hw4-BahramsariBehina

October 16, 2024

1 CMPT 140 Homework Assignment # 4 - Survey of Student Prior Programming Experience

[Behina Bahramsari]

Student # [660360]

Date: [October 16, 2024]

[2]:
```
from IPython.display import Markdown as md
import matplotlib.pyplot as plt

# TODO: read in survey data
# 1. open file
# 2. use for loop to load data into list of lists
#    i.e. each row would correspond to one item in the main (top-level) list and
#    each column would correspond to an item in the inner (1-level down) list.
#    For example, a csv file corresponding to a table of 15 rows and 7 columns␣
 ↪would
#    result in a list of length 15.  Each item in this list (of length 15)␣
 ↪would be a list
#    of length 7.
#

### PLEASE ENTER YOUR CODES HERE ###

# Initialize an empty list to store the data that has been read
db = []

# Path to the CSV file
file = "./student prior experience.csv"

# Open the file and read line by line, if we use f = open(file, "r") we should␣
 ↪manually close it
with open(file, "r") as f:
    # Read each line and split it by commas to form rows of data
    for line in f.readlines():
        # print(line)
```

```
        #Strip whitespace characters (like newline) and split by commas
        row = line.strip().split(',')
        db.append(row)

# print(db) # db is a list of list and include 65 rows which each row is a list␣
 ↪of 2 columns' values also first row is label (column names)

# TODO: count the number of students in the class and store this number in a
# variable name "num_students". The variable name has to be named EXACTLY.
# warning: some student id's appears on multiple rows!
num_students=0 # this is dummy value

### PLEASE ENTER YOUR CODES HERE ###

# Initialize an empty list to track unique student IDs
unique_ids = []

# Iterate through all rows except the first row since it is label (column␣
 ↪names) in the db and get the student ID which is located in the first column
for i in range(1,len(db)):
    row = db[i]
    student_id = row[0]
    if student_id not in unique_ids:
        unique_ids.append(student_id)

# print("Unique ids:", unique_ids)
num_students = len(unique_ids)
```

[3]:
```
# Report the number of students.
md("There are {} students in CMPT 140. ".format(num_students))
```

[3]: There are 47 students in CMPT 140.

[4]:
```
# TODO: find out what programming languages students have prior experience in.
#
# Please create a list and name this list "avail_languages".  The variable name␣
 ↪has to be
# named EXACTLY, otherwise, the barplot towards the end of this file will not␣
 ↪display correctly.
#
# Please put all observed programming languages in this list.
# BE CAREFUL - avail_languages should contain UNIQUE items e.g. this list␣
 ↪should NOT have
# more than one "Python" in it.
#
avail_languages = []
```

```
### PLEASE ENTER YOUR CODES HERE ###

# Iterate through all rows except the first row since it is label (column␣
 ↪names) in the db and get the languages which is located in the second column
for i in range(1,len(db)):
    row = db[i]
    student_language = row[1]
    if student_language not in avail_languages:
        avail_languages.append(student_language)

# print("Unique programming languages:", avail_languages)
```

[5]:
```
# TODO: find out how many students know the programming languages in␣
 ↪avail_languages.
#
# Please create a list, named "num_student_language".  The variable name has to␣
 ↪be
# named EXACTLY, otherwise, the barplot towards the end of this file will not␣
 ↪display
# correctly.  This list should contain the number of students with prior␣
 ↪experience
# in the programming languages in the list "avail_languages"
#
# For example, if 10 students have prior expeience in Java and 12 students have␣
 ↪prior
# experience in Python
# avail_languages = ["Java", "Python"]
# num_student_language = [10,12]
#
# Please note: you must NOT use Pandas or other Python libraries to do the␣
 ↪analysis.
# Also, you cannot manually count up the numbers of students and hard code the␣
 ↪numbers
# in this Jupyter notebook.  You must write Python codes to help you do the␣
 ↪counting
# programmatically.
#
num_student_language = []

### PLEASE ENTER YOUR CODES HERE ###

# Initialize an empty list to track all student langueges
student_languages = []

# Iterate through all rows except the first row since it is label (column␣
 ↪names) in the db and get the languages which is located in the second column
```

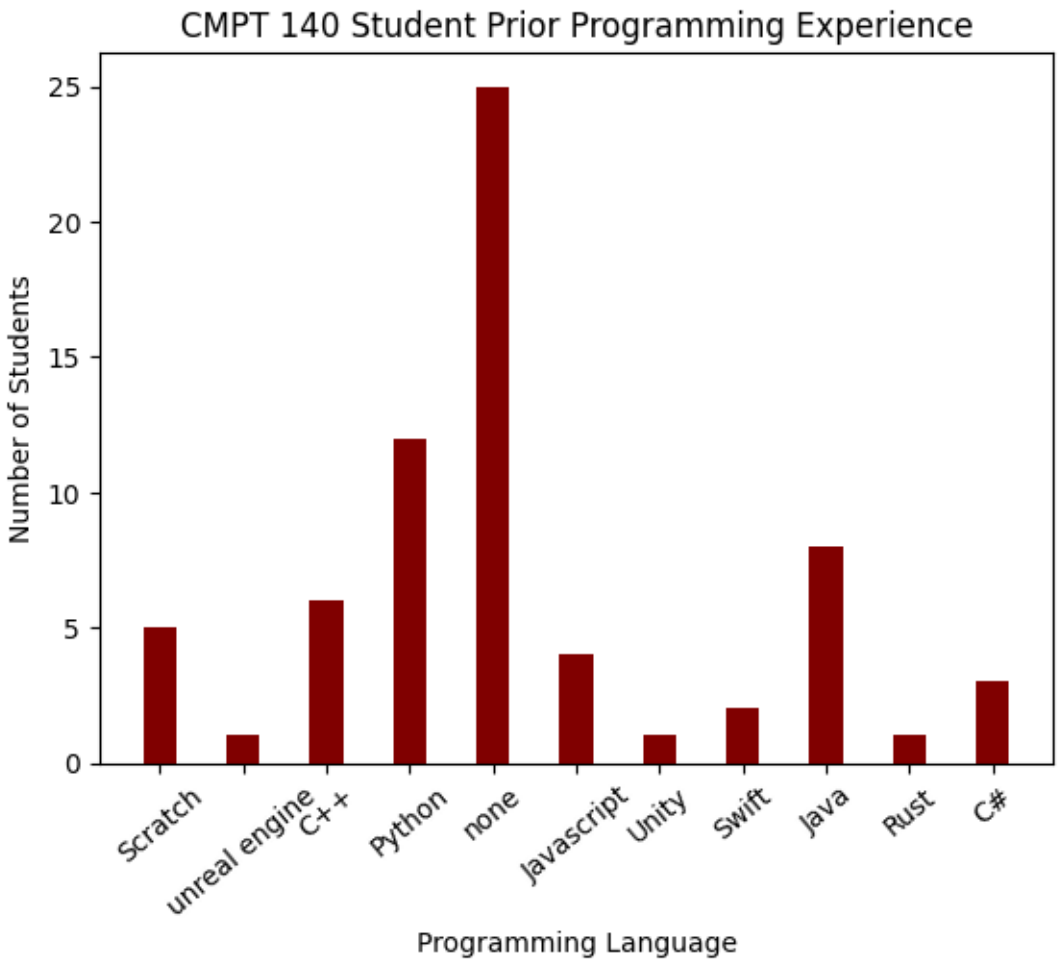
```
for i in range(1,len(db)):
    row = db[i]
    student_languages.append(row[1])

# print("all students' programming languages:", student_languages)
# print("Unique programming languages:", avail_languages)

for lang in avail_languages:
    count = 0
    # Iterate through each student's languages and check if the current␣
 ↪language is in their experience
    for student_lang in student_languages:
        if lang in student_lang:
            count += 1
    # Append the count to the num_student_language list
    num_student_language.append(count)

# print("Number of students for each language:", num_student_language)
```

[6]:
```
# display a barplot that shows the distribution of programming languages
# CMPT 140 student had before taking this class.
plt.bar(avail_languages, num_student_language, color ='maroon', width = 0.4)
plt.title("CMPT 140 Student Prior Programming Experience")
plt.ylabel("Number of Students")
plt.xlabel("Programming Language")
# the following line rotate x-axis label to declutter
# note the ";" at the end.  This is for suppressing the text output
plt.xticks(rotation=40);
```



1.1 Observation

[TODO: please write 1-2 brief sentences describing what you observed with the above barplot or data]

[It reveals that the most common prior programming languages among CMPT 140 students are "none" and "Python," with a significant number of students having no prior experience or knowing Python while languages like "unreal engine" and "Unity" and "Rust" have the fewest students with prior experience.]