Welcome to BBM 101

• This course teaches **core programming concepts** with an emphasis on **data manipulation tasks** from science, engineering, and business

• **Goal** by the end of the semester: Given a **data source** and a **problem description**, you can independently write a complete, useful program to solve the problem

Course Staff

• **Lecturers:**
  – Asst. Prof. Dr. Fuat Akal
  – Asst. Prof. Dr. Aykut Erdem
  – Asst. Prof. Dr. Erkut Erdem

• **TAs (Teaching Assistants):**
  – Necva Bölücü
  – Selma Dilek
  – Burcu Yalçınker
  – Selim Yılmaz

Do not hesitate to ask TAs for help!
Learning Objectives

• Computational problem-solving
  – Writing a program will become your “go-to” solution for data analysis tasks.

• Basic Python proficiency
  – Including experience with relevant libraries for data manipulation, scientific computing, and visualization.

• Experience working with real datasets
  – astronomy, biology, linguistics, oceanography, open government, social networks, and more.
  – You will see that these are easy to process with a program, and that doing so yields insight.

What This Course is not

• A “skills course” in Python
  – ...though you will become proficient in the basics of the Python programming language
  – ...and you will gain experience with some important Python libraries

• A data analysis / “data science” / data visualization course
  – There will be very little statistics knowledge assumed or taught

• A “project” course
  – the assignments are “real,” but are intended to teach specific programming concepts

• A “software engineering” course
  – Programming is the starting point of computer science and software engineering

All of Science is Reducing to Computational Data Manipulation

Old model: “Query the world” (Data acquisition coupled to a specific hypothesis)
New model: “Download the world” (Data acquisition supports many hypotheses)
– Astronomy: High-resolution, high-frequency sky surveys (SDSS, LSST, PanSTARRS)
– Biology: lab automation, high-throughput sequencing,
– Oceanography: high-resolution models, cheap sensors, satellites

“It’s a great time to be a data geek.”
-- Roger Barga, Microsoft Research

“The greatest minds of my generation are trying to figure out how to make people click on ads”
-- Jeff Hammerbacher, co-founder, Cloudera
**Example: Assessing Treatment Efficacy**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>fu_2wk</td>
<td>fu_4wk</td>
<td>fu_6wk</td>
<td>fu_12wk</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>12</td>
<td>98405</td>
<td>98405</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>98405</td>
<td>98405</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>98404</td>
<td>98405</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>12</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>12</td>
<td>98404</td>
<td>98332</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>98405</td>
<td>98405</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>14</td>
<td>98499</td>
<td>98418</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>98499</td>
<td>98406</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>98499</td>
<td>98445</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>98499</td>
<td>98405</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>98499</td>
<td>98498</td>
<td></td>
</tr>
</tbody>
</table>

**Question:** Does the distance between the patient’s home and clinic influence the number of follow-ups, and therefore treatment efficacy?

---

**Python Program to Assess Treatment Efficacy**

```python
# This program reads an Excel spreadsheet whose penultimate and antepenultimate columns are zip codes.
# It adds a new last column for the distance between those zip codes, and outputs in CSV (comma-separated-valued) format.
# Call the program with two numeric values: the first and last row to include.
# The output contains the column headers and those rows.
# Libraries to use
import random
import sys
import xlrd
import time
from gdapi import GoogleDirections
# No key needed if few queries
gd = GoogleDirections('dummy-Google-key')
wb = xlrd.open_workbook('mhip_zip_eScience_121611a.xls')
sheet = wb.sheet_by_index(0)
# User input: first row to process, first row not to process
first_row = max(int(sys.argv[1]), 2)
row_limit = min(int(sys.argv[2]) + 1, sheet.nrows)
def comma_separated(lst):
    return ','.join([str(s) for s in lst])
headers = sheet.row_values(0) + ['"distance"']
print comma_separated(headers)
for rownum in range(first_row, row_limit):
    row = sheet.row_values(rownum)
    (zip1, zip2) = row[-3:-1]
    # Clean the data
    zip1 = str(int(zip1))
    zip2 = str(int(zip2))
    row[-3:-1] = [zip1, zip2]
    # Compute the distance via Google Maps
    try:
        distance = gd.query(zip1, zip2).distance
e
except:
    print >> sys.stderr, "Error computing distance:" , zip1, zip2
distance = ""

# Print the row with the distance
print comma_separated(row + ["distance"])  # Avoid too many Google queries in rapid succession
```

---

**Some statistics (from U.S.)**

- **The value of a computer science education**
  - $0.58M lifetime earnings of a high school graduate
  - $1.19M lifetime earnings of a college graduate
  - $1.67M lifetime earnings of a computer science major

- **40% more than the average earnings**

- **$2.9M** social science
- **$1.1M** installation and repair
- **$0.8M** business

**Source:** Brookings

---

**Some statistics (from U.S.)**

- **Computing jobs are the #1 source of new wages in the United States**

- **500,000 current openings:** These jobs are in **every** industry and **every** state, and they’re projected to grow at twice the rate of all other jobs.

**Source:** code.org

---

*Slide credit: code.org*
Some statistics (from U.S.)
The STEM* problem is in computer science:

- 71% of all new jobs in STEM are in computing
- 8% of STEM graduates are in computer science

Sources: Bureau of Labor Statistics, National Center for Education Statistics

Slide credit: code.org

*STEM = Science, Technology, Engineering, and Math

Course Logistics

- Website: http://web.cs.hacettepe.edu.tr/~bbm101/
- See the website for all administrative details
- Read the handouts and required texts, before the lecture
- Take notes!
- Follow the course in Piazza
  https://piazza.com/hacettepe.edu.tr/fall2016/bbm101

Academic Integrity

- Honest work is required of a scientist or engineer.
- Collaboration policy on the course web. Read it!
  - Discussion is permitted.
  - Carrying materials from discussion is not permitted.
  - Everything you turn in must be your own work.
    - Cite your sources, explain any unconventional action.
  - You may not view others’ work.
  - If you have a question, ask.
- We trust you completely.
- But we have no sympathy for trust violations – nor should you!

How to Succeed

- No prerequisites
- Non-predictors for success:
  - Past programming experience
  - Enthusiasm for games or computers
- Programming and data analysis are challenging
- Every one of you can succeed
  - There is no such thing as a “born programmer”
  - Work hard
  - Follow directions
  - Be methodical
  - Think before you act
  - Try on your own, then ask for help
  - Start early
**Integrated Development Environment (IDE)**

- There are many!

**Choosing the Best Python IDE**

In this article, I’ll review six Python IDEs. I’m mainly interested in IDEs that are cross-platform and have strong web development support (Django, HTML templates, JavaScript, etc.). Because of this, well-regarded IDEs like PyCharm and Python Tools for Visual Studio are out since they are Windows-only. The Python website maintains a full list of Python IDEs.

**PyCharm**

PyCharm is one of the most popular Python IDEs and deservedly so. It’s packed with features such as incredible code completion, code analysis, code navigation, top-notch Django, JavaScript, HTML, and CSS support, great debugger, and much more.

---

**Python Version**

- Whatever IDE you choose to work with, always stick to **Python version 3.5.2**

- **Always** use this version to code your assignments.

---

**Books**

- There are many!

---

**Our Recommendation: PyCharm**
Our Recommendation for Books

- **The Python Tutorial**, available from the Python website.
  - This is good for explaining the nuts and bolts of how Python works.


- **Think Python, 2nd edition**
  - Freely available online in [HTML](#) and [PDF](#).
  - Also available for purchase as a printed book, but don’t buy the first edition.
  - This book introduces more conceptual material, motivating computational thinking.

- There is an interactive version of “How to Think Like a Computer Scientist” (the first edition of “Think Python”), which lets you type and run Python code directly while reading the book.