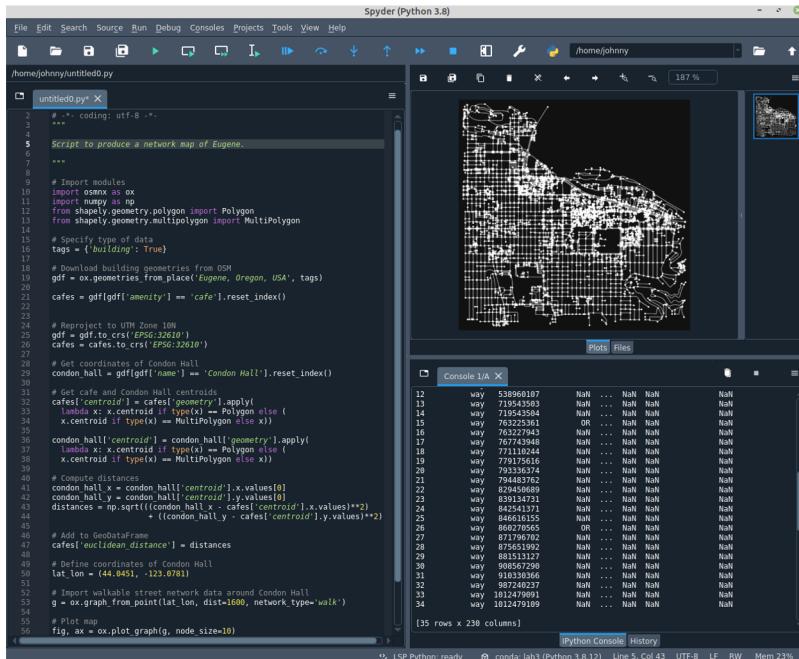


# Geospatial Data Science Applications: GEOG 4/590

## Lecture 1: Introduction



The screenshot shows the Spyder Python IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help. The left sidebar shows a file tree with 'untitled0.py' selected. The main code editor window contains a Python script for generating a network map of Eugene, Oregon. The script uses libraries like osmnx, shapely, and geopandas. It downloads building geometries from OSM, filters for cafes, reprojects to UTM Zone 10N, and computes distances between cafe centroids and Condon Hall's centroid. The right side of the interface features a map visualization showing the network and a data preview window. Below the code editor is a 'Console 1/A X' window displaying a table of data with columns for way, x, y, and various NaN values. The bottom status bar indicates 'LSP Python: ready' and 'conda-lab (Python 3.8.12) Line 5, Col 43 UTF-8 LF RW Mem 23%'. A small footer note at the bottom left says 'IPython Console History'.

```
# -*- coding: utf-8 -*-
"""
Script to produce a network map of Eugene.
"""

# Import modules
import osmnx as ox
import geopandas as gp
from shapely.geometry.polygon import Polygon
from shapely.geometry.multipolygon import MultiPolygon

# Specify type of data
tags = {'building': True}

# Download building geometries from OSM
gdf = ox.geometries_from_place('Eugene, Oregon, USA', tags)

cafes = gdf[gdf['amenity'] == 'cafe'].reset_index()

# Reproject to UTM Zone 10N
gdf = gdf.to_crs('EPSG:32610')
cafes = cafes.to_crs('EPSG:32610')

# Get coordinates of Condon Hall
condon_hall = gdf[gdf['name'] == 'Condon Hall'].reset_index()

# Get cafe and Condon Hall centroids
cafes['centroid'] = cafes['geometry'].apply(
    lambda x: x.centroid if type(x) == Polygon else (
        x.centroid if type(x) == MultiPolygon else x))

condon_hall['centroid'] = condon_hall['geometry'].apply(
    lambda x: x.centroid if type(x) == Polygon else (
        x.centroid if type(x) == MultiPolygon else x))

# Compute distances
condon_hall_x = condon_hall['centroid'].x.values[0]
condon_hall_y = condon_hall['centroid'].y.values[0]
distances = np.sqrt((condon_hall_x - cafes['centroid'].x.values)**2 +
                     (condon_hall_y - cafes['centroid'].y.values)**2)

# Add to GeoDataFrame
cafes['euclidean_distance'] = distances

# Define coordinates of Condon Hall
lat_lon = (44.0451, -123.0781)

# Import walkable street network data around Condon Hall
g = ox.graph_from_point(lat_lon, dist=1000, network_type='walk')

# Plot map
fig, ax = ox.plot_graph(g, node_size=10)
```

Johnny Ryan

[jryan4@uoregon.edu](mailto:jryan4@uoregon.edu)

Office hours: Monday 15:00-17:00

# Welcome

Dr. Johnny Ryan, Assistant Professor of Geography

- BSc in Geography at University of Nottingham, UK
- MPhil in Polar Studies at University of Cambridge, UK
- PhD in Geography at Aberystwyth University, UK
- Researching glaciology, hydrology, remote sensing

Email: [jryan4@uoregon.edu](mailto:jryan4@uoregon.edu)

Office hours: Monday 15:00-17:00

# COVID Policies

- All classes will be held **in-person** in accordance with the latest university policies
- Mask wearing is mandatory indoors for all students
- Please do not come to class or lab if you are feeling sick or have been exposed
- Let me and/or Insang know if this happens and we will work with you to make up the missed materials.

# COVID Policies

Time to get to know your neighbors!

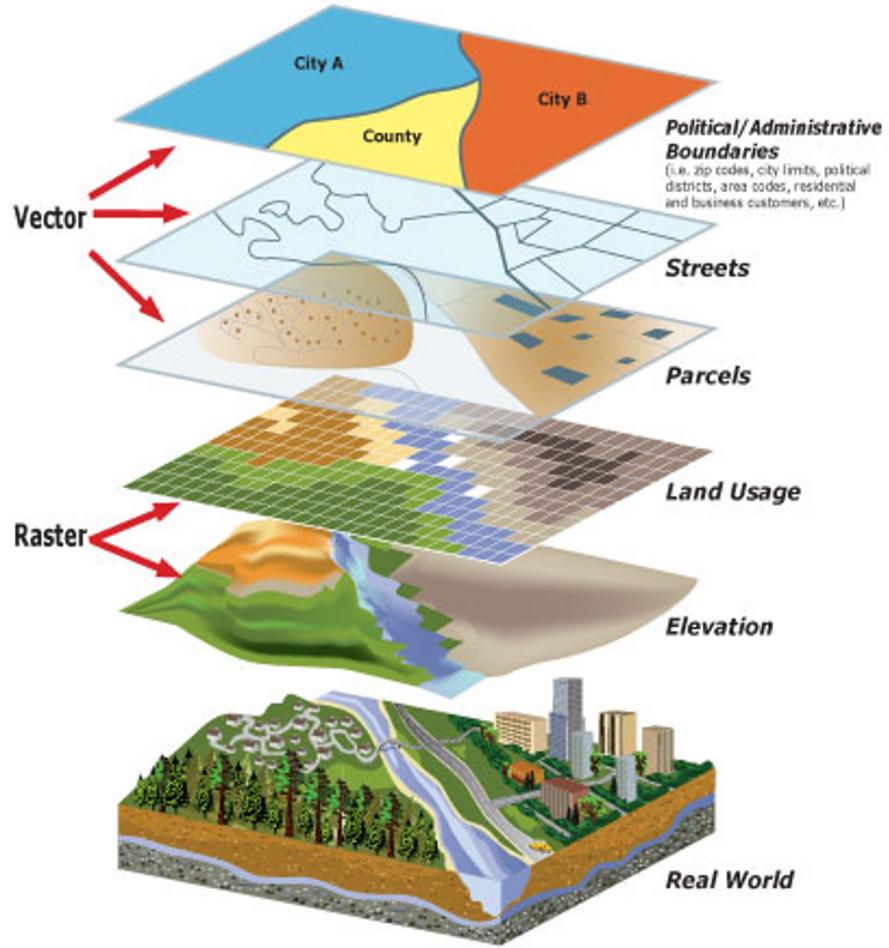
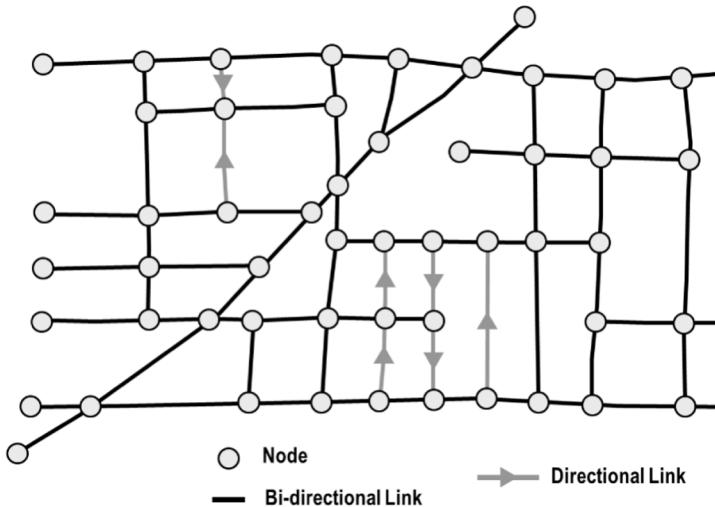
- Please introduce yourself with:
  - your name
  - your year
  - where you're from
  - your favorite place to eat in Eugene

# Overview

- What is Geospatial Data Science?
  - Spatial data (e.g. geometries and projections)
  - Coding (e.g. Python)
  - Collaborating (e.g. GitHub, Slack)
- Course schedule
  - Lectures, labs, grading
- Final project
- Some tasks
  - Complete background survey on Canvas
  - Join Slack workspace
  - Getting started in Wed/Fri lab session

# Geospatial data

- Vector data
- Network data
- Raster data



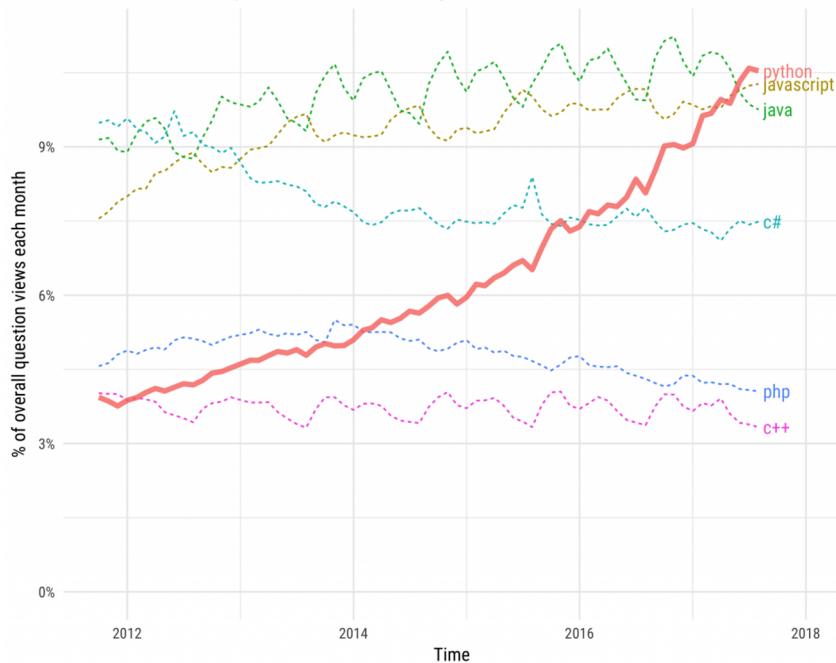
# Python

- Popular high-level programming language
- Easy-to-read
- Extensive library
- Free and open-source
  - Accessible
  - Can be examined, modified, and improved
- Constantly evolving



## Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



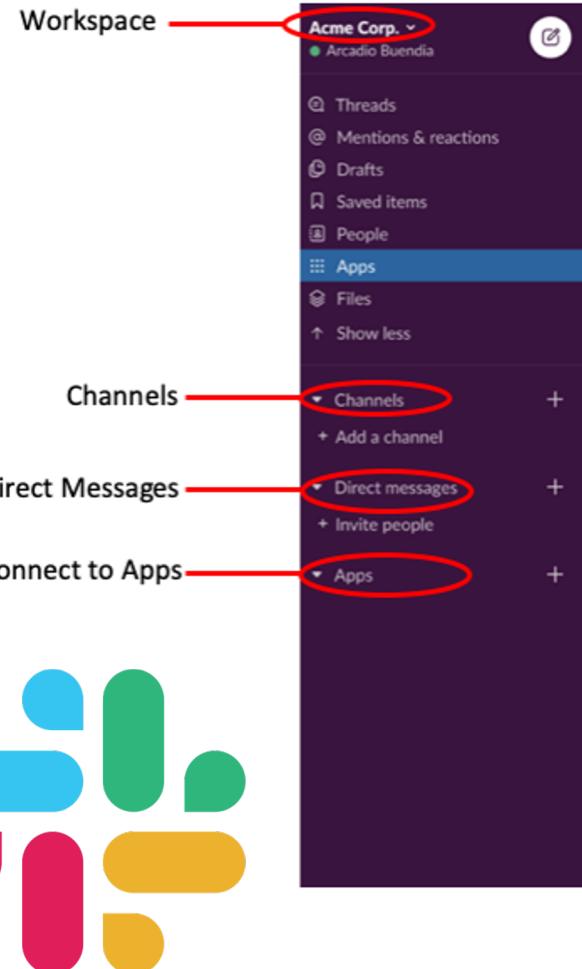
# Git and GitHub

- Git
  - Version control software for tracking changes to a set of files
- GitHub
  - A cloud-based Git repository hosting service
  - Makes it easier to coordinate work among programmers collaboratively developing source code during software development
- Python and projects that use Python (e.g. matplotlib) are all maintained and developed by a community of scientists and programmers on GitHub
- An active, up-to-date GitHub profile, with contributions to open-source project is a great way to provide evidence of skills



# Slack

- Messaging app that facilitates collaboration by having all features in one place
- Starting to replace email, more secure, organized
- You should have received invitation to join course workspace
  - Use the standalone **desktop app** rather than the web interface
- We will use Slack for announcements, discussion, and questions
  - Asking questions on Slack is **not** cheating!



# Course schedule

- **Lectures:** Monday 2pm in 106 Condon Hall
  - 8 lectures
  - Holiday on Jan 17, class presentations in Week 10
- **Labs:** Wednesday @ 2pm or Friday 10am in ~~442 MCK~~ 207 Condon Hall
  - 7 labs
  - 2 labs to concentrate on final project
  - Class presentations in Week 10

# Course evaluation

- Lab assignments (45%) due every **Friday 11:59 pm**
- Final project (45%)
  - Presentation due **March 6, 11:59 pm**
  - Write-ups due **March 11, 11:59pm**
- Participation (10%)
  - Credit can be earned through attendance in lectures, participation in class discussion (both virtual and in-person), visiting Professor and GE during office hours, and helping other students in labs.

Week	Date	Lecture	Lab
1	Jan 3	Introduction	Getting started with Python and GitHub
2	Jan 10	Table data	Wildfire and Census data
3	Jan 17	Network data	Walking distances
4	Jan 24	Gridded data	Remote sensing and climate reanalysis data
5	Jan 31	Machine learning	Three machine learning examples
6	Feb 7	Data/code management	Get started on project
7	Feb 14	Data access	Application 1
8	Feb 21	Visualization	Project work
9	Feb 28	Ethics and responsibility	Application 2
10	Mar 7	Project presentation	Project presentation

Working with geospatial data

Week	Date	Lecture	Lab
1	Jan 3	Introduction	Getting started with Python and GitHub
2	Jan 10	Table data	Wildfire and Census data
3	Jan 17	Network data	Walking distances
4	Jan 24	Gridded data	Remote sensing and climate reanalysis data
5	Jan 31	Machine learning	Three machine learning examples
6	Feb 7	Data/code management	Get started on project
7	Feb 14	Data access	Application 1
8	Feb 21	Visualization	Project work
9	Feb 28	Ethics and responsibility	Application 2
10	Mar 7	Project presentation	Project presentation

Working with geospatial data

Machine learning

Week	Date	Lecture	Lab
1	Jan 3	Introduction	Getting started with Python and GitHub
2	Jan 10	Table data	Wildfire and Census data
3	Jan 17	Network data	Walking distances
4	Jan 24	Gridded data	Remote sensing and climate reanalysis data
5	Jan 31	Machine learning	Three machine learning examples
6	Feb 7	Data/code management	Get started on project
7	Feb 14	Data access	Application 1
8	Feb 21	Visualization	Project work
9	Feb 28	Ethics and responsibility	Application 2
10	Mar 7	Project presentation	Project presentation

Working with geospatial data

Machine learning

Key skills

Week	Date	Lecture	Lab
1	Jan 3	Introduction	Getting started with Python and GitHub
2	Jan 10	Table data	Wildfire and Census data
3	Jan 17	Network data	Walking distances
4	Jan 24	Gridded data	Remote sensing and climate reanalysis data
5	Jan 31	Machine learning	Three machine learning examples
6	Feb 7	Data/code management	Get started on project
7	Feb 14	Data access	Application 1
8	Feb 21	Visualization	Project work
9	Feb 28	Ethics and responsibility	Application 2
10	Mar 7	Project presentation	Project presentation

Working with geospatial data

Machine learning

Key skills

What we *should* and *shouldn't* do

# Final project

- An opportunity to explore a particular topic of interest using some of the skills developed in this course
- Students can work independently or in groups of two or three
- Sharing of project ideas on Slack is encouraged so we can form teams

# Final project schedule

- **Week 5:** Discuss project ideas with peers and instructors, submit a short summary of a project idea on the #final-projects Slack channel
- **Week 6:** Form teams, create GitHub repo, and provide some basic info about project as a README.md
- **Week 8:** Provide informal update to instructors, ensure data has been accessed, goals are accomplishable
- **Week 10:** Present project to class and submit write-up by the end of the week

# Final project schedule

- **Week 5:** Discuss project ideas with peers and instructors, submit a short summary of a project idea on the #final-projects Slack channel
- **Week 6:** Form teams, create GitHub repo, and provide some basic info about project as a README.md
- **Week 8:** Provide informal update to instructors, ensure data has been accessed, goals are accomplishable
- **Week 10:** Present project to class and submit write-up by the end of the week

# Final project schedule

- **Week 5:** Discuss project ideas with peers and instructors, submit a short summary of a project idea on the #final-projects Slack channel
- **Week 6:** Form teams, create GitHub repo, and provide some basic info about project as a README.md
- **Week 8:** Provide informal update to instructors, ensure data has been accessed, goals are accomplishable
- **Week 10:** Present project to class and submit write-up by the end of the week

# Final project schedule

- **Week 5:** Discuss project ideas with peers and instructors, submit a short summary of a project idea on the #final-projects Slack channel
- **Week 6:** Form teams, create GitHub repo, and provide some basic info about project as a README.md
- **Week 8:** Provide informal update to instructors, ensure data has been accessed, goals are accomplishable
- **Week 10:** Present project to class and submit write-up by the end of the week

# Final project schedule

- **Week 5:** Discuss project ideas with peers and instructors, submit a short summary of a project idea on the #final-projects Slack channel
- **Week 6:** Form teams, create GitHub repo, and provide some basic info about project as a README.md
- **Week 8:** Provide informal update to instructors, ensure data has been accessed, goals are accomplishable
- **Week 10:** Present project to class and submit write-up by the end of the week

*I will send reminders out about upcoming milestones*

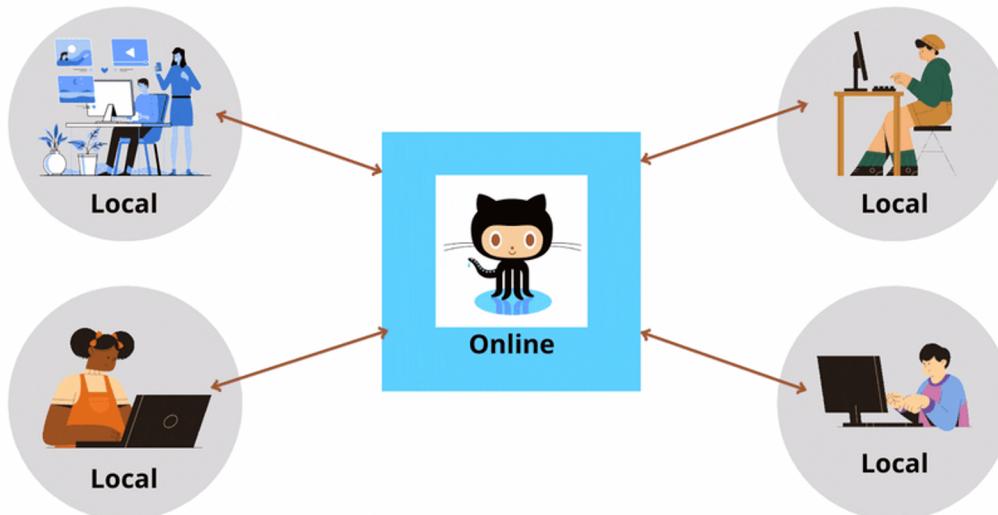
## Some course themes

- Everything is open-source
    - All software we use is freely available
    - Labs can be completed anytime, anywhere from any OS
    - Course materials are publicly-available on the internet



# Some course themes

- Promote collaboration and communication
  - With instructors **and** peers
  - On GitHub and Slack



# Some course themes

- Learn about environmental challenges in the Western US
  - Urban planning and zoning
  - Hazards (e.g. wildfires, flooding)
  - Energy, climate, hydrology



## Some course themes

- Don't try and write perfect code, if it works, it works
- Adapt code, don't write code
- Make use of stackoverflow
- Ask peers
- Be scrappy

when stack overflow doesn't help solve your problems



## Some course themes

- Take responsibility for learning
- Organize your files
- Check Canvas and Slack regularly
- Maintain your GitHub profile and repository



Some course themes

Learn by **DOING.**



# By the end of this course you will...

- Have confidence using Python specifically for GIS and other geospatial data science applications
  - In doing so, you will also be comfortable using Python for other things as well
- Be able to download, process, analyze, and visualize the main types of geospatial data
- Automate boring GIS tasks (no more clicking!)

# By the end of this course you will...

- Learn how to solve real-world problems using spatial analysis
- Understand basic machine learning concepts for data science
- Collaborate on software development using version control
- Communicate results of data science project orally and as short write-up

# Getting started on this week's lab

- Go to course page on GitHub:
  - <https://github.com/JohnnyRyan1/geospatial-data-science>
  - See Slack (or Canvas assignment) for links
- Operating systems... Linux, OSX are great, Windows is OK could consider using Windows Subsystem for Linux

