lab1_solutions

September 22, 2021

Exercise: Create a tuple called `tup` with the following seven objects:

.

- The first element is an integer of your choice
- The second element is a float of your choice

- The third element is the sum of the first two elements
- The fourth element is the difference of the first two elements
- The fifth element is first element divided by the second element

Display the output of `tup`. What is the type of the variable `tup`? What happens if you try and chage an item in the tuple?

[2]:

```python
# your code here
a = 1
b = 2.0
tup = (a, b, a + b, a - b, a/a)
print(tup, type(tup))
print(type(a))
```

```
(1, 2.0, 3.0, -1.0, 1.0) <class 'tuple'>
<class 'int'>
```

Exercise: Build a list that contains every prime number between 1 and 100, in two different ways:

1. Using for loops and conditional if statements.
2. *(Stretch Goal)* Using a list comprehension. You should be able to do this in one line of code, and it may be helpful to look up the function `all` in the documentation.

[35]:

```python
# your code here
N = 100;

# using loops and if statements
primes = [];
for j in range(2, N):
    count = 0;
    for i in range(2,j):
        if j % i == 0:
            count = count + 1;
    if count == 0:
        primes.append(j)
print(primes)
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
79, 83, 89, 97]
```

[36]:

```
# your code here
# using list comprehension
primes_lc = [j for j in range(2, N) if all(j % i != 0 for i in range(2, j))]

print(primes)
print(primes_lc)
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
79, 83, 89, 97]
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
79, 83, 89, 97]
```

Exercise:In an exercise from Lab 0, you wrote code that generated a list of the prime numbers between 1 and 100. For the excercise below, it may help to revisit that code.

Write a function called `isprime` that takes in a positive integer N, and determines whether or not it is prime. Return `True` if it's prime and return `False` if it isn't. Then, using a list comprehension and `isprime`, create a list `myprimes` that contains all the prime numbers less than 100.

[41]:

```
# your code here
def isprime(N):
    count = 0;
    if not isinstance(N, int):
        return False
    if N <= 1:
        return False
    for i in range(2, N):
        if N % i == 0:
            count = count + 1;
    if count == 0:
        return(True)
    else:
        return(False)
print(isprime(3.0), isprime("rahul"), isprime(0), isprime(-1), isprime(1),␣
 ↪isprime(2), isprime(93), isprime(97))
myprimes = [j for j in range(1, 100) if isprime(j)]
print(myprimes)
```

```
False False False False False True False True
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
79, 83, 89, 97]
```

Exercise

Create a scatter plot showing the co-variation between two columns of your choice. Label the axes. See if you can do this *without* copying and pasting code from earlier in the lab. What can you conclude, if anything, from your scatter plot?

[5]:

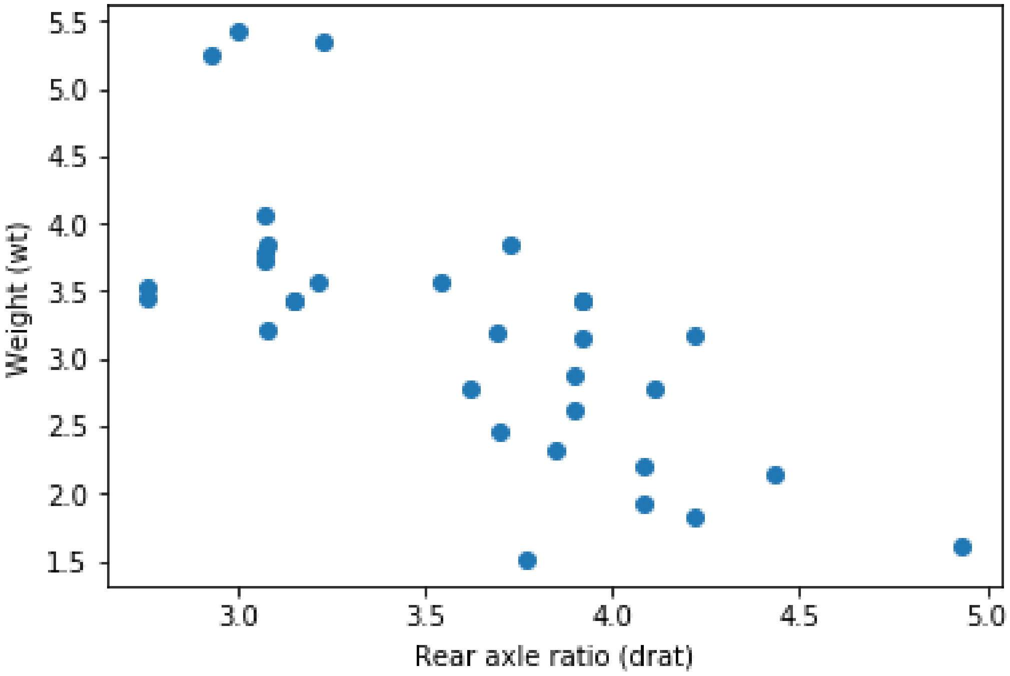
```
# your code here
# Read in the csv files
%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd
dfcars=pd.read_csv("../data/mtcars.csv")
type(dfcars)

# The columns to plot
col1 = 'drat'
col2 = 'wt'

# Plot
plt.plot(dfcars[col1], dfcars[col2], 'o')
plt.xlabel('Rear axle ratio (drat)')
plt.ylabel('Weight (wt)')
```

[5]:

```
Text(0, 0.5, 'Weight (wt)')
```



[ ]: