ECS10

2/4

Midterm curves

- Midterm1 Multiple Choice
  - 9-11 A
  - 7-8 B
  - 6-4 C
- Midterm1 Program
  - 25-34 A
  - 18-24 B
  - 10-17 C

Announcements

- Next assignment due Thurs 2/14.
- Checkpoint due Thurs 2/7.
- More helpful Web sites
  - PythonAnywhere – write programs online
  - CodingBat – additional exercises

Reading the whole file

```
while True:
    line = menuFile.readline()
    if line == "":
        break
    line = line.strip()
    print(line)
```

- When the file is over, line gets the empty string.
- Close file after reading with close() method.

Analyzing a line

```
line = line.strip()
print(line)
words = line.split()
print(words[0]) # first word
print(words[-1]) # last word
```

- We could look at a line one character at a time.
- Easier to break it into words or meaningful substrings.
- split() method breaks a line up into words.

Most caloric item on menu?

- See enclosed program.

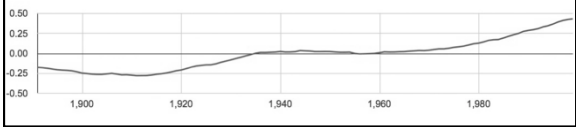
New assignment

- Global temperature data from NOAA. Download the text file.
- Temperature anomaly – deviation from average. So it is roughly between -1 and +1.
- Starts with 1880 and ends with 2011.

Smooth out to see trend

- Before

- After averaging 10 years in either direction



Moving average

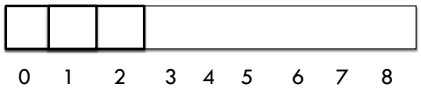
- Say we average 1 year before and after:

```
1927    -0.1045
1928    -0.1032
1929    -0.2198
```

- Average for 1928 would be:

  (-0.1045 + -0.1032 + -0.2198) / 3

Can't get moving average at ends



- Example: we have years 0-8 (total of 9)
- We can compute moving averages with one on either side only at positions 1-7 (total of 7)
- Your program will produce a different number of output lines depending on how many the user wants to average on either side.

Structure of program

- Two-loop program:
  - One loop to read input file, store it in a list
  - Second loop to compute averages, write output file
  - First loop due this week
- Output file in format that can be used by Excel, Google Documents, etc.: .csv
- Put output into Google Documents and use it to produce the graph.

Building a list – append() method

- The append() method sticks a new element onto the end of a list.

```
>>> shop=["cabbage","tea","yoghurt"]
>>> shop.append("bread")
>>> shop
["cabbage","tea","yoghurt","bread"]
```

- Notice you don't need an assignment statement.
- Lists are mutable. Strings aren't.

tuples

- A tuple is just like a list, is a sequence, but NOT mutable
- The in operator works, indexing works, the length function works, concatenation works
- The append method does not work.
- Written with () instead of []

```
[ 5, 6, 7 ]  # list
( 5, 6, 7 )  # tuple
```

Building a list of five zeros

```
i = 0
zeroList = []  # the empty list
while i < 5:
    zeroList.append(0)
    i = i+1
```

Special Python trick

```
zeroList = [0]*5
```

- Just the same as the previous program.

In moving average program

- Use the append method to build up a list of temperature anomalies, as you read the input file and extract them.
- At each line, get anomaly, convert it to integer, append it onto back of list.
- Hand in on Thursday.