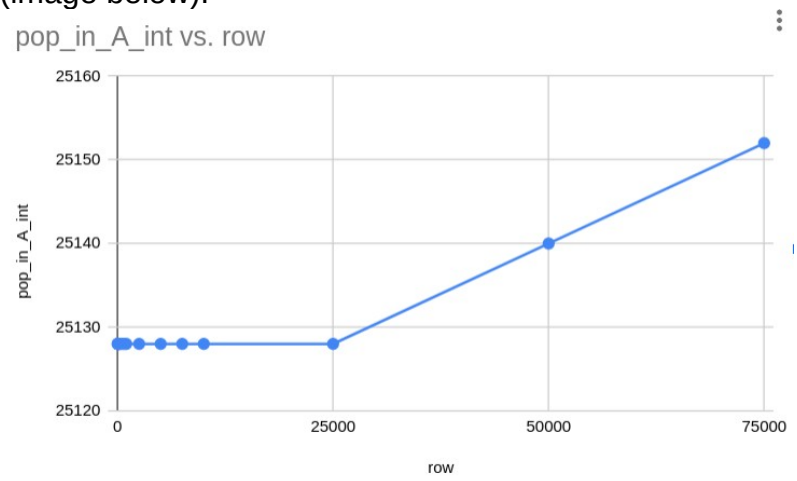
Vpython trace analysis on Weather data

1. Dataset

- Source: https://www.kaggle.com/datasets/budincsevity/szeged-weather
- Only keep temperature and humidity variables
- Round temperature to 2 decimal digits. We tried multiply temperature by 100 and convert to integers but their traces are not very informative with such given small dataset of ~96000 samples (image below).

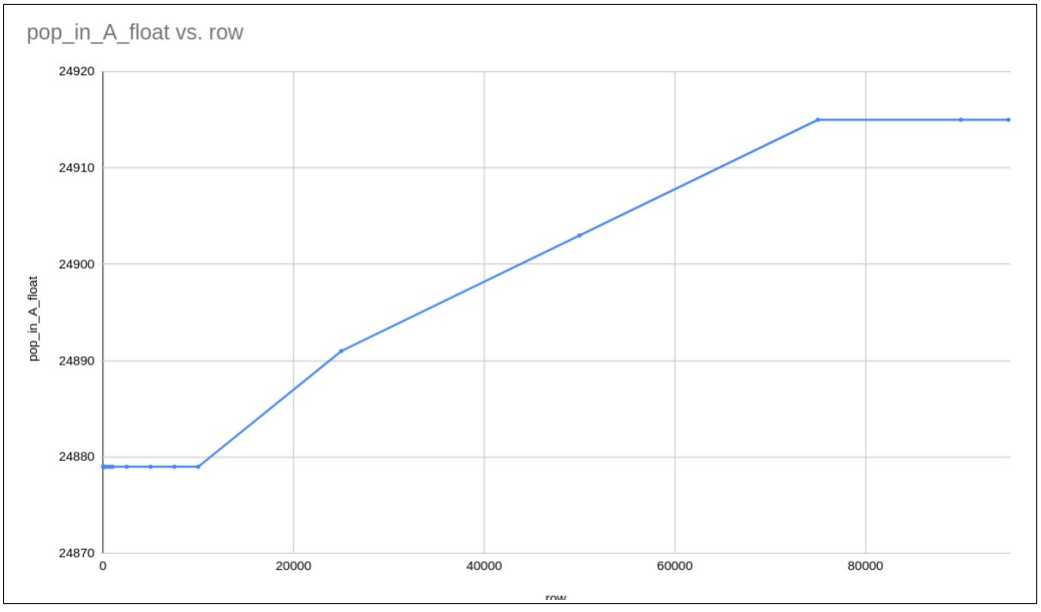


- **The regression model** aims to predict **humidity** from **temperature**
- To sample the dataset to different numbers of rows, we use this code:

```
df_crop = df.sample(
    n=row_num, axis=0, replace=True, ignore_index=True, random_state=10)
```

Note that we're setting replace=True to repeat some datapoints to achieve 100 000 rows from the original ~96 000 rows dataset. Why 100 000? Because it may show more insights as we look at/ try to predict the trace.

As can be seen from the graph below (pop vs. row), the figures show a good linear until the 90 000 rows milestone. So perhaps 100 000 will make the pop go up to the projected linear line again.

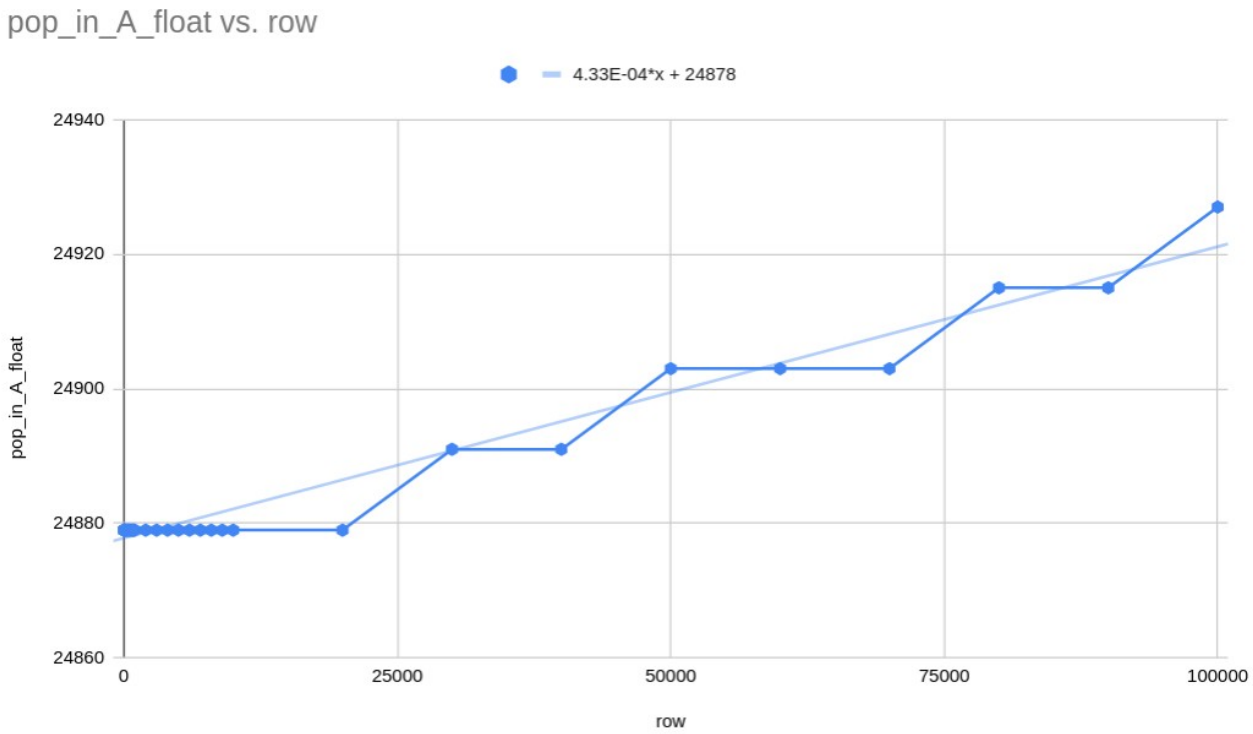


- Realising the limitation of the given sizes of rows list and **more hidden details may lie in finer row size like 90 000**, we decided to sample to finer sizes of rows as describe in the code below:

```
my_row_list = [i*(10**power) for power in range(1, 5)
               for i in range(1, 10)] + [100_000]
```

2. Trace analysis result

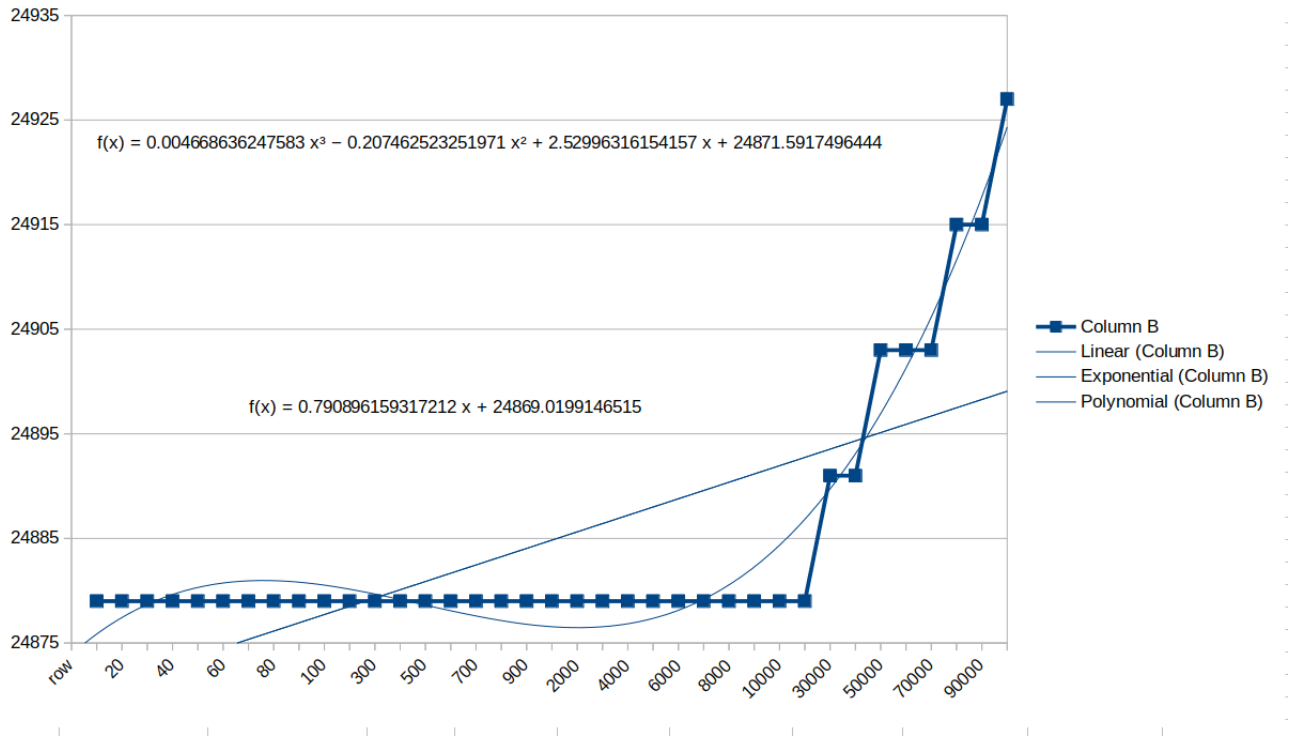
As we expected, the hidden details mentioned are some un-increasing pop number in some row sizes. Regarding this, we conjecture that there were not enough data to reach certain threshold where the vPython requires more pop operations.



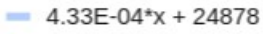
Overall, the trace shows a good linear trend and can be predicted with a regression model.

2.1 Some possible equation

From Libre Numbers



From Google Sheets



The sklearn regression model agrees with what we got from Google Sheets

```
regression = linear_model.LinearRegression()
regression.fit(x, y)

print(regression.coef_, regression.intercept_)
# output: [0.00043282] 24877.819753842254
```

(Well, the equations from Libre Numbers are not very trust worthy.)

3. Summary

So, what's next? Getting more metrics (RMSE, etc.) for the current linear regression model predicting trace?

Please let me know :)