

# CYCLING THROUGH THE PHILADELPHIA INDEGO BIKE-SHARE API



Eric O'Callaghan



ericoc



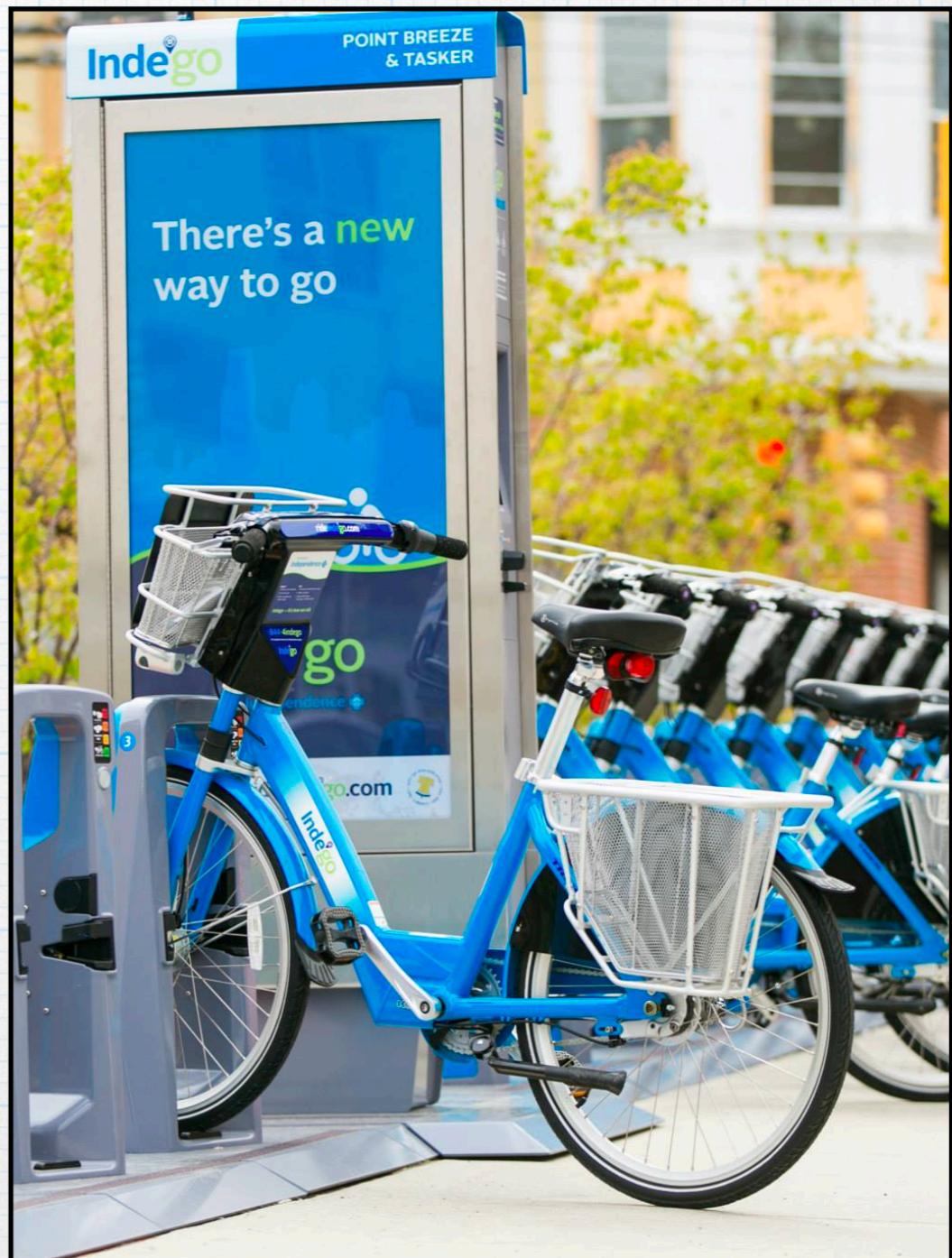
eric@ericoc.com



ericohsee

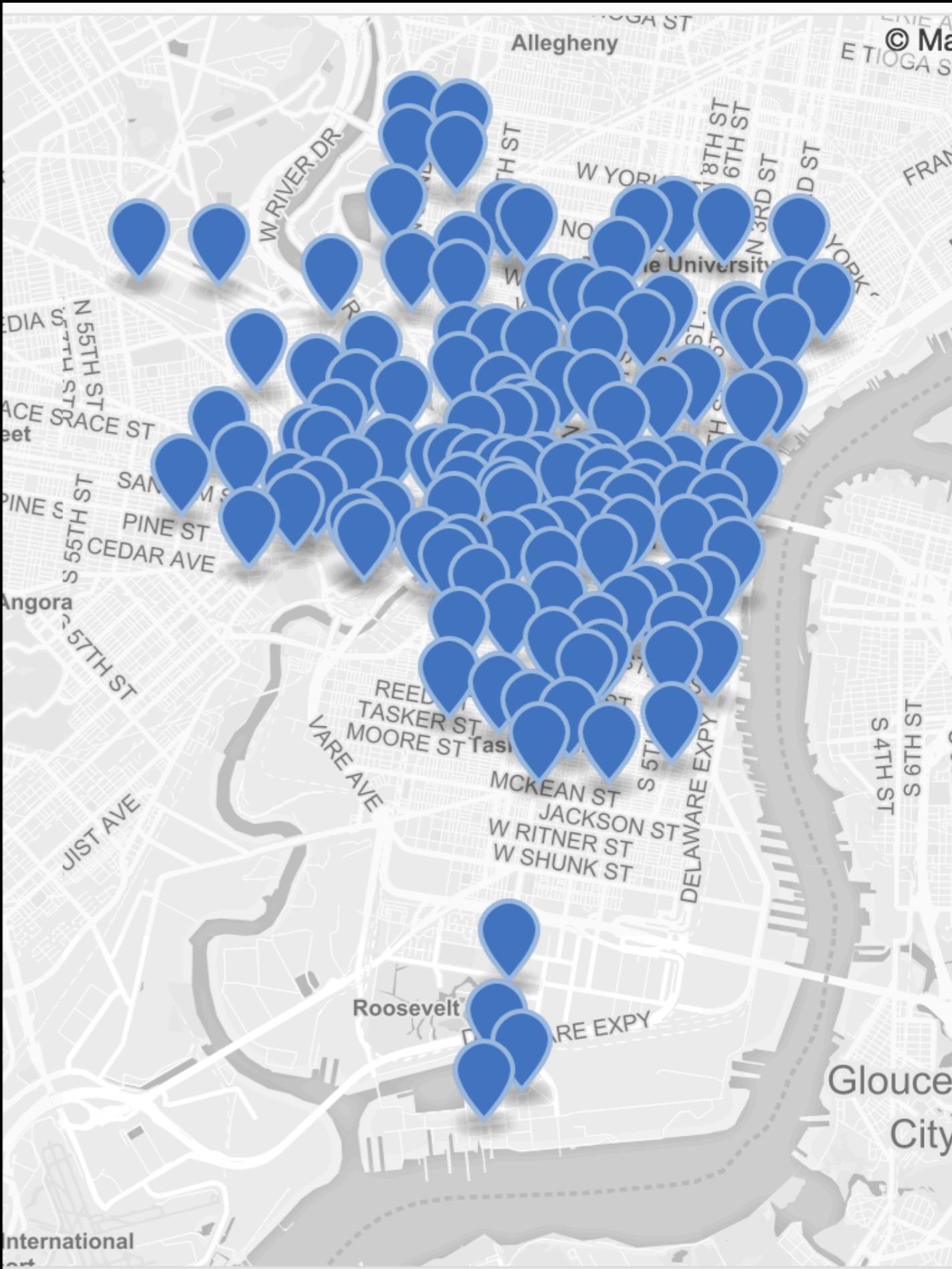
# What Bike-Share? What API?

- \* You know, those big blue stations with the signs, docks, and solar panels that are full of BICYCLES!
- \* There are about 120 of those stations all across the city of Philadelphia
- \* There is a (Geo)JSON API that returns the number of available bicycles, empty docks, and total docks at each station



# GeoJSON?

- \* Yes, GeoJSON! To put it simply, GeoJSON is JSON that also includes simple geographical features
- \* GPS coordinates (latitude and longitude) are included with each bike-share station in the API
- \* GitHub actually handles GeoJSON in a neat way - by showing a map!



# Background on the API

- \* The primary Indego bike-share station API is available here:
  - \* <https://www.rideindego.com/stations/json/>
- \* The API is simply a GeoJSON file that lists each of the stations and provides some (real-time?) data on each one
- \* In my opinion, the most interesting things that are provided about each station by the API are:
  - \* "kioskId" (station identification number; currently 3000 to 3200)
  - \* Location (GPS coordinates, street address, and zip/postal code)
  - \* Available number of bicycles and docks
  - \* Total number of docks (occasionally not equal to available bikes+docks!)

# Example

- \* This is an example showing three (3) stations
- \* This is not even all of the data available for each station; just the values that interest me most
- \* Each station has about 30 API properties available, but only about half of those are truly useful

```
{  
  "name": "Municipal Services Building Plaza",  
  "latitude": 39.95378,  
  "longitude": -75.16374,  
  "addressStreet": "1401 John F. Kennedy Blvd.",  
  "addressZipCode": "19102",  
  "bikesAvailable": 5,  
  "docksAvailable": 25,  
  "totalDocks": 30  
}  
  
{  
  "name": "Welcome Park, NPS",  
  "latitude": 39.94733,  
  "longitude": -75.14403,  
  "addressStreet": "191 S. 2nd St.",  
  "addressZipCode": "19106",  
  "bikesAvailable": 12,  
  "docksAvailable": 1,  
  "totalDocks": 13  
}  
  
{  
  "name": "40th & Spruce",  
  "latitude": 39.9522,  
  "longitude": -75.20311,  
  "addressStreet": "246 S. 40th St.",  
  "addressZipCode": "19104",  
  "bikesAvailable": 3,  
  "docksAvailable": 14,  
  "totalDocks": 17  
}
```

# Using the API

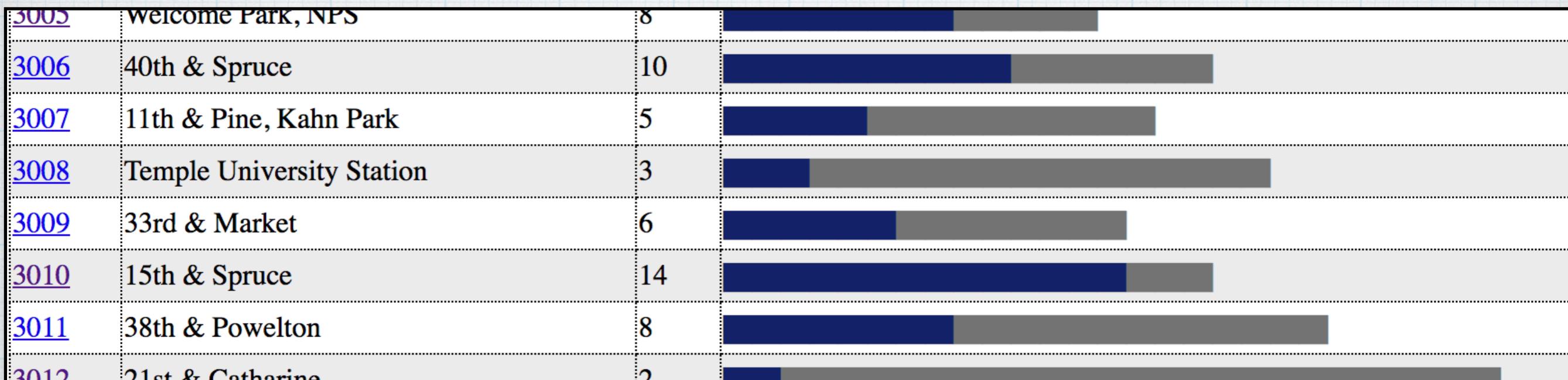
- \* The primary Indego bike-share API **GeoJSON** file lists all stations
- \* It is not REST-ful and does not offer any way to search for specific stations
- \* I created both a PHP and a Python library for accessing the API
  - \* My libraries sort of let you "search" for stations
  - \* I would love to see expanded documentation of the API and a proper search feature

# Searching Stations?

- \* I tried to build both my PHP and Python Indego API libraries identically. They accept a single search parameter:
  - \* No search parameter? Return all stations
  - \* Digits as search parameter?
    - \* Four (4) digits? Station with that unique "kioskId"
    - \* Five (5) digits? Stations with that zip/postal code
  - \* Any other search parameter? Regular expression match on:
    - \* Station name
    - \* Address

# Graphing Stations (Part 1)

- \* This was my first attempt at using the API
- \* It is a basic website with a table of colorful blocks showing bicycles and empty docks at each station



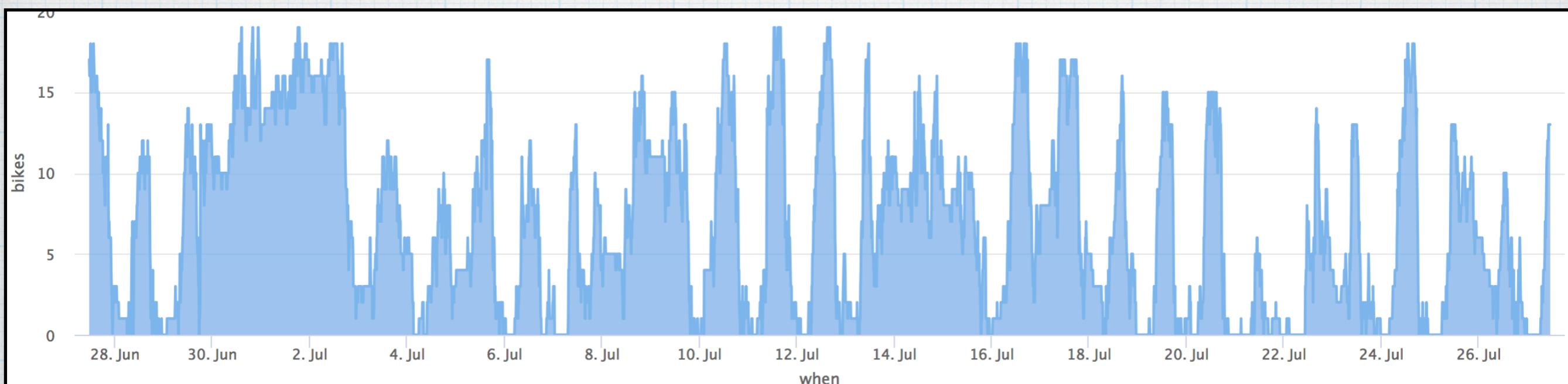
# Graphing Stations (Part 2)

- \* This is an example of a Python command-line program that is using my API library
- \* It shows another attempt at colorful graphs of each station

3004	Municipal Services Building Plaza	#####====
3005	Welcome Park, NPS	#####---
3006	40th & Spruce	#####---
3007	11th & Pine, Kahn Park	=====
3008	Temple University Station	##=====
3009	33rd & Market	#####---
3010	15th & Spruce	#####====
3011	38th & Powelton	#####---
3012	21st & Catharine	##=====
3013	6th & Fairmount	#####---

# Graphing Stations (Part 3)

- \* I have been recording the number of bicycles and docks at each station for years
- \* Creating historical graphs (via Highcharts) reveals commuting patterns



# Conclusion

- \* I really like the idea of visualizing the availability of bicycles and docks
- \* In my opinion, it looks especially cool during commuting hours
- \* This is just one idea of what to use the Philadelphia Indego Bike-Share API for
- \* I love seeing data that is straight from the real world - as people move throughout the city!

# More Information

## \* Me:

- \* <https://indego.ericoc.com/> - Page with block-type charts of each station (and historical graphs made using Highcharts). Clearly, I'm not a designer!
- \* <https://www.highcharts.com/blog/post/250-tracking-bike-share-usage-in-philadelphia/> - A blog post that I wrote for Highcharts.com, after originally using it for historical data via PHP

## \* RideIndego:

- \* [https://gbfs.bcycle.com/bcycle\\_indego/gbfs.json](https://gbfs.bcycle.com/bcycle_indego/gbfs.json) - A more minimal API for the Philadelphia bike-share that uses a different standard (GBFS) than the flat GeoJSON file
- \* <https://www.opendataphilly.org/organization/city-of-philadelphia?q=indego> - OpenDataPhilly list with additional details about the API(s)

## \* Randal Olson:

- \* These two blog posts were my first inspiration for visualizing bike-share usage:
  - \* <http://www.randalolson.com/2015/07/18/visualizing-indego-bike-share-usage-patterns-in-philadelphia/>
  - \* <http://www.randalolson.com/2015/09/05/visualizing-indego-bike-share-usage-patterns-in-philadelphia-part-2/>