PM 2.5: 0
The average amount of PM 10.0: 1
```



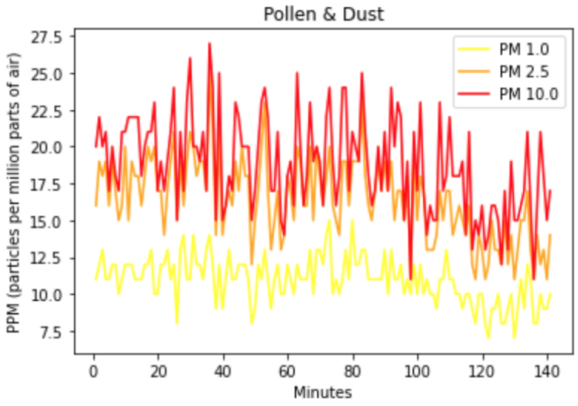

```
The average amount of TVOC: 1029
```



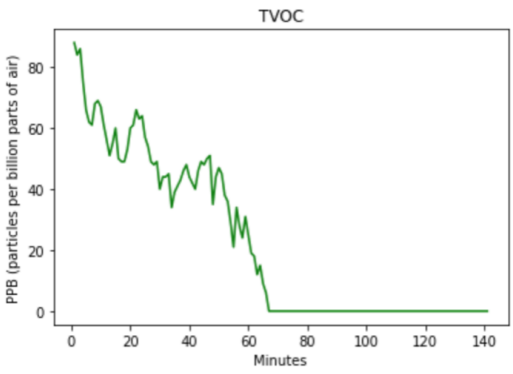

```
The average amount of CO2: 1407
```

After opening the window, both TVOC and CO2 decreased significantly, but the level of pollen and dust went drastically up:

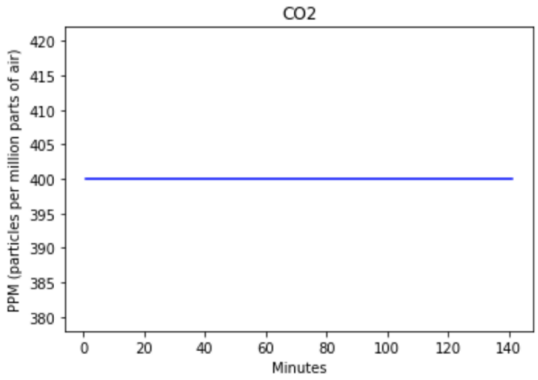
#	MM:DD:YYYY	HH:MM:SS	PM1.0	PM2.5	PM10.0	TVOC(ppb)	CO2(ppm)
1	9/12/2021	12:29: 0 AM	11	16	20	88	400
2	9/12/2021	12:30: 0 AM	12	19	22	84	400
3	9/12/2021	12:31: 0 AM	13	18	20	86	400
4	9/12/2021	12:32: 0 AM	11	19	21	75	400
5	9/12/2021	12:33: 0 AM	11	16	17	66	400
6	9/12/2021	12:34: 0 AM	12	19	20	62	400
7	9/12/2021	12:35: 0 AM	12	17	18	61	400
8	9/12/2021	12:36: 0 AM	10	15	17	68	400
9	9/12/2021	12:37: 0 AM	11	16	21	69	400
10	9/12/2021	12:38: 0 AM	12	20	21	67	400
11	9/12/2021	12:39: 0 AM	12	15	22	61	400
12	9/12/2021	12:40: 0 AM	12	19	22	56	400
13	9/12/2021	12:41: 0 AM	11	18	22	51	400
14	9/12/2021	12:42: 0 AM	11	18	22	55	400




```
The average amount of PM 1.0: 11
The average amount of PM 2.5: 16
The average amount of PM 10.0: 19
```




```
The average amount of TVOC: 22
```




```
The average amount of CO2: 400
```

Using Data To Improve Air Quality

Based on data collected by Snoopy, we realized that even without investing in an expensive ventilation system, we can still improve the quality of air by simply opening or closing windows and doors, and turning air purifiers on. The following data collected by Snoopy when the window was open, show how the level of pollen and TVOC changed when the air purifier with HEPA filter was first turned off and then on (with the open window, the level of CO2 was always at its lowest, regardless of whether the air purifier was turned on or off):

Filter Off:

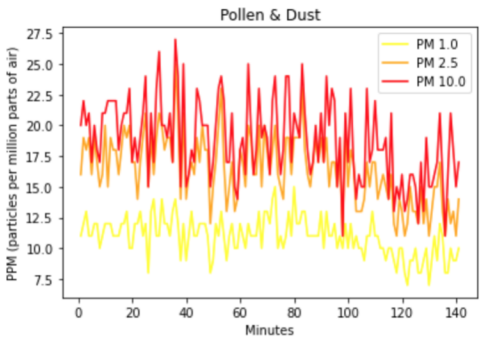
#	MM:DD:YYYY	HH:MM:SS	PM1.0	PM2.5	PM10.0	TVOC(ppb)	CO2(ppm)
1	9/12/2021	12:29: 0 AM	11	16	20	88	400
2	9/12/2021	12:30: 0 AM	12	19	22	84	400
3	9/12/2021	12:31: 0 AM	13	18	20	86	400
4	9/12/2021	12:32: 0 AM	11	19	21	75	400
5	9/12/2021	12:33: 0 AM	11	16	17	66	400
6	9/12/2021	12:34: 0 AM	12	19	20	62	400
7	9/12/2021	12:35: 0 AM	12	17	18	61	400
8	9/12/2021	12:36: 0 AM	10	15	17	68	400
9	9/12/2021	12:37: 0 AM	11	16	21	69	400
10	9/12/2021	12:38: 0 AM	12	20	21	67	400
11	9/12/2021	12:39: 0 AM	12	15	22	61	400
12	9/12/2021	12:40: 0 AM	12	19	22	56	400
13	9/12/2021	12:41: 0 AM	11	18	22	51	400
14	9/12/2021	12:42: 0 AM	11	18	22	55	400

Filter On:

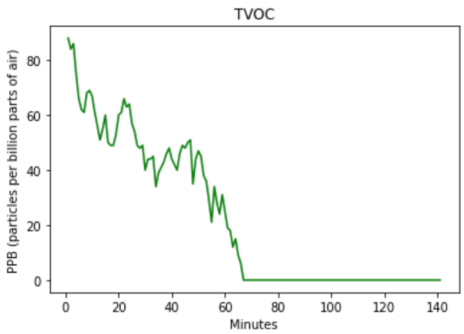
#	MM:DD:YYYY	HH:MM:SS	PM1.0	PM2.5	PM10.0	TVOC(ppb)	CO2(ppm)
1	9/12/2021	3:37: 0 PM	4	5	8	0	400
2	9/12/2021	3:38: 0 PM	4	7	8	0	400
3	9/12/2021	3:39: 0 PM	4	5	5	0	400
4	9/12/2021	3:40: 0 PM	4	7	7	0	400
5	9/12/2021	3:41: 0 PM	4	5	8	0	400
6	9/12/2021	3:42: 0 PM	4	5	6	0	400
7	9/12/2021	3:43: 0 PM	4	5	5	0	400
8	9/12/2021	3:44: 0 PM	4	5	8	0	400
9	9/12/2021	3:45: 0 PM	2	4	5	0	400
10	9/12/2021	3:46: 0 PM	4	5	8	0	400
11	9/12/2021	3:47: 0 PM	3	4	4	0	400
12	9/12/2021	3:48: 0 PM	2	3	3	0	400
13	9/12/2021	3:49: 0 PM	4	5	5	0	400
14	9/12/2021	3:50: 0 PM	3	4	4	0	400
15	9/12/2021	3:51: 0 PM	3	5	6	0	400

As shown on the charts below, the average level of pollen decreased about four times and the average level of TVOC dropped to 0 after the air purifier was turned on, which proves the effectiveness of HEPA filters.

Air purifier with HEPA filter turned OFF:

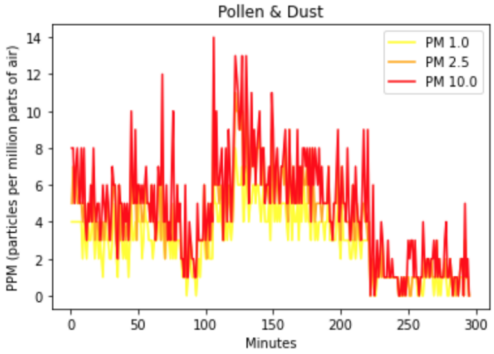



```
The average amount of PM 1.0: 11
The average amount of PM 2.5: 16
The average amount of PM 10.0: 19
```

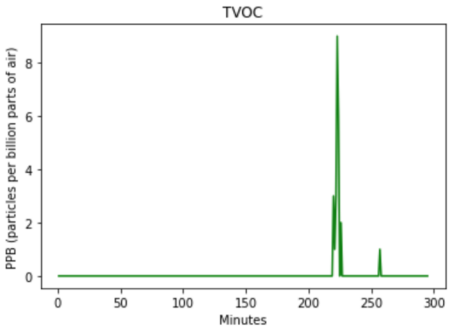



```
The average amount of TVOC: 22
```

Air purifier with HEPA filter turned ON:




```
The average amount of PM 1.0: 3
The average amount of PM 2.5: 4
The average amount of PM 10.0: 5
```




```
The average amount of TVOC: 0
```

Computational/Mathematical Model

In order to better understand and analyze the correlation between all the conditions, during which our data were collected, as well as to identify the most influential ones that either significantly decreased or increased the quality of air, we decided to make further statistical analysis on our data in the form of a computational model. After being introduced by our other (coding) mentor to linear regression, we've learned that it measures the relationship between the

input variables (x), also known as independent variables, and the single output variable (y), also known as dependent variable (Brownlee, 2020). In order to better see how each condition affected the quality of air, we decided to use this mathematical concept in our computational model. We learned that a variety of techniques can be used to prepare the linear regression equation from data, and that the most common one is Ordinary Least Squares Linear Regression (Brownlee, 2020). Its purpose is to find the values of coefficients b (y-intercept) and m (slope of the line) used in the linear function together with independent variables x and dependent variable y, for which the sum of the squared distances (also called errors) between the actual data points and the line of such function is the least. And finally, in order to assess how well a regression model fits a dataset, it is necessary to also calculate the root mean square error (RMSE), which indicates the average distance between the predicted values from the model and the actual data points. The lower the RMSE, the better a model fits a dataset (Zach, 2021).

Since two of the students in our team have not taken calculus class yet, we first decided to study the mathematical concepts and equations used in this kind of linear regression model:

Linear function: $y = b + mx$

Where:

$y \rightarrow$ dependent variable

$b \rightarrow$ y-intercept

$m \rightarrow$ slope of the line

$x \rightarrow$ independent variable

For the purpose of our model that uses many data points, we replaced the above formula with the following:

$q_1 = \beta_0 + \beta_1 p_1 + E_1$

…

$q_n = \beta_0 + \beta_1 p_n + E_n$

Where:
$q_1, q_2, \dots q_n \rightarrow$ actual data points
$\beta_0, \beta_1 \rightarrow$ constants (coefficients), think about them as b and m
$p_1, p_2, \dots p_n \rightarrow$ predicted points (points on the line)
$E_1, E_2, \dots E_n \rightarrow$ errors (distances from the line)
The above can be represented in this simplified form:

$$\vec{q} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \begin{bmatrix} 1 & p_1 \\ . & . \\ . & . \\ 1 & p_n \end{bmatrix} + \vec{E}$$

1) Since we are looking for the least errors, let's solve for E:

$$\vec{E} = \vec{q} - \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \begin{bmatrix} 1 & p_1 \\ . & . \\ . & . \\ 1 & p_n \end{bmatrix}$$

2) Let's calculate the norm (||...||) of E (a norm is a square root of the sum of the squared elements):

$$\|\vec{E}\| = \sqrt{E_1^2 + ... + E_n^2} = \sqrt{(q_1 - (\beta_0 + \beta_1 p_1))^2 + ... + (q_n - (\beta_0 + \beta_1 p_n))^2}$$

3) Let's square the norm to get rid of square root:

$$\|\vec{E}\|^2 = \sqrt{E_1^2 + ... + E_n^2}^2 = \sqrt{(q_1 - \beta_0 - \beta_1 p_1)^2 + ... + (q_n - \beta_0 - \beta_1 p_n)^2}^2$$

$$\vec{E} = (q_1 - \beta_0 - \beta_1 p_1)^2 + ... + (q_n - \beta_0 - \beta_1 p_n)^2$$

$$\vec{E} = \sum_i (q_i - \beta_0 - \beta_1 p_i)^2$$

We will now learn how to take partial derivative (∂) in respect to x (∂_y / ∂_x) in the following example: $y = ax^3 + bx^2 + x + 5$. In order to do so, use x's exponent as a constant value and multiply it by x, and then decrease this x's exponent by 1:

$\partial_y / \partial_x = 3ax^{3-1} + 2bx^{2-1} + 1x^{1-1} + 0*5x^{0-1} = 3ax^2 + 2bx + 1 + 0$; Now, back to our problem:

4) Take partial derivative in respect to β_0:

$y = (q_i - \beta_0 - \beta_1 p_i)^2 = u^2$

$u = q_i - \beta_0 - \beta_1 p_i$

Let's apply the chain rule as shown in this video: https://www.youtube.com/watch?v=Ic_LW7K8eGE

$$\frac{\partial y}{\partial \beta_0} = \frac{\partial u}{\partial \beta_0} * \frac{\partial y}{\partial u}$$

$$\frac{\partial u}{\partial \beta_0} = 0 * q_i \beta_0^{0-1} - 1 * \beta_0^{1-1} - 0 * \beta_1 p_i \beta_0^{0-1} = -1$$

$$\frac{\partial y}{\partial u} =$$ Notice that $y = u^2$, so taking partial derivative in respect to u means placing the value of the exponent (2) on the left side of u and decreasing the actual exponent by 1:

$$\frac{\partial y}{\partial u} = (q_i - \beta_0 - \beta_1 p_i)^2 = 2\,(q_i - \beta_0 - \beta_1 p_i)^{2-1} = 2\,(q_i - \beta_0 - \beta_1 p_i)$$

Now, let's combine all of this together:

$$\frac{\partial y}{\partial \beta_0} = \frac{\partial u}{\partial \beta_0} * \frac{\partial y}{\partial u}$$

$$\frac{\partial y}{\partial \beta_0} = -1 * 2(q_i - \beta_0 - \beta_1 p_i) = -2(q_i - \beta_0 - \beta_1 p_i)$$

5) Take partial derivative of the error E (from step 3 above) in respect to β_0 in order to find the least sum of the squared errors (which means finding the value of β_0 for which the function equals to 0). Since we already solved it above, we only need to add a Σ symbol and replace y with E:

$$\frac{\partial E}{\partial \beta_0} = -2 \sum_i (q_i - \beta_0 - \beta_1 p_i) = 0$$

Simplify the above, and solve for β_0:

- Divide both sides by -2:

$$\frac{\partial E}{\partial \beta_0} = \sum_i (q_i - \beta_0 - \beta_1 p_i) = 0$$

- Break up the sum by all its parts by removing parentheses (remember that β_0 repeats n times; why? Because is inside Σ):

$$\frac{\partial E}{\partial \beta_0} = \sum q_i - \beta_0 n - \beta_1 \sum p_i = 0$$

/ why β_1 is not multiplied by n? Look at this: let's say $\beta_1 = 2$, $p_1 = 1$, $p_2 = 2$, $p_3 = 3$, then $\beta_1 * p_1 + \beta_1 * p_2 + \beta_1 * p_3 = 2*1 + 2*2 + 2*3 = 2+4+6 = 12$, which can be also written as $\beta_1 * \Sigma p_i = 2 * (1+2+3) = 12$; if however we multiplied β_1 by n (so by 3 p's), then we would get 2 * 3 = 6, which when multiplied by the sum of p's: (1+2+3) = 6 * 6 = 36. Conclusion: use either Σ or n (not both!) /

- Get rid of n by multiplying each part by 1/n:

$$\frac{\partial E}{\partial \beta_0} = \frac{\sum q_i}{n} - \frac{\beta_0 n}{n} - \beta_1 \frac{\sum p_i}{n} = 0$$

- Notice that the sum of the components when divided by n, equals their average values (the formula for average: $\Sigma p_i / n$, and it's represented using a dash at the top):

$$\frac{\partial E}{\partial \beta_0} = \bar{q} - \beta_0 - \beta_1 \bar{p} = 0$$

- Solve for β_0 by adding β_0 to both sides:

$$\bar{q} - \beta_1 \bar{p} = \beta_0$$

$$\beta_0 = \bar{q} - \beta_1 \bar{p}$$

6) Take partial derivative in respect to β_1:

Like in step 4 above, let's apply the chain rule:

$$\frac{\partial y}{\partial \beta_1} = \frac{\partial u}{\partial \beta_1} * \frac{\partial y}{\partial u}$$

Where:

$y = (q_i - \beta_0 - \beta_1 p_i)^2 = u^2$

$u = q_i - \beta_0 - \beta_1 p_i$

$$\frac{\partial u}{\partial \beta_1} = 0 * q_i * \beta_1^{0-1} - 0 * \beta_0 * \beta_1^{0-1} - 1 * \beta_1^{1-1} * p_i$$

$$\frac{\partial u}{\partial \beta_1} = -1 p_i$$

$$\frac{\partial y}{\partial u} = 2 (q_i - \beta_0 - \beta_1 p_i)^{2-1} = 2 (q_i - \beta_0 - \beta_1 p_i)$$

$$\frac{\partial u}{\partial \beta_1} * \frac{\partial y}{\partial u} = -1 p_i * 2 (q_i - \beta_0 - \beta_1 p_i) = -2 (q_i - \beta_0 - \beta_1 p_i) p_i$$

7) Take partial derivative of the error E in respect to β_1:

$$\frac{\partial E}{\partial \beta_1} = -2 \sum_i (q_i - \beta_0 - \beta_1 p_i) * (p_i) = 0$$

Simplify the above, and solve for β_1:

- Divide both sides by -2:

$$\frac{\partial E}{\partial \beta_1} = \sum_i (q_i - \beta_0 - \beta_1 p_i) * (p_i) = 0$$

- Break up the sum by all its parts multiplied by p_i (like we did in step 5):

$$\frac{\partial E}{\partial \beta_1} = \sum q_i p_i - \beta_0 \sum p_i - \beta_1 \sum p_i^2 = 0$$

- Replace Σp_i with the average of p_i * n:

$$\frac{\partial E}{\partial \beta_1} = \sum q_i p_i - \beta_0 \bar{p} n - \beta_1 \sum p_i^2 = 0$$

- Replace β_0 with the result from point 5 ($\beta_0 = \bar{q} - \beta_1 \bar{p}$):

$$\frac{\partial E}{\partial \beta_1} = \sum q_i p_i - (\bar{q} - \beta_1 \bar{p}) n\bar{p} - \beta_1 \sum p_i^2 = 0$$

- Multiply the parentheses by $n\bar{p}$:

$$\frac{\partial E}{\partial \beta_1} = \sum q_i p_i - (\bar{q} n\bar{p} - \beta_1 \bar{p} n\bar{p}) - \beta_1 \sum p_i^2 = 0$$

- Get rid of parentheses:

$$\frac{\partial E}{\partial \beta_1} = \sum q_i p_i - \bar{q} n\bar{p} + \beta_1 \bar{p} n\bar{p} - \beta_1 \sum p_i^2 = 0$$

- Simplify:

$$\frac{\partial E}{\partial \beta_1} = \sum q_i p_i - \bar{q} n\bar{p} + \beta_1 \left(n\bar{p}^2 - \sum p_i^2 \right) = 0$$

- Subtract the part with β_1 from both sides:

$$\frac{\partial E}{\partial \beta_1} = \sum q_i p_i - \bar{q} n\bar{p} = -\beta_1 \left(n\bar{p}^2 - \sum p_i^2 \right)$$

- Remove - from β_1:

$$\frac{\partial E}{\partial \beta_1} = \sum q_i p_i - \bar{q} n\bar{p} = \beta_1 \left(\sum p_i^2 - n\bar{p}^2 \right)$$

- Solve for β_1 (divide both sides by $\left(\sum p_i^2 - n\bar{p}^2 \right)$):

$$\beta_1 = \frac{\sum q_i p_i - \bar{q} n \bar{p}}{\sum p_i^2 - n \bar{p}^2}$$

The next step was to write a computational model using the Python programming language. However, thanks to one of the scientific libraries called sklearn, we did not have to translate the above mathematical equations into Python, but instead we imported the ready LinearRegression() function from this library, which we named ‘regressor’. We then used a variety of conditions as our independent variable x, which data collected by Snoopy (variable y) depended on. And finally, we used this linear regression function with the fit(x,y) method in order to find the function that was the closest to all the data points, as well as regressor.coef_ to get the values of this function’s coefficients that represented the effect of each condition (variable x) on our data points (variable y). Our model is shown below:

Import libraries and read data from file:

In [120]:
```
import pandas as pd

from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

dataset = pd.read_csv("https://raw.githubusercontent.com/btete/Snoopy_Data/main/rawSnoopyData.csv")
# dataset.drop(dataset.index[dataset['Restroom'] == 1], inplace=True)
# dataset.drop(dataset.index[dataset['Incense'] == 1], inplace=True)
# dataset.drop(dataset.index[dataset['Chemicals'] == 1], inplace=True)
dataset
```

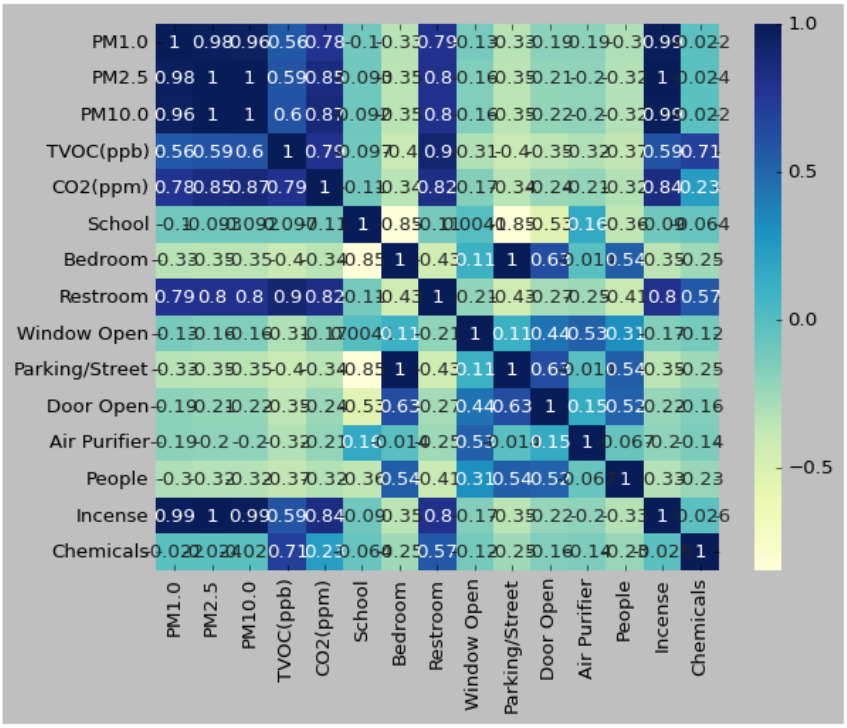
Out[120]:

	MM:DD:YYYY	HH:MM:SS	PM1.0	PM2.5	PM10.0	TVOC(ppb)	CO2(ppm)	School	Bedroom	Restroom	Window Open	Parking/Street	Door Open	Air Purifier	People
0	9/12/2021	3:36:00 PM	4	7	10	0	400	0	1	0	1	1	1	1	1
1	9/12/2021	3:37:00 PM	4	5	8	0	400	0	1	0	1	1	1	1	1
2	9/12/2021	3:38:00 PM	4	7	8	0	400	0	1	0	1	1	1	1	1
3	9/12/2021	3:39:00 PM	4	5	5	0	400	0	1	0	1	1	1	1	1
4	9/12/2021	3:40:00 PM	4	7	7	0	400	0	1	0	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2741	1/10/2022	7:51:00 PM	2	3	4	3695	743	0	0	1	0	0	0	0	0
2742	1/10/2022	7:52:00 PM	3	5	8	3580	734	0	0	1	0	0	0	0	0
2743	1/10/2022	7:53:00 PM	3	3	9	3640	733	0	0	1	0	0	0	0	0
2744	1/10/2022	7:54:00 PM	3	5	14	3564	735	0	0	1	0	0	0	0	0
2745	1/10/2022	7:55:00 PM	5	6	12	3364	689	0	0	1	0	0	0	0	0

2746 rows × 17 columns

Looking for relationships in a correlation heatmap:

```
In [30]: sns.heatmap(dataset.corr(), cmap="YlGnBu", annot = True)
         plt.show()
```



Thorough analysis of the above graphical representation of the correlation of data and all their conditions indicates that all the factors (or conditions) affected the air quality, incl. burning incense, sprayed chemicals, close location of parking lot and/or street or its lack, presence of people, open/closed window/door, air purifier on/off, but also the actual location of data collection, such as school, bedroom, and restroom. While data gathered at school and bedrooms differed significantly due to being away from parking lot and street, and having better ventilation system and air purifiers at school, as opposed to our bedrooms, the only reason why restroom seems to have such a significant effect on air, is the fact that it is where incense was burnt and chemicals were sprayed. This fact lets us see that using restroom as an independent variable would not be correct, because it's not the location itself that affected the air quality, but the actions performed there. That is why we decided to uncomment the first line with the

dataset.drop() function, and this way exclude the restroom from analyzing the level of pollen. Similarly with people, whose presence seems to have a positive effect on air quality at school. It's because while taking data at school, we had to have air purifiers on and windows open all the time (due to district-wide guidelines caused by the pandemic), which means that our data did not really inform how the presence of people at school affected air quality also when the windows were closed and air purifiers were off during school hours. This made our data lack some additional conditions and circumstances, but due to some limitations, which we couldn't control, our best option was to simply exclude some of these conditions from further analysis.

Select independent variable (features) and dependent variable (outcome):

```
[ ]  X = dataset[['Window Open', 'Door Open', 'Air Purifier']] #'People', 'School', 'Bedroom'
     y = dataset['PM10.0']
```

As shown on the above screenshot of our code, the independent variables, which we decided to focus on were: Window Open, Door Open and Air Purifier, and as the dependent variable, we used the level of the largest pollen (with a size of 10.0 microns).

Split datasets for validating model:

```
In [123]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
          X_train

Out[123]:
```

As shown above, 80% of data was used for training our model, and 20% was used for testing its accuracy.

Train linear regression model with training data:

```
In [127]: regressor = LinearRegression()
          regressor.fit(X_train, y_train)

Out[127]: LinearRegression()
```

This is where we used the LinearRegression() function imported previously from the sklearn library. We first saved it in the regressor variable, and then used it with the fit(x,y) method in order to train our model, so it could find the best fit linear function for 80% of our data.

Review model intercept and coefficients:

```
[ ] regressor.intercept_

    1.2287782035463906
```

```
[ ] coeff_df = pd.DataFrame(regressor.coef_, X.columns, columns=['Coefficient'])
    coeff_df
```

	Coefficient
Window Open	7.581324
Door Open	2.152548
Air Purifier	-2.994513

The above coefficients indicate that opening windows and doors increased the level of pollen, whereas turning air purifier on decreased it.

Test model predictions with testing data:

```
[ ] # PM10 = 1.228 + (WinOpen * 7.5) + (DoorOpen * 2.15) + (AirPurifier * -2.99)

    y_pred = regressor.predict(X_test)
```

```
[ ] df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
    df
```

	Actual	Predicted
182	7	7.968137
224	3	7.968137
1946	0	1.228778
289	2	7.968137
1231	6	3.381326
...	...	...
716	1	7.968137
1119	0	-1.765735
2299	0	-1.765735
125	9	7.968137
1150	0	-1.765735

Review model performance:

```
[ ] import numpy as np

    # See https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

    print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
    print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
    print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 3.72707815803822
Mean Squared Error: 31.004970842460786
Root Mean Squared Error: 5.5682107397673795
```

As explained before, the lower the Root Mean Squared Error (RMSE), the better our model fits actual data. Since its value is only about 5.6, we can assume that it correctly predicts the amount of pollen depending on whether the window and/or door is open or not and/or the air purifier is on or off.

Repeat process above for TVOC, instead of PM10:

```
X = dataset[['Window Open', 'Door Open', 'Air Purifier', 'Incense', 'Chemicals']]
y = dataset['TVOC(ppb)']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train, y_train)
regressor.intercept_
```

```
471.9253872977935
```

```
coeff_df = pd.DataFrame(regressor.coef_, X.columns, columns=['Coefficient'])
coeff_df
```

	Coefficient
Window Open	-157.601225
Door Open	-169.084094
Air Purifier	-126.633370
Incense	3283.929876
Chemicals	5561.337771

Since both incense and chemicals had a significant impact on the level of total volatile organic compounds in the air, we decided to include these factors in the analysis of TVOC. As shown on the above screenshot of our code, simply opening the windows and doors as well as turning the air purifier on helped decrease the level of chemicals in the air, whereas burning incense and spraying chemicals significantly increased it.

```
y_pred = regressor.predict(X_test)
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

	Actual	Predicted
2401	222	187.690792
1590	278	302.841294
2381	238	187.690792
2683	2641	3755.855263
1147	322	345.292017
...	...	...
1043	186	176.207924
878	12	145.240068
1762	397	471.925387
303	214	18.606698
132	0	18.606698

550 rows × 2 columns

```
[74] print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
     print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
     print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

     Mean Absolute Error: 156.42583394082982
     Mean Squared Error: 174371.94026412637
     Root Mean Squared Error: 417.5786635642755
```

As opposed to RMSE obtained for pollen, this one was much bigger, which means that the model did not predict the amount of TVOC correctly. We were certain that including the outliers, such as incense and chemicals, contributed to this poor performance of our model a lot. Also, incomplete data about the impact of people, which - due to the school district's directives - always required windows to be opened and air purifier to be turned on (which caused the lack of data informing about the effect of people on the air quality when the windows were closed and air purifier was off), might cause the discrepancy between our model and the actual data. After removing these factors, the RMSE, as shown below, improved a lot:

```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

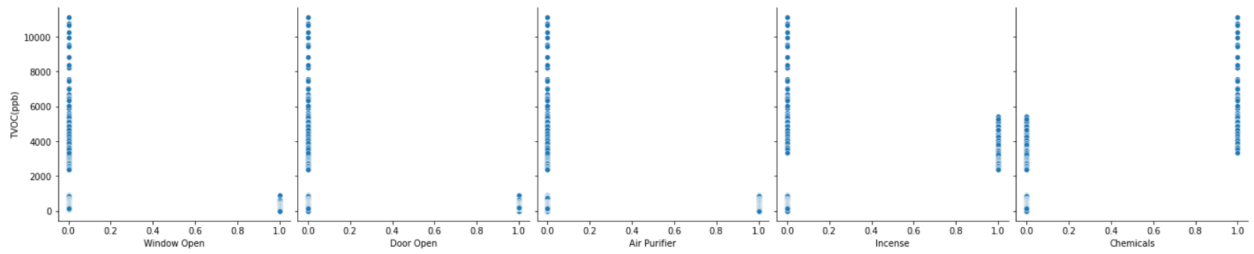
```
Mean Absolute Error: 54.14432043841056
Mean Squared Error: 5086.660344896204
Root Mean Squared Error: 71.32082686632428
```

As shown on the screenshot above, the RMSE significantly improved, but it was still much higher than the one for pollen. We believe that the main reason for it, is the fact that TVOC readings were much larger than PM 10.0 data, and that this higher RMSE might be caused by the scale differences between these two.

Reviewing distribution of data graphically:

```
import seaborn as sns
#Using pairplot we'll visualize the data for correlation
sns.pairplot(dataset, x_vars=['Window Open', 'Door Open', 'Air Purifier', 'Incense', 'Chemicals'],
             y_vars='TVOC(ppb)', height=4, aspect=1, kind='scatter')
```

```
<seaborn.axisgrid.PairGrid at 0x7f3914de9110>
```



As shown on the charts above, closed windows and doors, turned air purifiers off, and sprayed chemicals caused significant accumulation of TVOC in the air. What surprised us here was the higher level of TVOC when the incense was not burnt. This could be caused by the fact that when incense was not burnt in the restroom (that was without a window and with the door closed) we then sprayed a variety of chemicals there. Thus, when the binary value of incense was 0, the binary value of chemicals was 1 (which had a stronger effect on the level of TVOC than incense). We can also clearly see that when chemicals were not sprayed, the amount of TVOC was exactly the same as the amount of TVOC accumulated while burning incense. This finding

lets us clearly see that exploring the presence of the factors without exploring their lack certainly contributed to such a wrong outcome. This could be fixed by measuring the quality of air in the restroom when neither incense nor chemicals were used. Another approach could be separating incense from chemicals by placing them in different locations (e.g. Restroom1 and Restroom2) in our dataset, which would give us better control over them. It would also be more accurate if the same amount of data was gathered for each condition explored. The measurement performed in the restroom took only about two hours, while the measurement of other conditions and locations lasted much longer, spanning even the entire night and day.

Exploring CO2:

In [64]:
```
import pandas as pd
import numpy as np

from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

dataset = pd.read_csv("https://raw.githubusercontent.com/btete/Snoopy_Data/main/Snoopy_Data.csv")
dataset
```

Out[64]:

	School	Bedroom	Restroom	Window Open	Trees	Cars	Door Open	Air Purifier	People	Incense	Chemicals	Avg Pollen/Dust 1.0	Avg Pollen/Dust 2.5	Avg Pollen/Dust 10	Avg CO2	Avg TVOC
0	1	0	0	1	1	0	0	1	1	0	0	8	10	13	411	2142
1	1	0	0	1	1	0	0	1	0	0	0	7	9	12	400	153
2	1	0	0	0	1	0	0	0	0	0	0	5	6	6	400	623
3	0	1	0	0	1	1	0	0	1	0	0	0	0	1	1407	1029
4	0	1	0	1	1	1	1	0	1	0	0	11	16	19	400	22
5	0	1	0	1	1	1	1	1	1	0	0	3	4	5	400	0
6	0	0	1	0	1	1	0	0	0	1	0	295	2139	3273	2868	3782
7	0	0	1	0	1	1	0	0	0	0	1	4	10	26	1427	6036

In [65]:
```
X = dataset[['School', 'Window Open', 'Cars', 'Door Open',]]
y = dataset['Avg CO2']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train, y_train)
regressor.intercept_
```

Out[65]: 1215.8000000000002

In [66]:
```
coeff_df = pd.DataFrame(regressor.coef_, X.columns, columns=['Coefficient'])
coeff_df
```

Out[66]:

	Coefficient
School	-201.2
Window Open	-609.1
Cars	201.2
Door Open	-407.9

In [67]:
```
y_pred = regressor.predict(X_test)
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[67]:

	Actual	Predicted
6	2868	1417.0
2	400	1014.6

In [40]:
```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 1032.8000000000002
Mean Squared Error: 1241567.08
Root Mean Squared Error: 1114.2562900876978
```

Based on the values of the above coefficients, the level of CO2 in school was always at its minimum, as well as opening the window and door always helped with decreasing its amount. However, since RMSE was so high, which made our model even more ineffective than our initial TVOC model, we also decided to exclude the outliers (incense and chemicals) together with incomplete data about people, and this is what we ended up with after modeling linear regression again with our updated dataset:

```
[39] X = dataset[['Window Open', 'Door Open', 'Air Purifier']] #'People', 'School', 'Bedroom'
     y = dataset['CO2(ppm)']
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
     regressor = LinearRegression()
     regressor.fit(X_train, y_train)
     regressor.intercept_

     400.0
```

```
     coeff_df = pd.DataFrame(regressor.coef_, X.columns, columns=['Coefficient'])
     coeff_df
```

	Coefficient
Window Open	0.0
Door Open	0.0
Air Purifier	0.0

```
[34] y_pred = regressor.predict(X_test)
     df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
     df
```

```
[38] print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
     print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
     print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

     Mean Absolute Error: 0.0
     Mean Squared Error: 0.0
     Root Mean Squared Error: 0.0
```

Both the coefficients and RMSE are equal to zero meaning that our model accurately represents the actual data. Without the outliers and incomplete data about the effect of people on the quality of air, the level of CO2 was always at 400 particles per million parts of air, which is the exact value of the y-intercept as shown above. Thus, our modified CO2 model can be represented by the following equation:

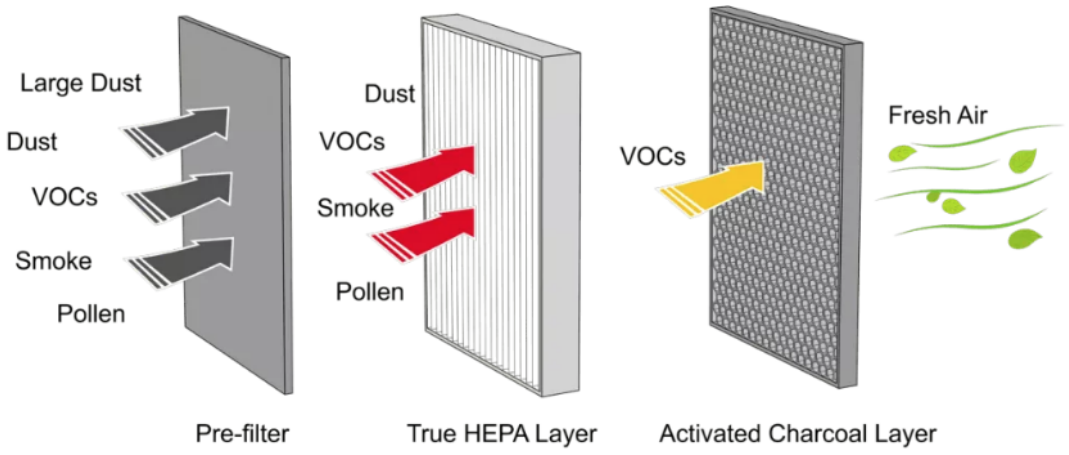
y (CO2 at 400 ppm) = b (y-intercept) + m_1x_1 + m_2*x_2 + m_3x_3

CO2 at 400 ppm = 400.0 + 0*Window Open + 0*Door Open + 0*Air Purifier

CO2 at 400 ppm = 400.0

Conclusion

As we've learned, in order to improve the air quality indoors, in addition to helping VOC and CO2 escape from the room by simply opening the window (if it's OK to bring pollen inside) or at least the door, it would be also very beneficial to replace the carpet with tiles that can be washed frequently, use ecological paints, invest in good ventilation system, and use air purifiers with HEPA filters, like this one:



As officially defined by the U.S. Department of Energy, high efficiency particulate air (HEPA) filters can theoretically remove at least 99.97% of dust, pollen, mold, bacteria, and any airborne particles with a size of 0.3 microns (µm). The diameter specification of 0.3 microns responds to the most penetrating particle size (MPPS) (United States Environmental Protection Agency, 2021).

We also learned that collecting data is a very important process in scientific research. If there is no way that a particular condition (for example the presence of people) can be measured in a variety of circumstances (e.g. open/closed window/door, air purifier on/off), then it's better to exclude such incomplete data from the model. Otherwise, the coefficient representing such a condition will most likely be incorrect, and the entire model will not be an accurate

representation of the reality. Thus, while focusing on a particular condition, it's important to collect an equal amount of data in a variety of circumstances, which will show a more accurate and realistic effect of such a condition on the measured outcomes. In other words, while collecting data when the specific conditions are true, we also should collect data when one of them is true and the other one is false, and vice versa, as well as when all of them are false.

And finally, we strongly believe that collecting the same amount of data for different circumstances should also be taken into account in order to make them more comparable, which would lead to a more accurate model. If gathering the same amount of data for each condition would not be possible, another solution would be a separate model for each explored location.

Collaboration

Roles / Responsibilities

We've shared the responsibilities and roles while working on this project. It all depended on our interests and certain natural skills. Some of us were better in public speaking and oral presentations than others, some others were more interested and skilled in computer programming, whereas others were able to explain the mathematical concepts, which we all learned about, and finally there were also those, who mostly enjoyed working with hardware.

Contributions

We all were meeting regularly after school with our mentors and our teacher-sponsor. We listened to the lectures and followed our mentor while working with hardware. All of us gathered our own data, which we then plotted individually using our collaboratively developed Python

program. We all also explored and studied all the mathematical concepts used in linear regression, including line function, partial derivatives, chain rule, etc., as well as we learned how to write mathematical equations using LaTeX to explain these concepts. We then were regularly meeting with our other mentor, who taught us how to apply such linear regression in a computational model in order to perform statistical analysis and to predict outcomes from actual data. And finally, we were all working together on preparing this report as well as the final presentation.

Works Cited

Advanced Solutions Nederland B.V. “How TVOC affects indoor air quality: effects on wellbeing and health.” *Advanced Solutions Nederland B.V.*, 3 December 2020, https://www.advsolned.com/how-tvoc-affects-indoor-air-quality-effects-on-wellbeing-and-health/. Accessed 9 November 2021.

Ang, Carmen, et al. “Zooming In: Visualizing the Relative Size of Particles.” *Visual Capitalist*, 10 October 2020, https://www.visualcapitalist.com/visualizing-relative-size-of-particles/. Accessed 8 November 2021.

Brownlee, Jason. “Linear Regression for Machine Learning”. *Machine Learning Mastery*, 15 August, 2020, https://machinelearningmastery.com/linear-regression-for-machine-learning. Accessed 26 March, 2022.

Fields, Lisa, and Carol DerSarkissian. “Pollen Allergies Overview.” *WebMD*, 19 April 2021, https://www.webmd.com/allergies/pollen_allergies_overview. Accessed 14 November 2021.

Lindsey, Rebecca. “Climate Change: Atmospheric Carbon Dioxide.” *Understanding Climate*, climate.gov, 14 August 2020, https://www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide. Accessed 10 November 2021.

Smith, Dianna. “Everything You Need To Know About Carbon Dioxide (CO2).” *kaiterra*, 25 September 2019, https://learn.kaiterra.com/en/air-academy/carbon-dioxide-co2. Accessed 10 November 2019.

United States Environmental Protection Agency. “What is a HEPA filter?” *EPA*, 3 March 2021, https://www.epa.gov/indoor-air-quality-iaq/what-hepa-filter-1. Accessed 8 November 2021.

The World Green Building Council. “Indoor Environmental Quality: A How-To Guide.” *WorldGBC*, 2020, https://www.worldgbc.org/sites/default/files/bp-resource/BPFP-IEQ-Guidance-Note.pdf. Accessed 9 November 2021.

Zach. “How to Interpret Root Mean Square Error (RMSE)”. *Statology*, 10 May 2021, https://www.statology.org/how-to-interpret-rmse. Accessed 26 March 2022.