Network Analyst: Using the Python API

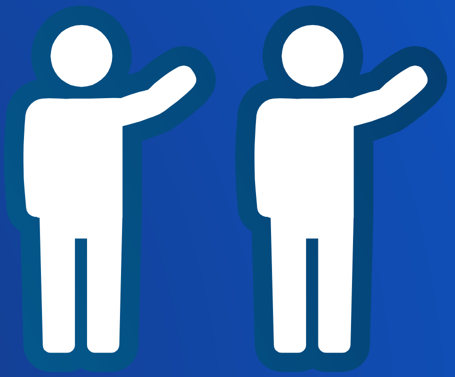
Deelesh Mandloi

Jeff Wickstrom

SEE WHAT OTHERS CAN'T

Introductions

- Who are we?
  - Members of the Network Analyst development team
- Who are you?
  - Network Analyst users?
  - Comfortable with Python code?
  - Familiar with arcpy and arcpy.na?



Metadata

- Slides and code samples for this workshop on Using the Python API are available at

http://esriurl.com/uc19nax

Topics

- What is Network Analyst
  - Types of analysis
  - Network dataset
- Network Analyst Python API (arcpy.nax)
  - Performing analysis
  - Working with network dataset

This workshop is not about…

- Performing analysis on utility networks
- Using ArcGIS API for Python (i.e. arcgis Python package)
  - It is possible to perform network analysis using the ArcGIS API for Python (arcgis.network module)
  - arcgis.network allows you to perform network analysis using web services
- Performing network analysis using ArcMap
  - Everything we talk today is only available with ArcGIS Pro 2.4 or later

Network Analyst Concepts

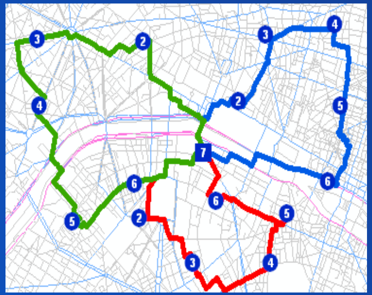
ArcGIS Network Analyst

Solve Transportation Problems

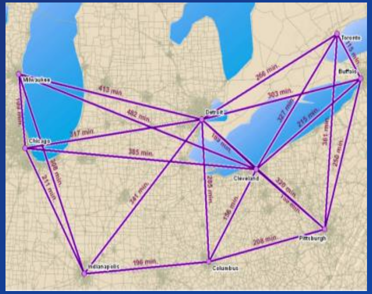
Route

Closest Facility

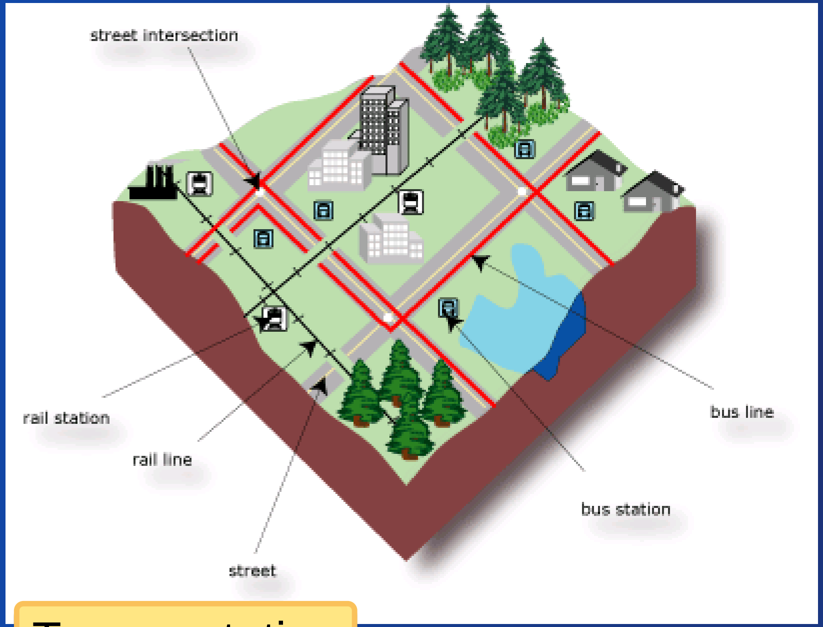
Service Area


Vehicle Routing Problem

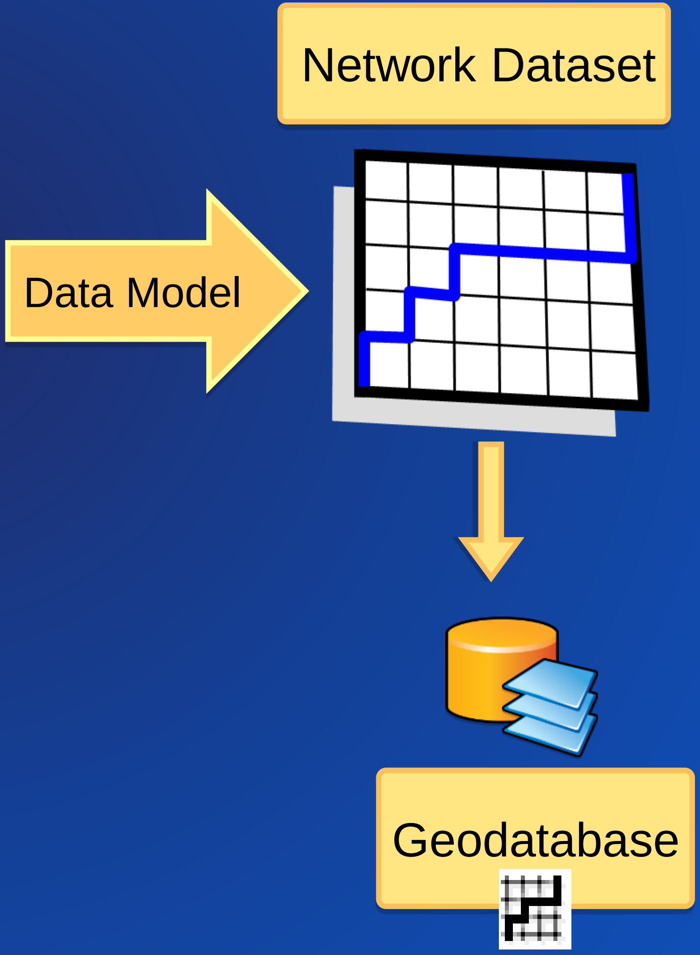
Location-Allocation


Origin-Destination Cost Matrix

Analysis is performed on a network dataset



Transportation Network



Where do I get a network dataset?

- Purchase StreetMap Premium for ArcGIS
  - High quality ready-to-use network dataset
  - Can add your own street data as well

OR

- Build your own network dataset from:
  - Your organization's data
  - TIGER
  - OpenStreetMap (try ArcGIS Editor for OpenStreetMap)

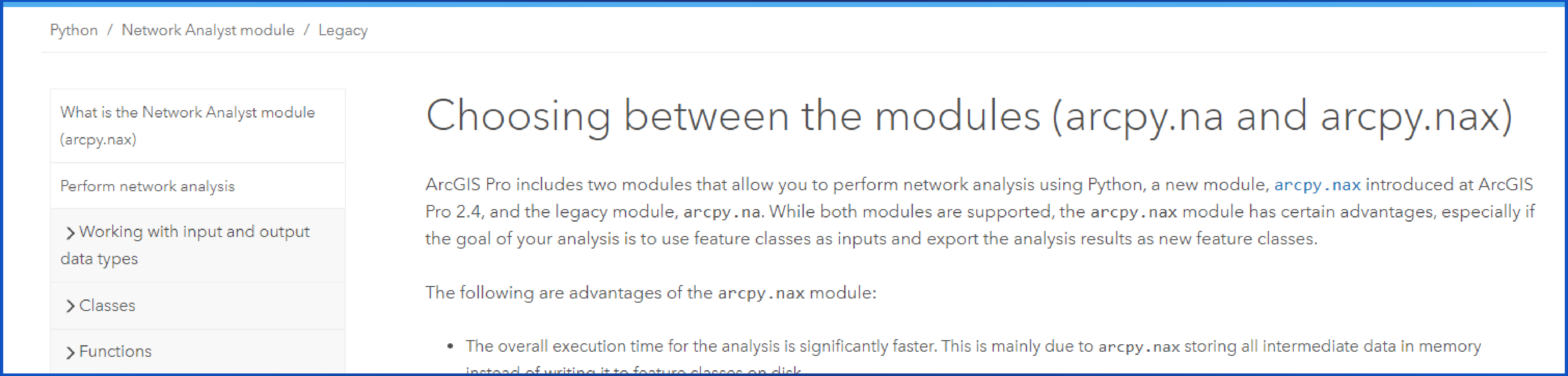
Network Analyst Module (arcpy.nax)

arcpy.nax

- Easy to use Python module that allows you to
  - Perform network analysis
  - Access connectivity information from a network dataset
- New with ArcGIS Pro 2.4

arcpy.nax vs arcpy.na

- Both arcpy.na and arcpy.nax allows you to perform network analysis using Python
- Help



Python / Network Analyst module / Legacy

What is the Network Analyst module (arcpy.nax)

Perform network analysis

> Working with input and output data types

> Classes

> Functions

Choosing between the modules (arcpy.na and arcpy.nax)

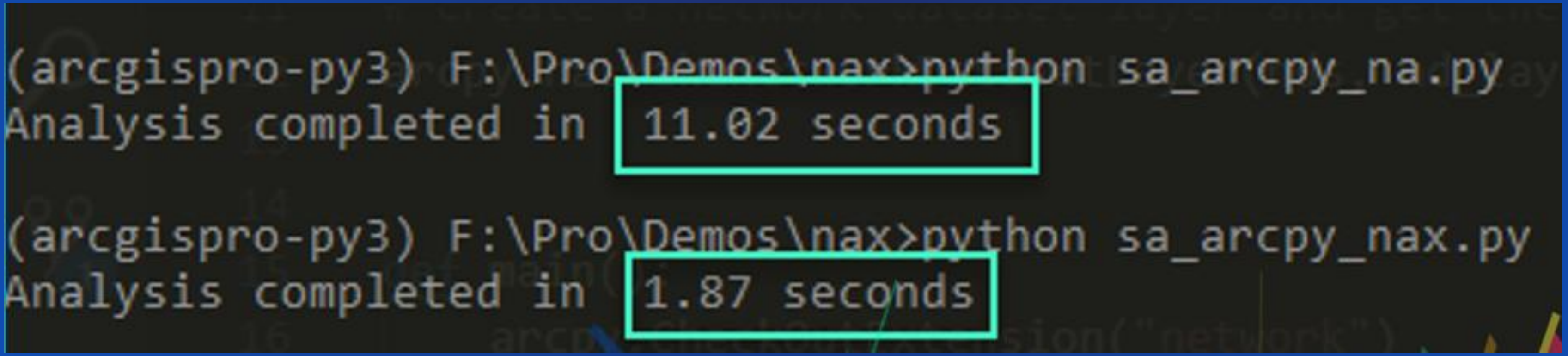
ArcGIS Pro includes two modules that allow you to perform network analysis using Python, a new module, `arcpy.nax` introduced at ArcGIS Pro 2.4, and the legacy module, `arcpy.na`. While both modules are supported, the `arcpy.nax` module has certain advantages, especially if the goal of your analysis is to use feature classes as inputs and export the analysis results as new feature classes.

The following are advantages of the `arcpy.nax` module:

- The overall execution time for the analysis is significantly faster. This is mainly due to `arcpy.nax` storing all intermediate data in memory

Benefits of using arcpy.nax

- ArcGIS Online network analysis services use arcpy.nax
- Provides a modern and easy to use Python API (aka Pythonic)
- Significantly faster analysis time as all data is stored in memory and not in file geodatabase



```
(arcgispro-py3) F:\Pro\Demos\nax>python sa_arcpy_na.py
Analysis completed in 11.02 seconds

(arcgispro-py3) F:\Pro\Demos\nax>python sa_arcpy_nax.py
Analysis completed in 1.87 seconds
```

5,10,15 minutes drive time areas around a single facility and exporting results to a file geodatabase feature class

arcpy.na is still required if

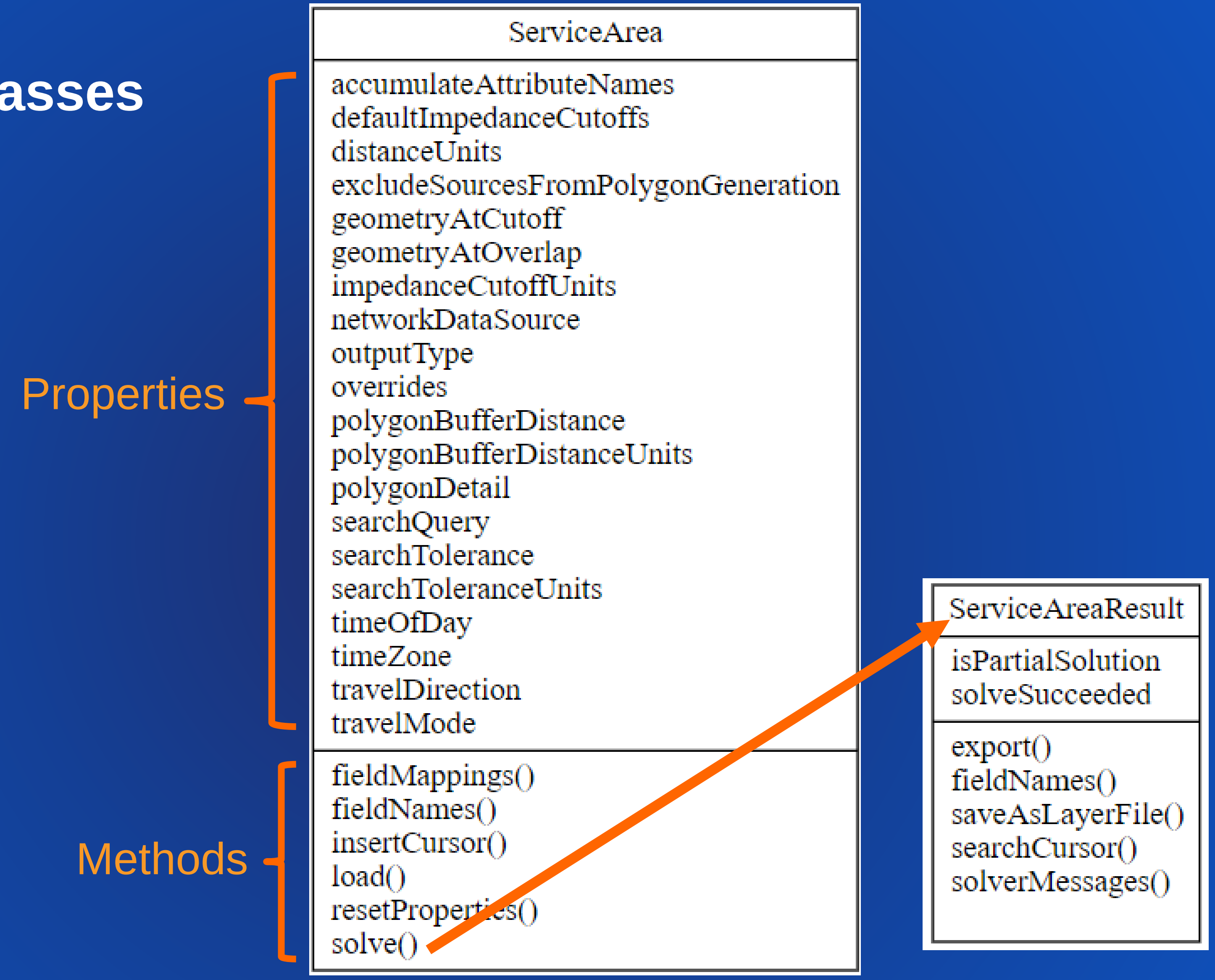
- Working with network analysis layers such as from within Python Window in ArcGIS Pro
- Publishing map service with network analysis capability
- Need to write code that works with ArcGIS Pro version 2.3 or earlier

Network Analysis Workflow

1. **Initialize the analysis**
2. **Set the properties for the analysis**
3. **Load the inputs**
4. **Solve the analysis**
5. **Work with the results**

Common to all the network analyses

Analysis (Solver) Classes



ServiceArea

accumulateAttributeNames
defaultImpedanceCutoffs
distanceUnits
excludeSourcesFromPolygonGeneration
geometryAtCutoff
geometryAtOverlap
impedanceCutoffUnits
networkDataSource
outputType
overrides
polygonBufferDistance
polygonBufferDistanceUnits
polygonDetail
searchQuery
searchTolerance
searchToleranceUnits
timeOfDay
timeZone
travelDirection
travelMode

fieldMappings()
fieldNames()
insertCursor()
load()
resetProperties()
solve()

ServiceAreaResult

isPartialSolution
solveSucceeded

export()
fieldNames()
saveAsLayerFile()
searchCursor()
solverMessages()

For each solver, work with the solver class and the result class

```
# create new route solver

route = arcpy.nax.Route(nd_layer_name)

# load data

route.load(arcpy.nax.RouteInputDataType.Stops, stops_path)

# solve route

result = route.solve()
```

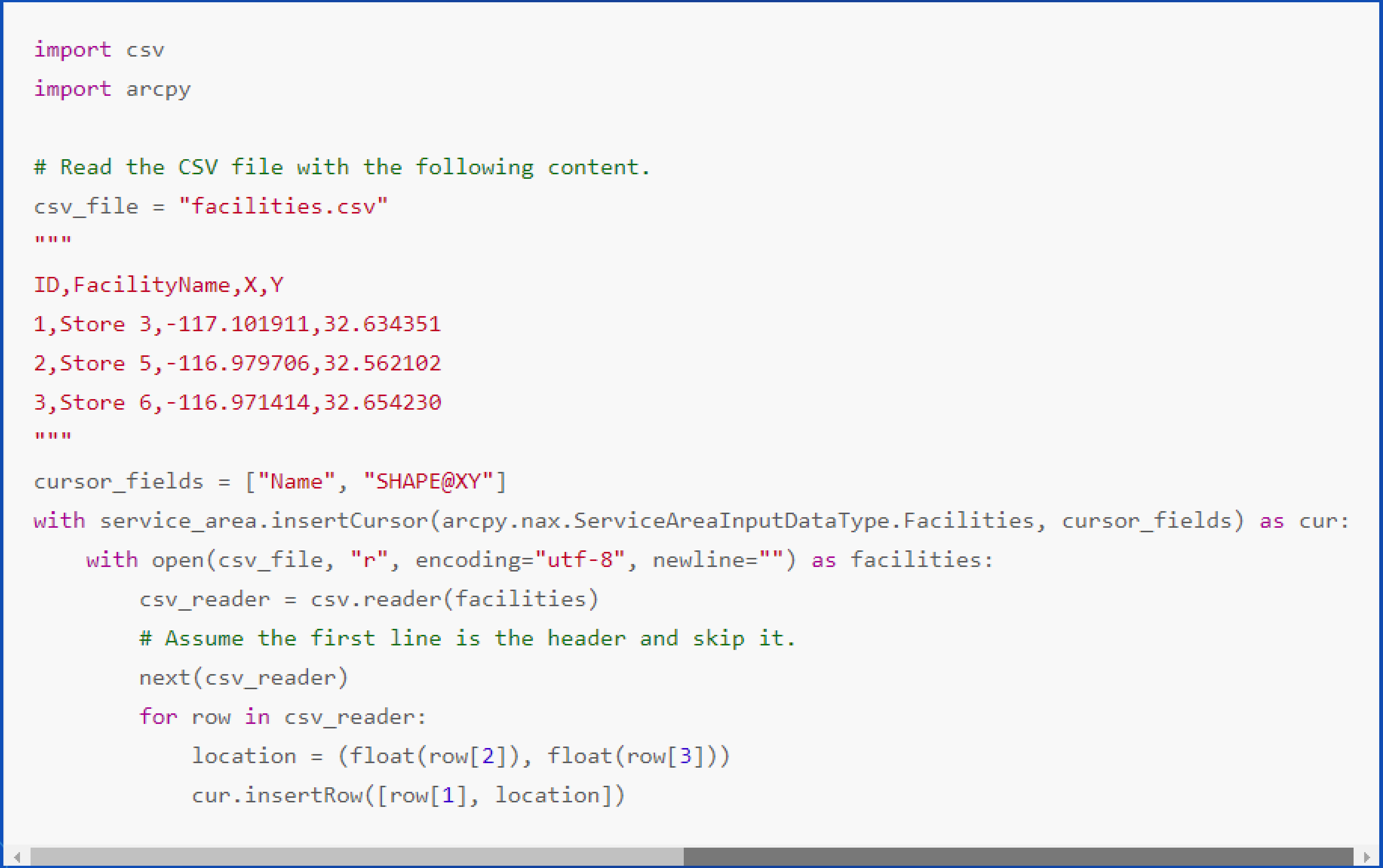
Route and VRP analysis

Jeff Wickstrom

Using Cursors for working with inputs

- You can load the inputs using the insertCursor method on the solver object
- Similar to working with arcpy.da.InsertCursor
- Can be faster in certain workflows since you do not need to first convert your data into a feature class or a table

Example: Using Cursors for working with inputs

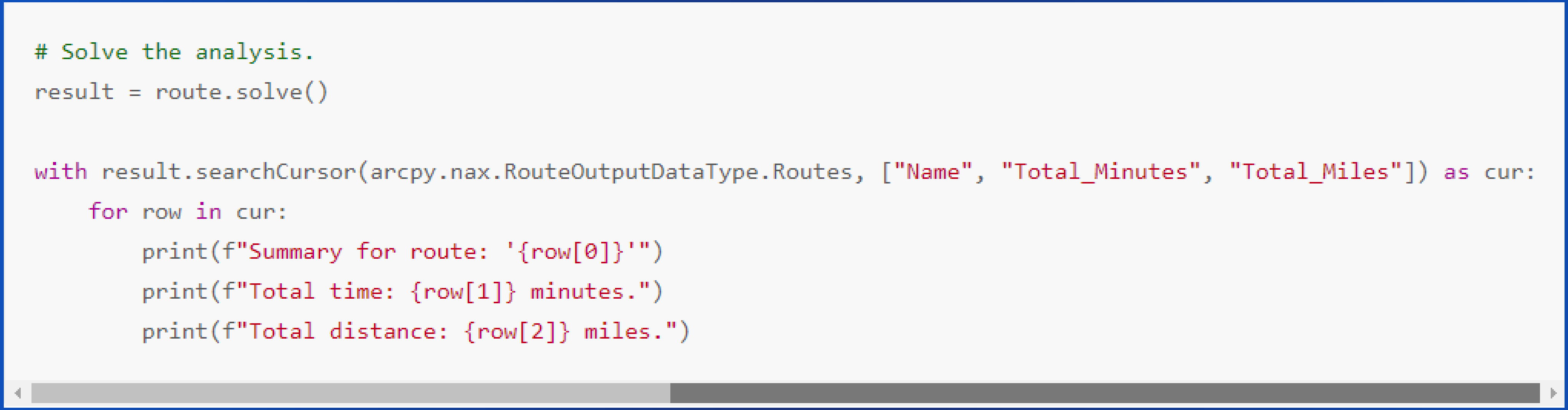


```
import csv
import arcpy

# Read the CSV file with the following content.
csv_file = "facilities.csv"
"""
ID,FacilityName,X,Y
1,Store 3,-117.101911,32.634351
2,Store 5,-116.979706,32.562102
3,Store 6,-116.971414,32.654230
"""
cursor_fields = ["Name", "SHAPE@XY"]
with service_area.insertCursor(arcpy.nax.ServiceAreaInputDataType.Facilities, cursor_fields) as cur:
    with open(csv_file, "r", encoding="utf-8", newline="") as facilities:
        csv_reader = csv.reader(facilities)
        # Assume the first line is the header and skip it.
        next(csv_reader)
        for row in csv_reader:
            location = (float(row[2]), float(row[3]))
            cur.insertRow([row[1], location])
```

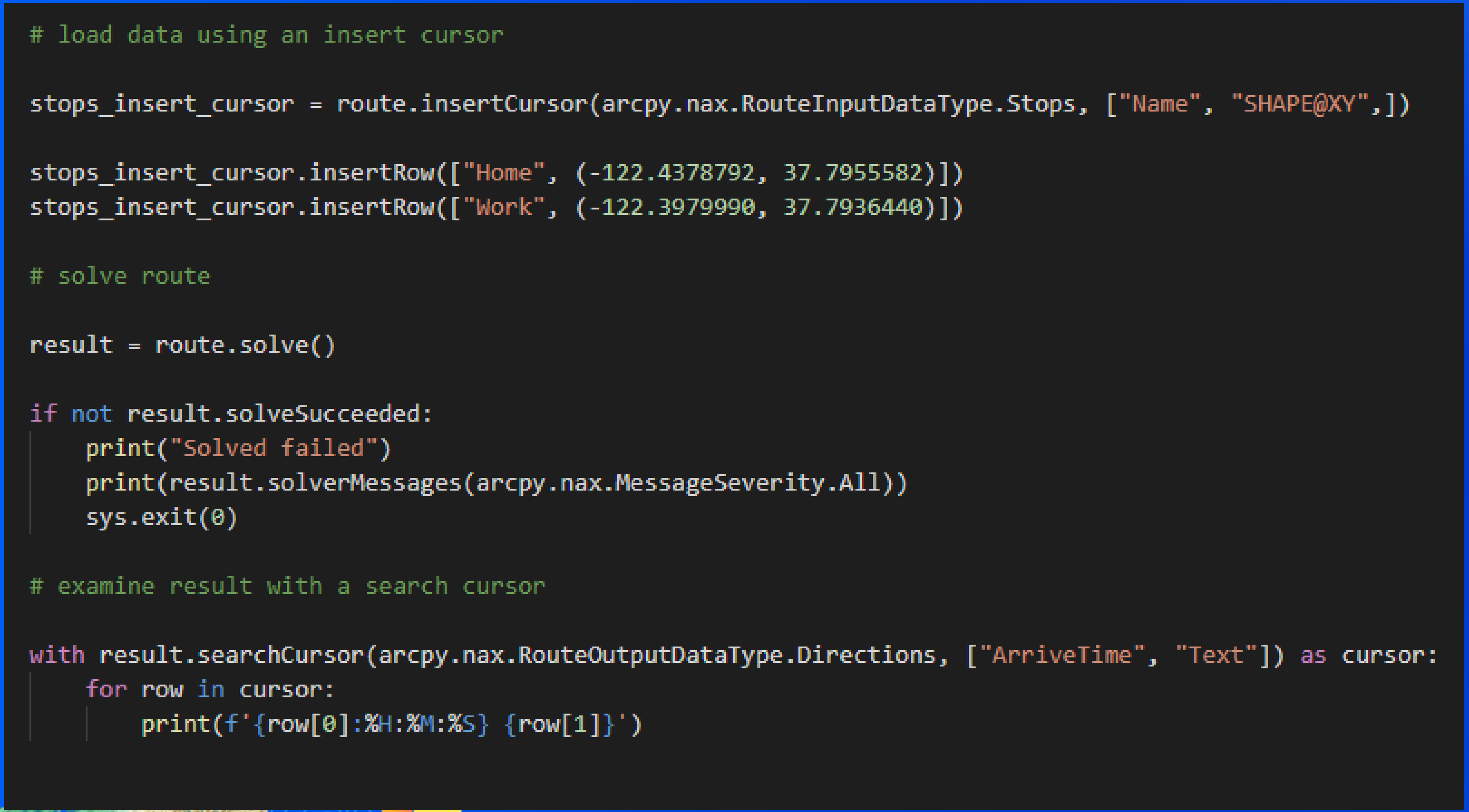
Using Cursors for working with outputs

- You can export the outputs using the searchCursor method on the result object
- Similar to working with arcpy.da.SearchCursor
- Can be faster in certain workflows since you do not need to first export the outputs to a feature class or a table.

Example: Using Cursors for working with outputs



```python
# Solve the analysis.
result = route.solve()

with result.searchCursor(arcpy.nax.RouteOutputDataType.Routes, ["Name", "Total_Minutes", "Total_Miles"]) as cur:
    for row in cur:
        print(f"Summary for route: '{row[0]}'")
        print(f"Total time: {row[1]} minutes.")
        print(f"Total distance: {row[2]} miles.")
```



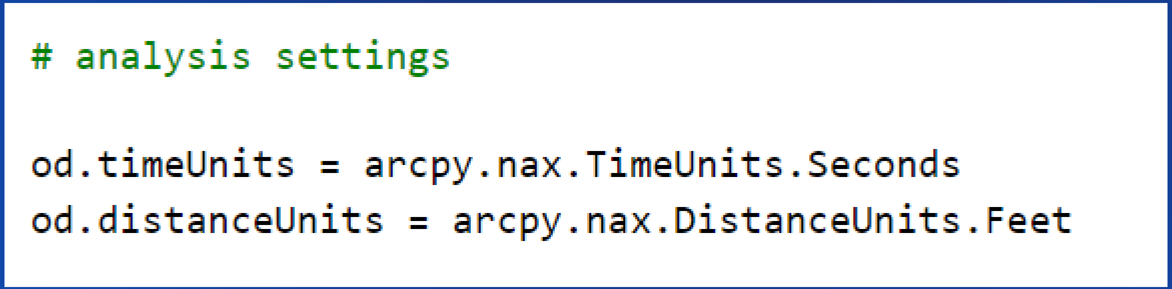
```python
# load data using an insert cursor

stops_insert_cursor = route.insertCursor(arcpy.nax.RouteInputDataType.Stops, ["Name", "SHAPE@XY",])

stops_insert_cursor.insertRow(["Home", (-122.4378792, 37.7955582)])
stops_insert_cursor.insertRow(["Work", (-122.3979990, 37.7936440)])

# solve route

result = route.solve()

if not result.solveSucceeded:
    print("Solved failed")
    print(result.solverMessages(arcpy.nax.MessageSeverity.All))
    sys.exit(0)

# examine result with a search cursor

with result.searchCursor(arcpy.nax.RouteOutputDataType.Directions, ["ArriveTime", "Text"]) as cursor:
    for row in cursor:
        print(f'{row[0]:%H:%M:%S} {row[1]}')
```

Using Cursors with Route and OD Cost Matrix analysis

Jeff Wickstrom

Tips and Tricks: Units for the analysis

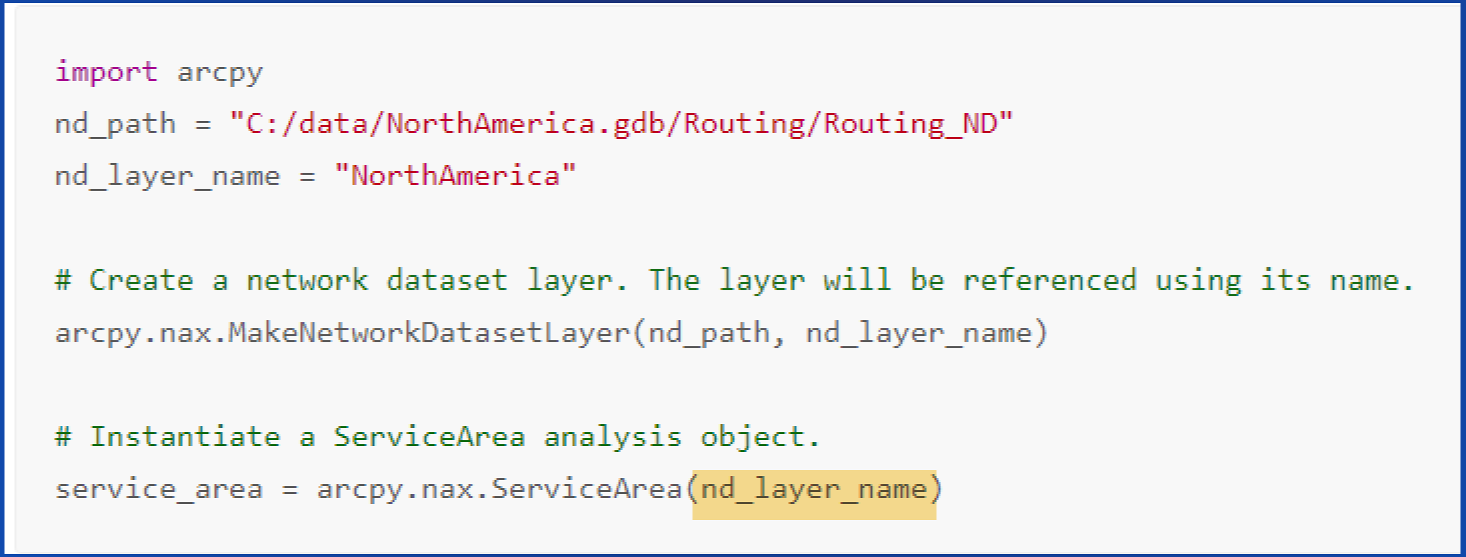
- As part of the analysis settings, you can set the timeUnits and distanceUnits properties
- These can be different than the units of the cost attributes on your network dataset
- The travel costs in the output are always returned in the units you specify.



```
# analysis settings

od.timeUnits = arcpy.nax.TimeUnits.Seconds
od.distanceUnits = arcpy.nax.DistanceUnits.Feet
```

Tips and Tricks: Use the network dataset layer

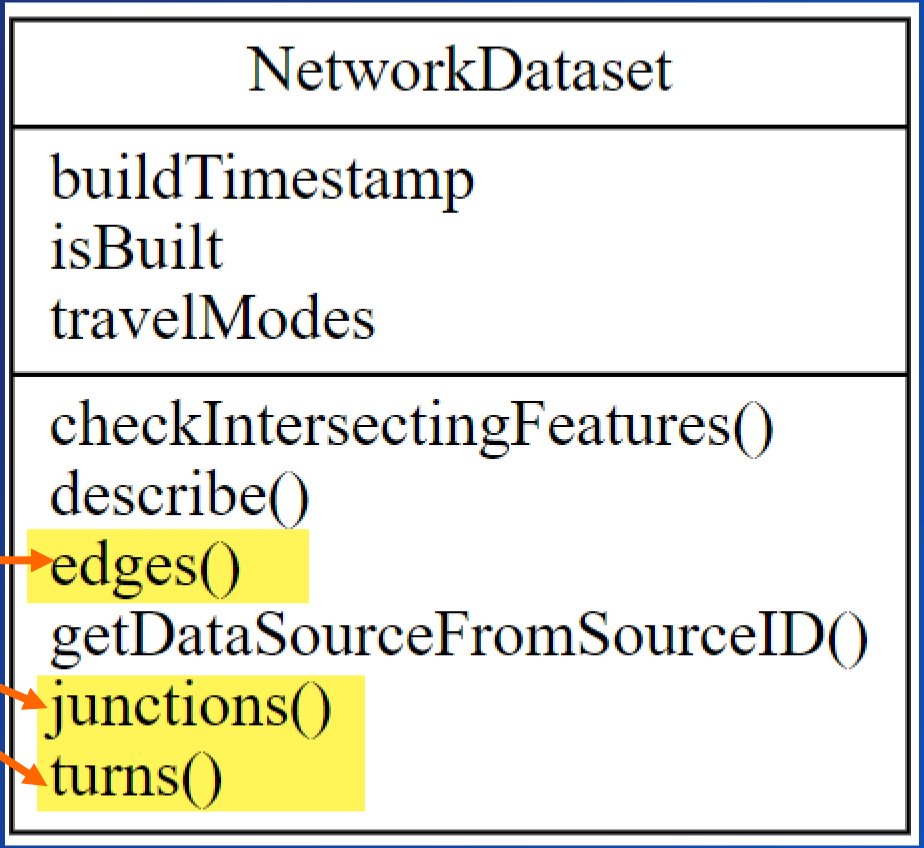
- For best performance, do not use the catalog path of the network dataset when initializing the analysis
- Create a network dataset layer and use it’s name



```python
import arcpy
nd_path = "C:/data/NorthAmerica.gdb/Routing/Routing_ND"
nd_layer_name = "NorthAmerica"

# Create a network dataset layer. The layer will be referenced using its name.
arcpy.nax.MakeNetworkDatasetLayer(nd_path, nd_layer_name)

# Instantiate a ServiceArea analysis object.
service_area = arcpy.nax.ServiceArea(nd_layer_name)
```

Accessing Network Dataset Connectivity

- arcpy.nax.NetworkDataset allows you to work with connectivity and attribute information from a network dataset



Return cursors to iterate over the elements

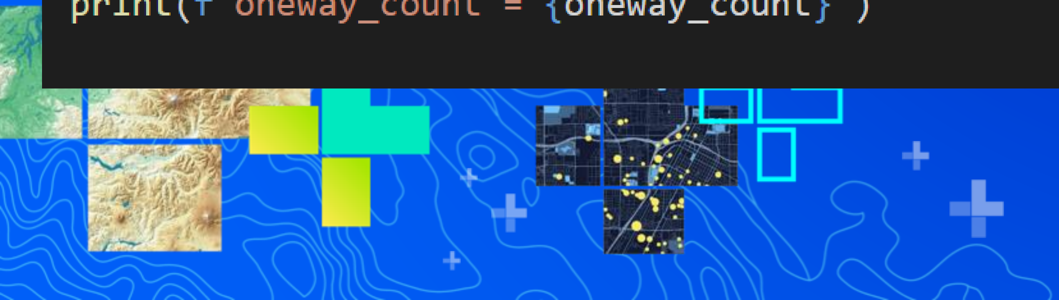
Using Network Dataset with other software packages

- arcpy.nax.NetworkDataset can be used to export the network dataset for use with 3rd Party libraries such as networkx Python package
- Can use algorithms not available in Network Analyst but available in other libraries
- Beware of the network data model differences between Network Analyst and other libraries

```python
# create network dataset object

network_dataset = arcpy.nax.NetworkDataset(nd_layer_name)

# sum up the TravelTime attribute values for all edges

total_travel_time = 0
edge_count = 0
oneway_count = 0

for edge in network_dataset.edges([], ["TravelTime", "Oneway"]):
    total_travel_time += edge[0]
    edge_count += 1
    if edge[1]:
        oneway_count += 1

print(f"total_travel_time = {total_travel_time}")
print(f"edge_count = {edge_count}")
print(f"oneway_count = {oneway_count}")
```

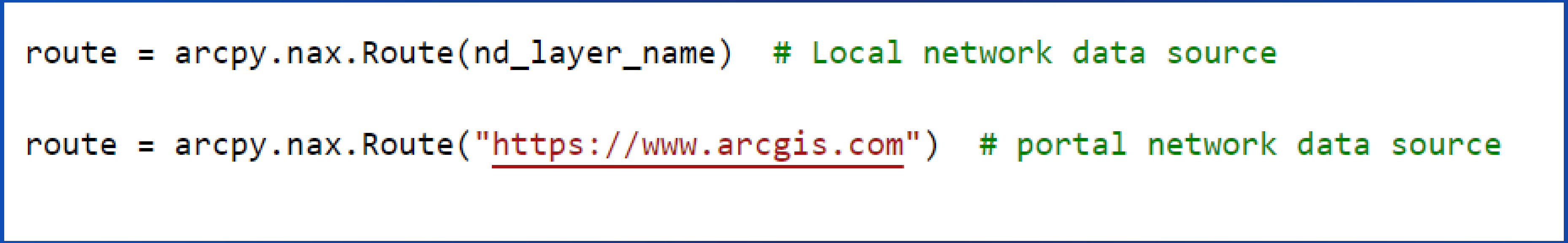
Summarize a network dataset and Find disconnected subgraphs

Jeff Wickstrom



What’s next?

- Allow working with network analysis web services
  - Initialize the solver using the URL to your portal



```
route = arcpy.nax.Route(nd_layer_name)  # Local network data source

route = arcpy.nax.Route("https://www.arcgis.com")  # portal network data source
```

- Direct methods to access certain results such as total time and distance from a route

- You can suggest improvements on https://ideas.arcgis.com

Resources

- arcpy.nax help
- Help also includes many more code samples

Final Thoughts…

- arcpy.nax allows you to perform network analysis using Python in ArcGIS Pro 2.4 and later

- Benefits of using arcpy.nax
  - Significantly faster analysis time as all data is stored in memory and not in file geodatabase
  - Provides a modern and easy to use Python API (aka Pythonic)

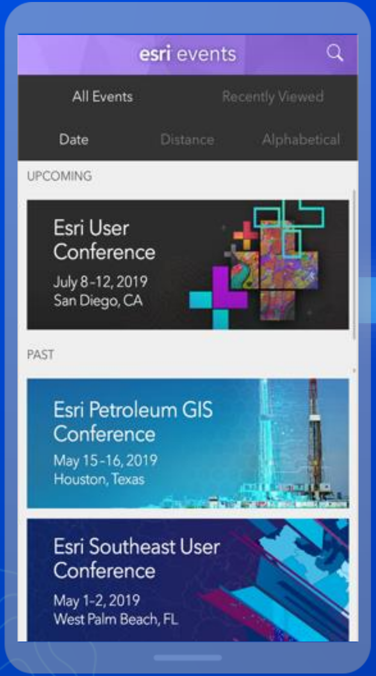
- ArcGIS Online network analysis services use arcpy.nax

Network Analyst Presentations

	Tuesday July 9	**Wednesday** July 10	**Thursday** July 11
8A	Network Analyst: An Introduction - Room 30B; Network Analyst: Using the Python API - Room 16B	Network Analyst: Automating Workflows with Geoprocessing - Room 30D	
9A			
10A		Network Analyst: Creating High Density Routes with the VRP Solver - Demo Theater 10	Network Analyst: Automating Workflows with Geoprocesssing - Room 30A
11A	Publish Your Own Network Analysis Services with ArcGIS Enterprise - Demo Theater 10		Building Routing Applications with ArcGIS Online - Demo Theater 07
12P			
1P	Network Analyst: Creating Network Datasets - Room 30B; Navigator for ArcGIS: Connecting to Preplanned Routes - Room 30A	ArcGIS Enterprise: Deep Dive into Geoprocessing Services - Room 03; Network Analyst: Solving Transportation Analysis Problems with Public Transit Data - Room 30B	ArcGIS Online: Routing and Network Analysis using Web Services - Room 33C
2P	Network Analyst: Optimize Your Fleet of Vehicles with the VRP Solver - Room 30B; Navigator for ArcGIS: An Introduction - Room 15A	Network Analyst: Optimize Your Fleet of Vehicles with the VRP Solver - Room 30B; Navigator for ArcGIS: Connecting to Preplanned Routes - Demo Theater 09	Network Analyst: Using the Python API - Room 30A
3P			
4P		Network Analyst: Creating Network Datasets - Room 30B	Network Analyst: An Introduction - Room 16A
5P			

Please Share Your Feedback in the App

Download the Esri Events app and find your event


esri events

All Events · Recently Viewed

Date · Distance · Alphabetical

UPCOMING

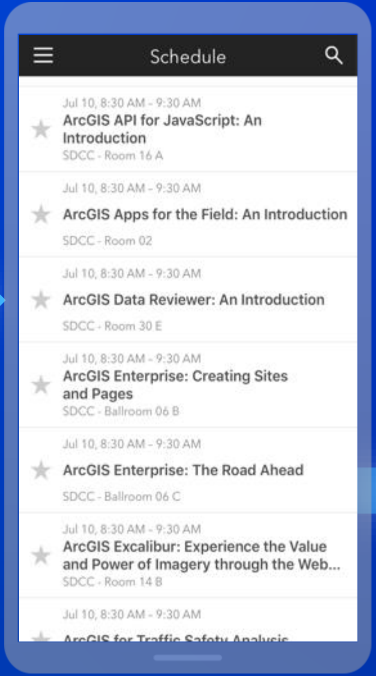
Esri User Conference
July 8-12, 2019
San Diego, CA

PAST

Esri Petroleum GIS Conference
May 15-16, 2019
Houston, Texas

Esri Southeast User Conference
May 1-2, 2019
West Palm Beach, FL

Select the session you attended


Schedule

Jul 10, 8:30 AM - 9:30 AM
ArcGIS API for JavaScript: An Introduction
SDCC - Room 16 A

Jul 10, 8:30 AM - 9:30 AM
ArcGIS Apps for the Field: An Introduction
SDCC - Room 02

Jul 10, 8:30 AM - 9:30 AM
ArcGIS Data Reviewer: An Introduction
SDCC - Room 30 E

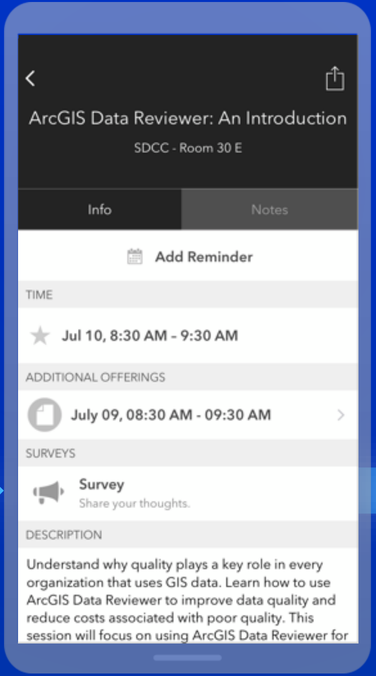
Jul 10, 8:30 AM - 9:30 AM
ArcGIS Enterprise: Creating Sites and Pages
SDCC - Ballroom 06 B

Jul 10, 8:30 AM - 9:30 AM
ArcGIS Enterprise: The Road Ahead
SDCC - Ballroom 06 C

Jul 10, 8:30 AM - 9:30 AM
ArcGIS Excalibur: Experience the Value and Power of Imagery through the Web...
SDCC - Room 14 B

Jul 10, 8:30 AM - 9:30 AM
ArcGIS for Traffic Safety Analysis

Scroll down to “Survey”


ArcGIS Data Reviewer: An Introduction
SDCC - Room 30 E

Info · Notes

Add Reminder

TIME

Jul 10, 8:30 AM - 9:30 AM

ADDITIONAL OFFERINGS

July 09, 08:30 AM - 09:30 AM

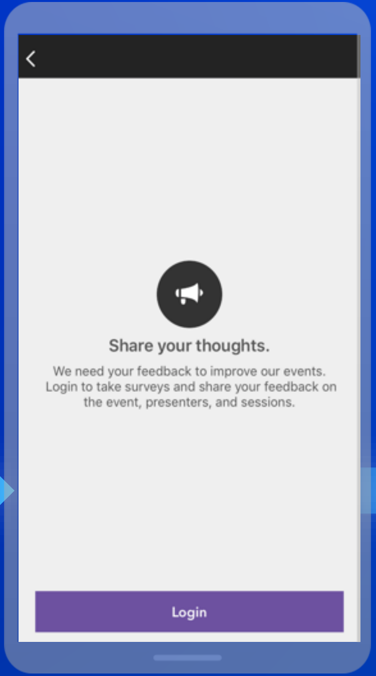
SURVEYS

Survey
Share your thoughts.

DESCRIPTION

Understand why quality plays a key role in every organization that uses GIS data. Learn how to use ArcGIS Data Reviewer to improve data quality and reduce costs associated with poor quality. This session will focus on using ArcGIS Data Reviewer for

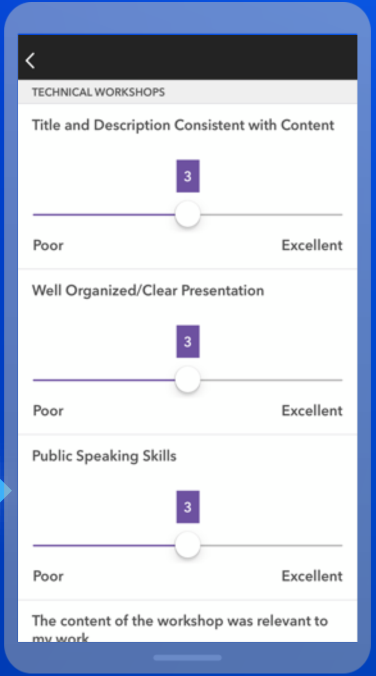
Log in to access the survey


Share your thoughts.

We need your feedback to improve our events. Login to take surveys and share your feedback on the event, presenters, and sessions.

Login

Complete the survey and select “Submit”


TECHNICAL WORKSHOPS

Title and Description Consistent with Content
3
Poor · Excellent

Well Organized/Clear Presentation
3
Poor · Excellent

Public Speaking Skills
3
Poor · Excellent

The content of the workshop was relevant to my work

Questions

- Slides and code samples for this workshop on Using the Python API are available at

http://esriurl.com/uc19nax