

REPORT

Reflection:

1. What was the easiest and hardest part of this assignment?

The easiest part of the assignment was actually doing the calculations. The hardest part of the assignment was 1) parsing through the data to get input that was actually usable and 2) understanding where to put the “try/except” blocks and how that differs from a “raise” block.

The most time-consuming part of the lab was definitely just parsing through the data. I originally started the lab by just trying to return any line that contains “PORTLAND”, but that didn’t work and I’m still not sure I understand why. Instead, it would only return data that contained the letter “P”, which obviously wasn’t what I was looking for. I fixed this by creating a list of lists first and then having the function return lines where `index[1]` started with the string “PORTLAND” and somehow this fixed the issue.

Regarding error handling, I now understand that “try/except” blocks can be used to continue after an error occurs. Whereas the “raise” block is used to draw attention to a specific error and halt the program from moving forward until that error is resolved.

2. What did you learn?

I had an opportunity to practice using exception and error handling. This is obviously critical for writing programs down the line and is a great way to let users know when something is wrong without writing a lot of code.

This lab also provided the opportunity to continue to work with reading in and writing out data from another file. This can be challenging but is an extremely important skill that I’d like to get more practice with. For me, reading in outside data and converting it to the right data type is still difficult, so I appreciate the insight this lab provided. I can now go back to this lab if I ever need some extra guidance on how to do this in the future.

Output:



```
Lab08 > output_data.csv
1  Portland Average Rainfall:
2  2010 : 0.160
3  2011 : 0.152
4  2012 : 0.158
5  2013 : 0.128
6  2014 : 0.154
7  2015 : 0.133
8  2016 : 0.121
9  2017 : 0.121
10 2018 : 0.150
11 2019 : 0.151
12 2020 : 0.127
13 2021 : 0.123
```

Extensions:

1. My first extension (extension01.py) adds a parameter to the read_in function that asks the user for input to select the city they would like to collect data on. I used the .upper() function to make sure the input matched what was stored in the raw data list. I also created another “try/except” block in the get_avgs since some cities are missing data during certain years, which would make the denominator a zero.
2. My second extension (extension02.py) looks at the snowfall data (line_list[10]) instead of precipitation and prints those results to the output file.

Grading Statement:

1. What grade do you think you deserve and why?

I would give myself a 28 out of 30 because, although I struggled with this lab, I was able to complete all the rubric requirements as well as add two extensions (explained above). My code is clean, well-commented, and runs as intended. I also was able to print the number of times the read_in() function hit the empty data error (in the terminal) and I figured out how to quickly append each line of data to correct list line with a single line of code.