Python for Machine Learning

Control & Boolean Statements

CE 5331 Machine Learning for Civil Engineers

Venki Uddameri, Ph.D. , P.E.

Recap and Goals

- Installed Python and Anaconda Environments
- Introduction to Python
  - Setting working directory
  - Adding comment lines
    - Docstrings
- Introduction to Pandas
  - Reading a csv
  - Extracting columns (attributes)
  - Extracting rows
  - Obtaining summary measures

Goal of this Module is to introduce
Look at Control Statements in
Python

Control Constructs

- Python is an interpreter, so the flow of the code is sequentially downwards
  - Sometimes we have to perform repetitive calculations
    - Summing up the numbers in a list to calculate mean
  - We have to make decisions
    - Perform one set of calculations when condition A is satisfied and another set of calculations when condition B is satisfied
- Python provides a set of control statements
  - if, if-else, if-elif-else
  - for loop
  - while loop

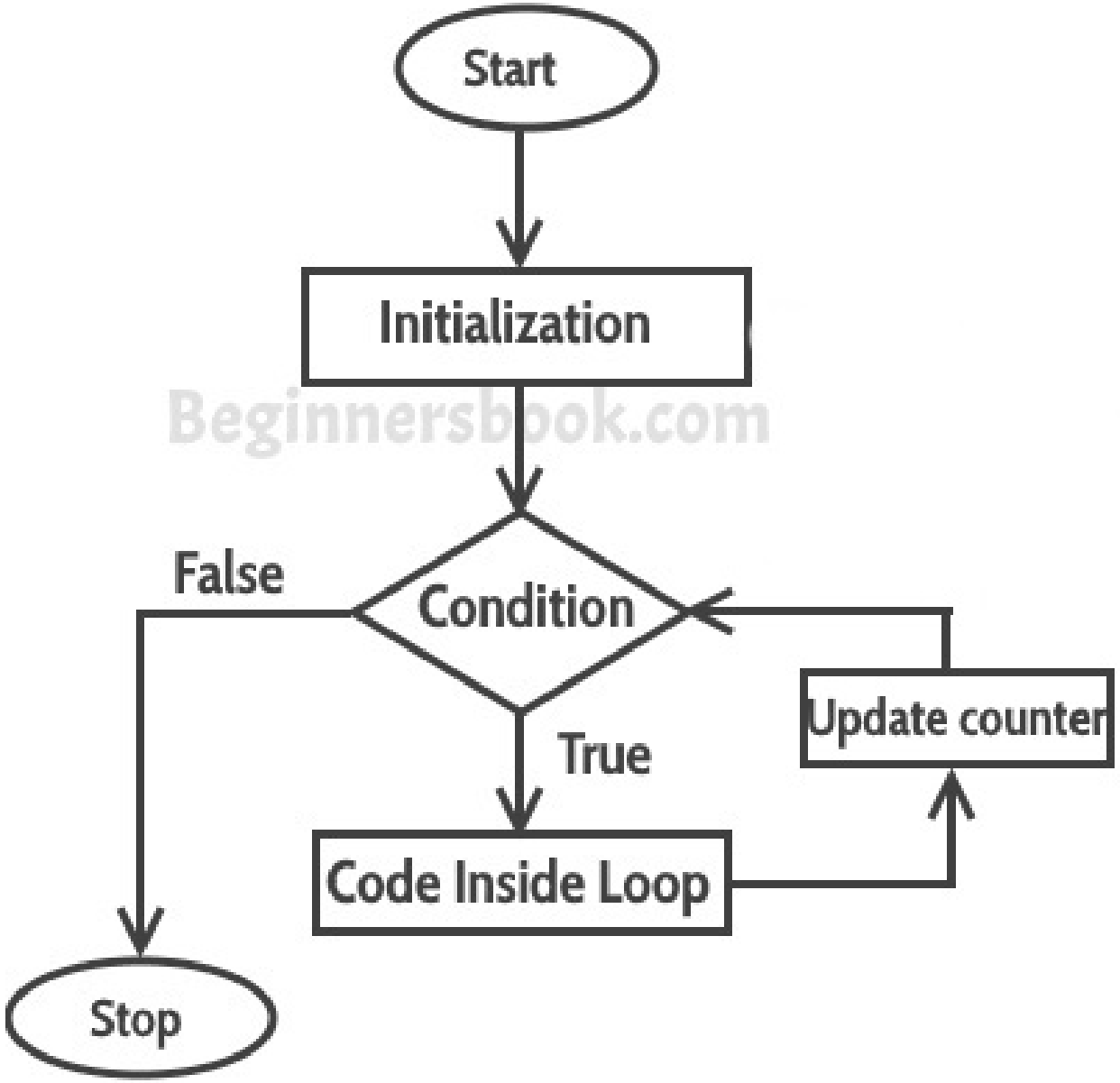
Break and **Continue** Statements are also provided by Python and can be used with other control statements

Python Indentation

- A unique feature of Python is lack of brackets for defining code blocks
- Python uses indentation to keep statement sets together
- Data read using pandas can be used with control statements

I will illustrate the use of these control statement using a few examples

For Statement



For Statement

- Read the Ogallala.csv file using pandas read_csv function
- Extract the Average Nitrate Concentrations (NO3Av) into a series
- Use the for loop to calculate the mean nitrate concentration in the wells
  - Round it to two decimal places
- Use the python statistics library to calculate the mean value to compare

$$\bar{x} = \frac{\sum_{i=0}^{(N-1)} x_i}{N}$$

Sample Mean

Steps:

1. Import pandas module
2. Import os module
3. Import statistics module for later use
4. Set working directory (where the 'ogallaladata.csv' file is)
5. Read the data file as a data frame (read_csv)
6. Extract NO3AV into a panda series
7. Calculate the length N (Number of data points)
8. Use for statement to loop through and sum values
9. Divide sum from step 4 by N from step 3 (mean)
10. Use the round function to round the result to 2 decimals
11. Compute mean using statistics module (round off)
12. Compare mean from step 7 to mean from step 9

For Statement

- Notice the indentation around the for statement
- Notice the colon

Stores the current element of the list

This should be a list (iterable)

```
for j in Y:
    do something 1
    do something 2
You are out of the loop
```

Indentation says this part of for loop

Indentation is removed to get out of the loop

```
# Script to calculate average NO3 Conc
# Venki Uddameri, 12/26/2019
# Import libraries
import pandas as pd
import os
import statistics as st

# set working directory
os.chdir('D:\Dropbox/000CE5333Machine Learning/Module3/Code')
a = pd.read_csv('OgallalaData.csv')


# Extract Average Nitrogen Concentration (NO3Av)
NO3 = a.NO3Av
N = len(NO3)  # Number of wells where NO3 is measured

# Use for loop to calculate the mean NO3
sum = 0.0
for i in NO3:
    sum = sum + i
mean = round(sum/N,2)  # Round to 2 decimals
mean # write the mean value to the console

round(st.mean(NO3),2) # Use statistics mean function
```

Both methods give a value of 26.23 mg/L

Unlike other programming languages (e.g., R) the index of iteration can be implicit in Python

For statement with range

- One can use the range statement to explicitly have an index of iteration

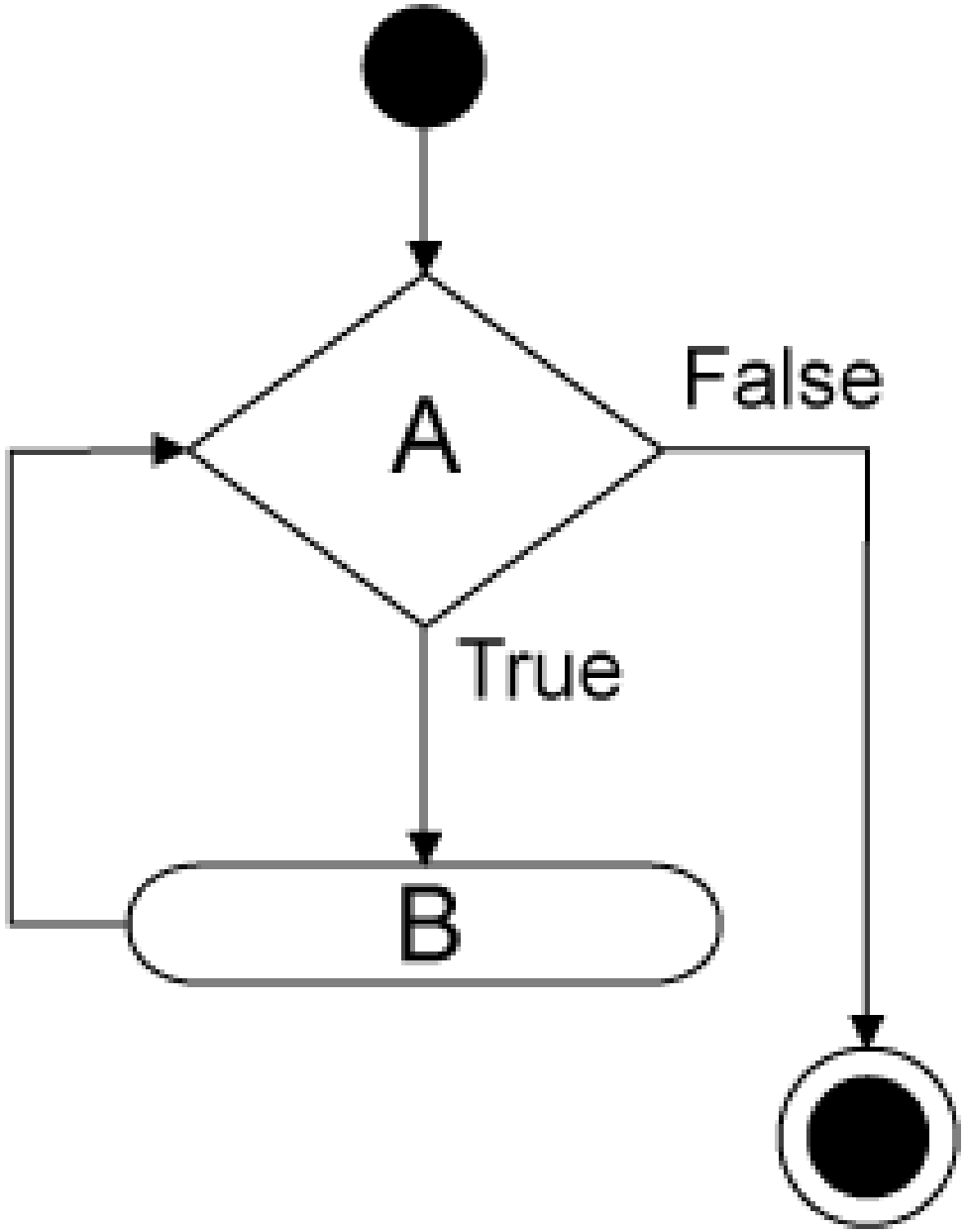
- Remember Python indexes from 0 to (N-1)
  - Avoid off-by-one error

Goes from 0 to N-1
(when N is specified)

```
# Use for loop with explicit range
idx = range(N) # create a sequence of numbers 0:(N-1)
sum = 0
for j in idx:
    sum = sum + NO3[j]
meanx = sum/N
meanx = round(meanx,2)
```

Make sure you specify N in Range function when you want to go from 0 to N-1
Python uses [) rule for Ranges

While Loop



While Loop

- A while loop is generally used when the number of iterations is unknown
  - But can also be known when the number of iterations are known

Indentation indicates these are part of the loop

while *condition = TRUE:*
do something 1
do something 2
do something 3
You are out of the loop

Usually the *condition* is updated within the loop

Illustrative Example

- Use the while loop to iterate through the AvNO3 data series to calculate its standard deviation
  - You can use the statistics library to compute the sample mean
  - You will need to import the math library to compute the square root (sqrt)
  - Compare the result with that obtained using statistics module

$$sd = \frac{\sum_0^{N-1} (x_i - \bar{x})^2}{N-1}$$

Code

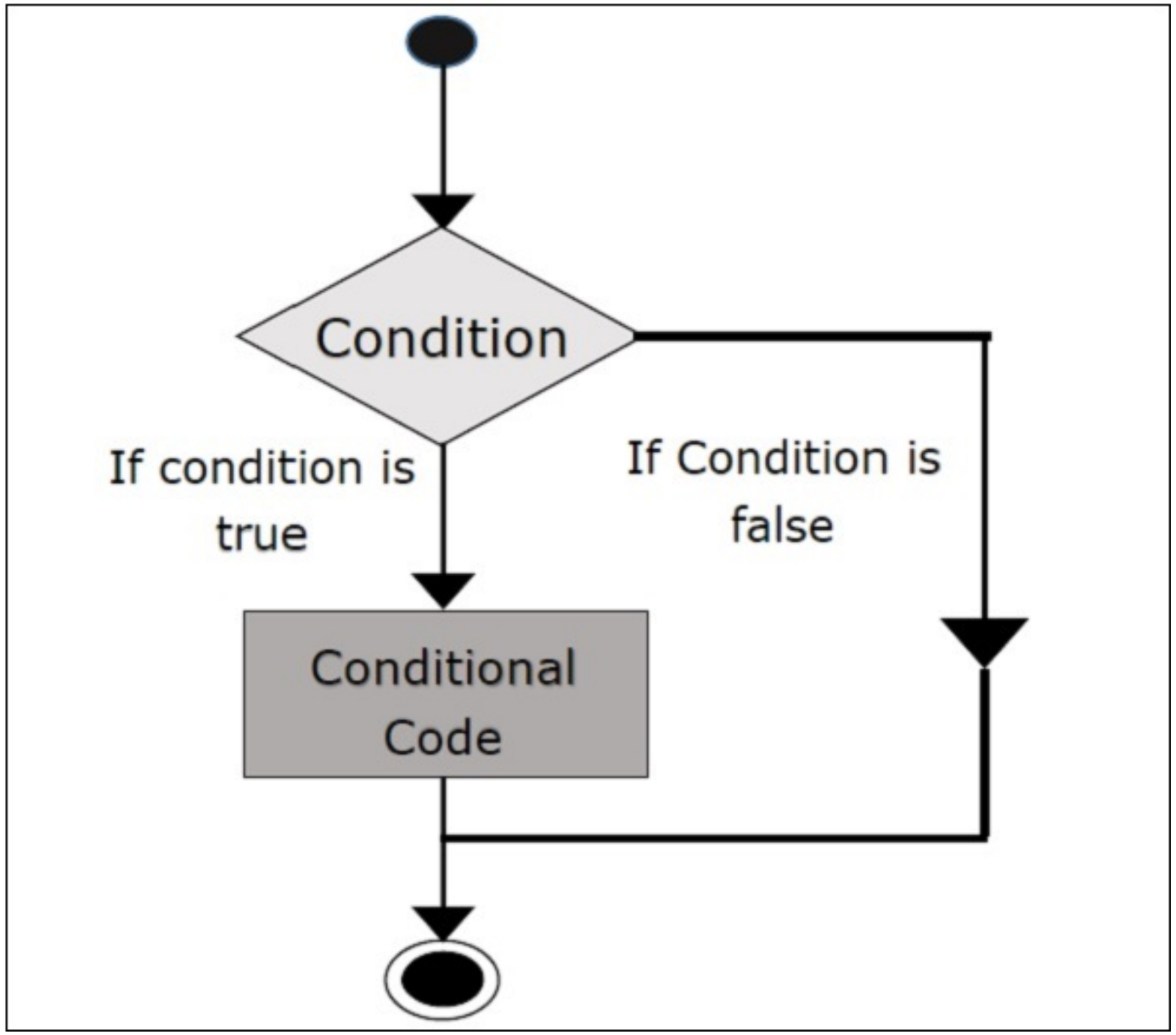
This is an 'augmented assignment' operator
Sum = Sum + z can be written as sum += z

```
# Script to calculate average NO3 Conc
# Venki Uddameri, 12/26/2019
# Import libraries
import pandas as pd
import os
import statistics as st
import math

# set working directory and read data
os.chdir('D:\Dropbox/000CE5333Machine
Learning/Module3/Code')
a = pd.read_csv('OgallalaData.csv')
NO3 = a.NO3Av  # extract NO3 data
```

```
xm = st.mean(NO3) # Compute the mean
N = len(NO3)  # Get the length of the data
idx = 0  # Set index to zero
sum = 0  # Set sum to zero
while (idx < N):  # Begin while loop
   sum += (NO3[idx]-xm)**2   # Add the difference square
   idx = idx + 1        #Update index
var = sum/(N-1)  # Compute variance
sd = math.sqrt(var)  # Take sqrt to obtain Std. Dev
round(sd,2)  # Round to 2 decimals

rund(st.stdev(NO3),2)  # Compute using stat module
```

Both methods should give value of 36.24 mg/L

IF Statement



Python If statement

- Python offers three variants of the if statement
  - If statement
  - If-else statement
  - If-elif-…-else statement

```
if condition:
    do something if condition is true
    do something if condition is true
    do something if condition is true
Get out of the loop
```

Program flow direction

```
if condition A:
    do something if condition A is true

elif condition B:
    do something if condition B is true
elif condition C:
    do something if condition C is true
else:
    do something if condition C is false
Get out of the loop
```

```
if condition:
    do something if condition is true
    do something if condition is true
else:
    do something if condition is false
Get out of the loop
```

Illustrative Example

- Compute the coefficient of variation of the Average Nitrate Concentrations for those wells that are in compliance with the drinking water standard (AvNO3 <= 10 mg/L) and those that are not (AvNO3 > 10 mg/L)
  - You can use the mean function in the statistics library
  - Compute the variance and standard deviation by summing up appropriately
- Subset the data using pandas loc method and check your calculations

$$COV = \frac{SD}{\bar{x}}$$

Code

```
# Script to calculate COV for contaminated and not
contaminated
# Venki Uddameri, 12/26/2019
# Import libraries
import pandas as pd
import os
import statistics as st
import math

# set working directory
os.chdir('D:\Dropbox/000CE5333Machine
Learning/Module3/Code')
a = pd.read_csv('OgallalaData.csv')


# Extract Average Nitrogen Concentration (NO3Av)
NO3 = a.NO3Av
N = len(NO3)  # Number of wells where NO3 is measured
```

```
# Estimate Mean values
# Initialize variables c for contaminated n of not
sumc = 0
sumn = 0
lenc = 0
lenn = 0
for i in NO3:  # loop through all samples
    if(i > 10.0):  #update if contaminated
        sumc += i
        lenc += 1
    else:  # Update if not contaminated
        sumn += i
        lenn += 1
NO3cav = round(sumc/lenc,2)  # Final Average Contaminated
NO3nav = round(sumn/lenn,2)  # Final Average Not Cont.
```

Code Cont..

```
# Estimate Standard Deviations
sumc = 0
sumn = 0
for i in NO3:  # loop through all samples
   if(i > 10.0):  #update if contaminated
      sumc += (i-NO3cav)**2
   else:  # Update if not contaminated
      sumn +=  (i-NO3nav)**2
NO3csd = math.sqrt(sumc/(lenc-1))  # Final SD Contaminated
NO3nsd = math.sqrt(sumn/(lenn-1))  # Final SD Not
Contaminated
# Calculate Coefficient of Variation
NO3ccov = round(NO3csd/NO3cav,2) # Round and write to
console
NO3ncov = round(NO3nsd/NO3nav,2) # Round and write to
console
```

(Both methods give same results)

COV contaminated	1.00
COV not contaminated	0.37

```
# Estimating COV using Pandas Subsetting
NO3c = a.loc[a['NO3Av'] > 10,['NO3Av']] #Subset contaminated
NO3c = NO3c['NO3Av'].tolist() # Convert to list
NO3COVc = round(st.stdev(NO3c)/st.mean(NO3c),2) # Round COV

NO3n = a.loc[a['NO3Av'] <=10,['NO3Av']] # Subset Not Cont.
NO3n = NO3n['NO3Av'].tolist() # Convert to list
NO3COVn = round(st.stdev(NO3n)/st.mean(NO3n),2) # Round COV
```

Subset Pandas dataframe and convert it into a python list to pass to stdev and mean functions in statistics package

Boolean Operators

- Decision (control) Statements can have more than one criterion
- Python offers two Boolean operators
  - **and** is used when more than one criteria have to be simultaneously met
  - **or** is used when at least obe criteria is to be met

Find the mean NO3Av Concentration for shallow well defined here as
WellDepth < 100 ft **or** depth to water table (DWT) < 50 ft

Compare by subsetting using pandas

Code

```
# Script to calculate mean NO3 of Shallow Wells
# Venki Uddameri, 12/26/2019
# Import libraries
import pandas as pd
import os
import statistics as st
import math

# set working directory
os.chdir('D:\Dropbox/000CE5333Machine
Learning/Module3/Code')
a = pd.read_csv('OgallalaData.csv')
```

```
# Extract Average Nitrogen Concentration (NO3Av)
NO3 = a.NO3Av
DWT = a.DWT
WD = a.WellDepth
N = len(NO3)  # Number of wells where NO3 is measured
# Initialize variables
sum = 0.0
len = 0
for i in range(N): # Loop through all wells
   if(WD[i]<100 or DWT[i] < 50): # Check shallow condition
      sum += NO3[i]
      len += 1
NO3sh = round(sum/len,2) #Compute shallow
```

Code – Subsetting using pandas

```
# Subset using Pandas iloc statement
# Note Pandas uses & (and) and | (or) as Boolean Operators
NO3sw = a.loc[(a['DWT'] <50) | (a['WellDepth'] <
100),['NO3Av']]
NO3sw = NO3sw['NO3Av'] # Convert to list
round(st.mean(NO3sw),2)
```

You should Know

- Control statements in R
  - if statement
  - if-else
  - if-elif-else
  - for loop
  - while loop
- Boolean Statement
  - **and**
  - **or**

Pandas uses & and | as Boolean operators within loc