

Insights for ArcGIS for Developers

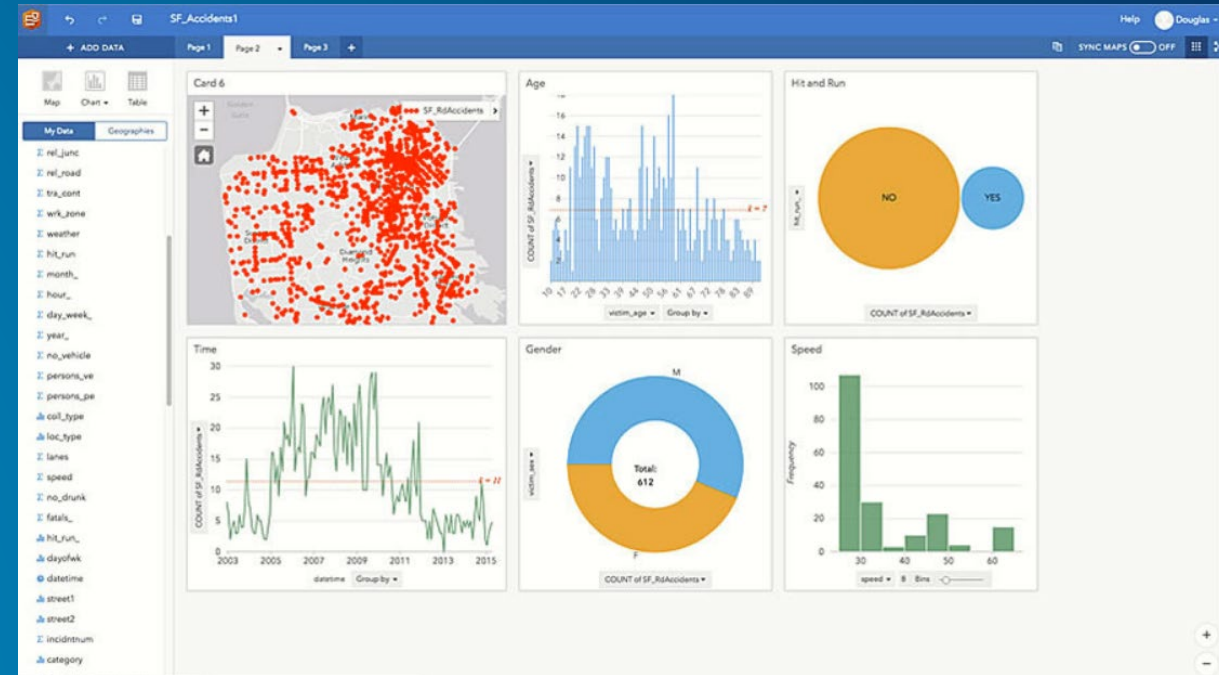
Sébastien Szollosi

ESRI EUROPEAN DEVELOPER SUMMIT



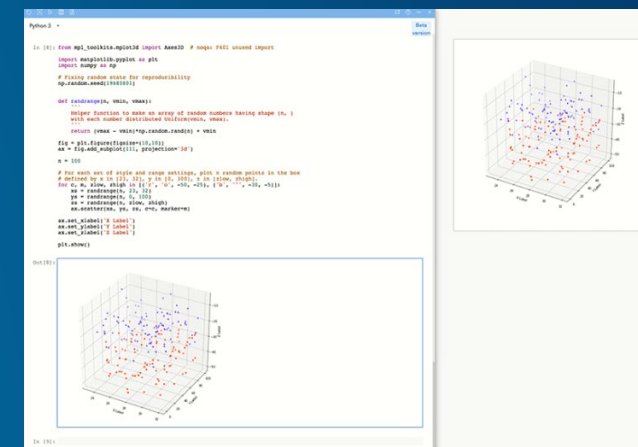
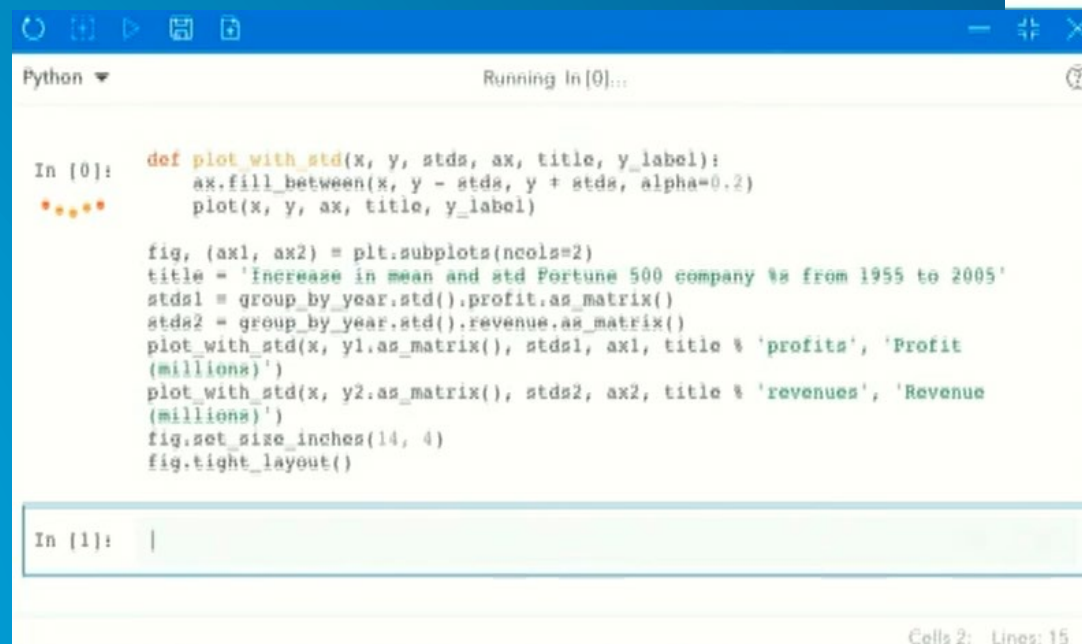
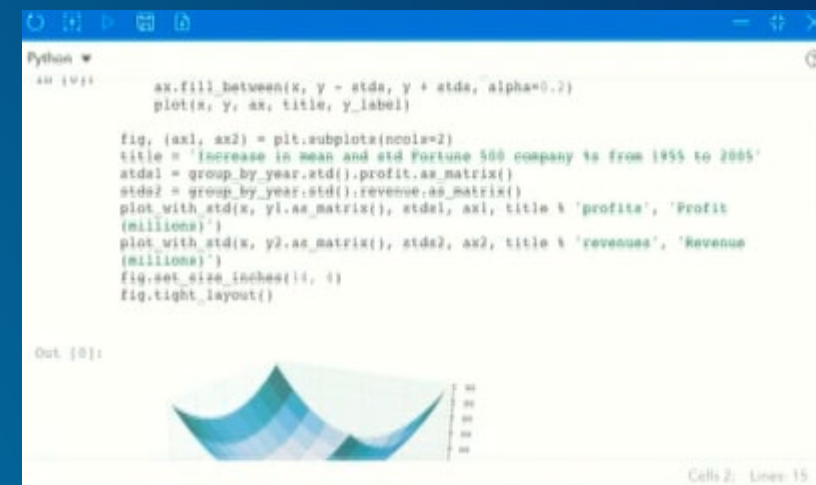
What is ArcGIS Insights ?

- Insights for ArcGIS was the previous name
- Explore spatial and non-spatial data,
- Answer questions you didn't know to ask
- Deliver powerful results
- Share them



Python and R with Insights

- **Extend your analysis using Python and/or R**
- **Incorporate visualisations as card**
- **Manage data**



How do you get started ?

- Insights for ArcGIS 3.2 now supports scripting in Python and R
- Allows you to use existing open data science platforms
- Uses a Jupyter Kernel Gateway

Connecting to Python and R

- From an Insights Workbook
- Click Console button on toolbar
- Connect to Jupyter Kernel Gateway
 - <https://insights.esri.com:9999>
 - <wss://insights.esri.com:9999>



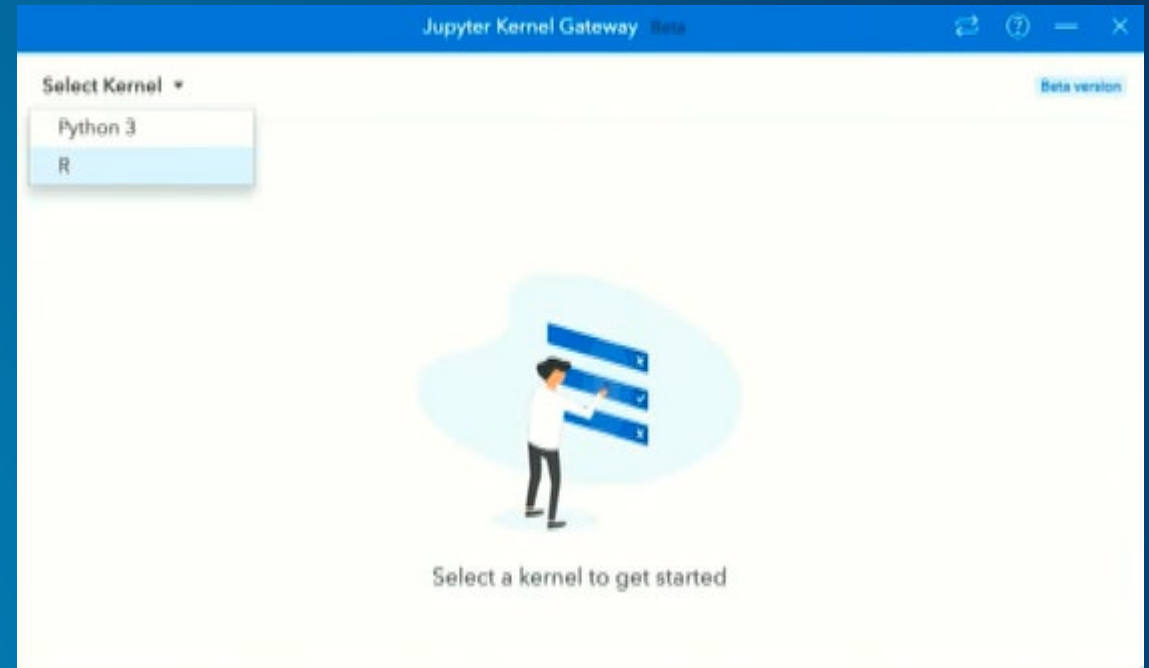
New Jupyter Kernel Gateway connection

• URL

• WEB SOCKET

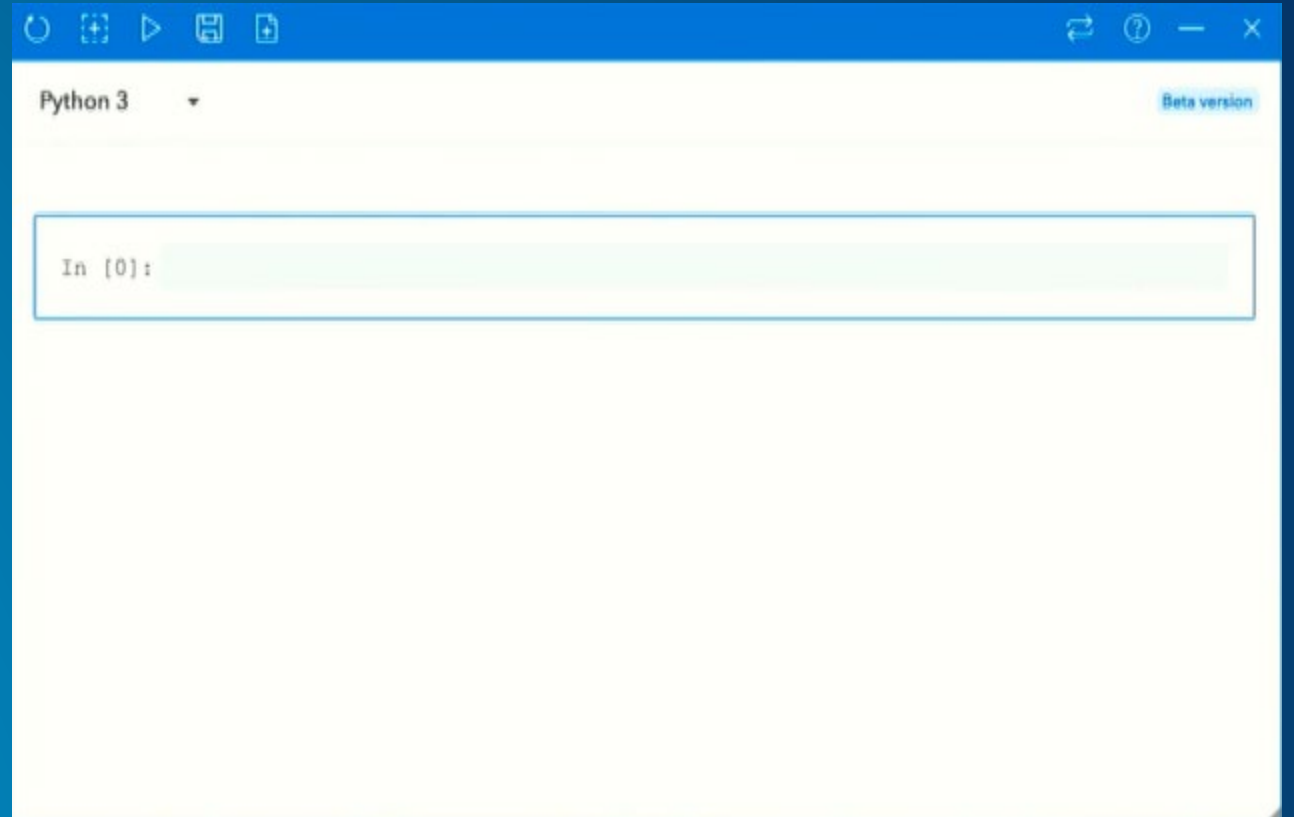
Choose Kernel

- **Select Kernel dropdown**
 - Python
 - R



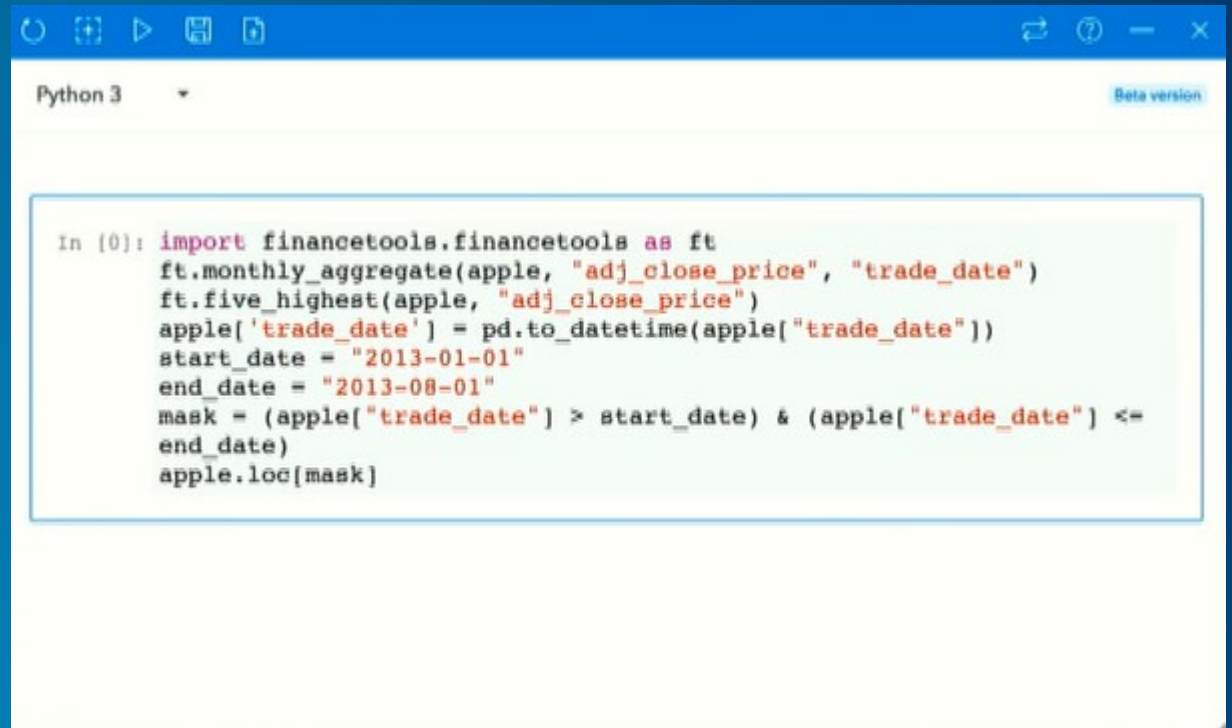
The Console

- Where you interact with the kernel
 - Code editing
 - Imports scripts
 - Intellisense and code completion



Write Code

- Python
- R

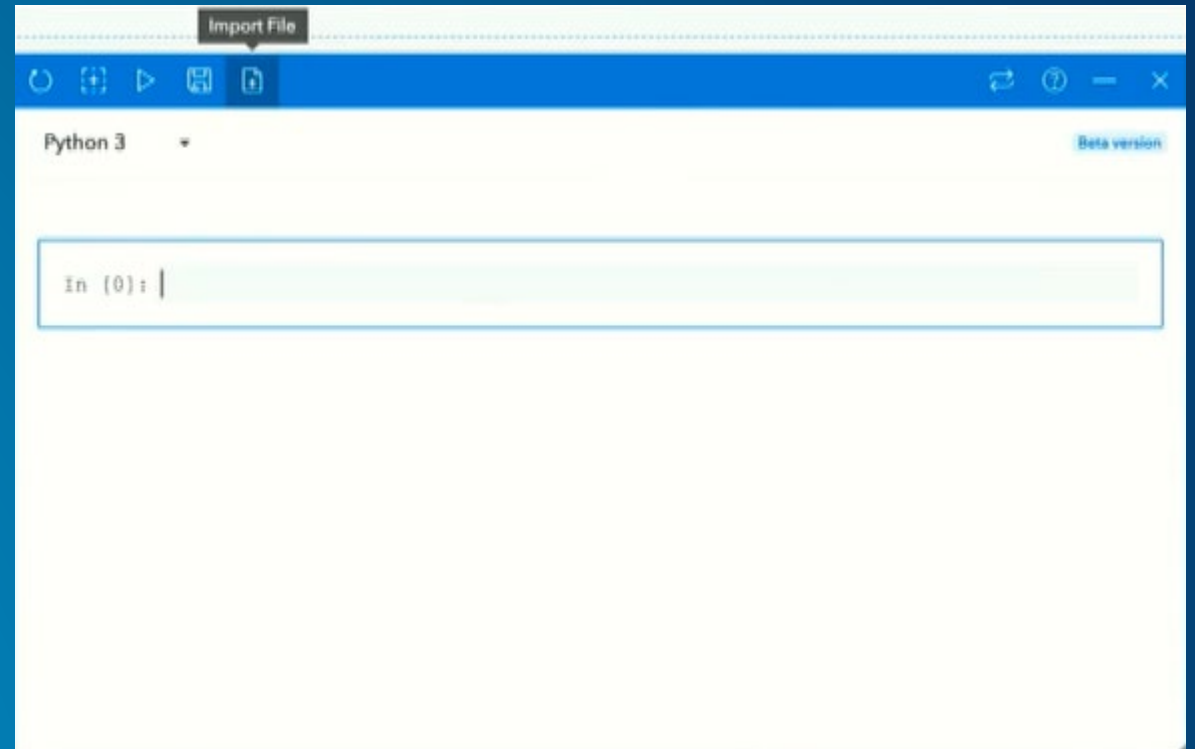


The image shows a Jupyter Notebook window with a blue header bar containing standard icons (refresh, zoom, play, save, and a document icon). Below the header, the text "Python 3" is displayed on the left, and "Beta version" is on the right. The main area of the notebook contains a code cell with the following Python code:

```
In [0]: import financetools.financetools as ft
ft.monthly_aggregate(apple, "adj_close_price", "trade_date")
ft.five_highest(apple, "adj_close_price")
apple['trade_date'] = pd.to_datetime(apple["trade_date"])
start_date = "2013-01-01"
end_date = "2013-08-01"
mask = (apple["trade_date"] > start_date) & (apple["trade_date"] <=
end_date)
apple.loc[mask]
```

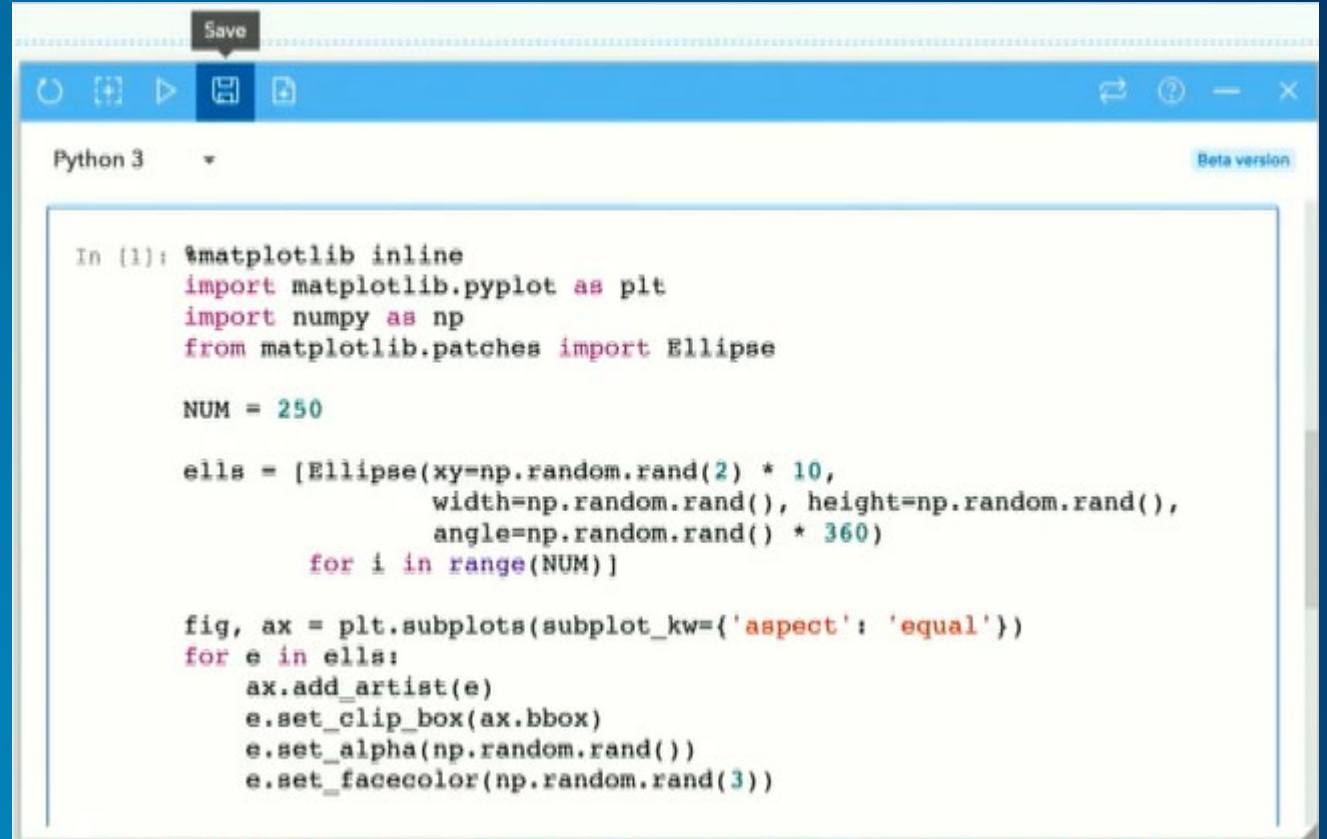

Importing Code

- Import code in single cell
 - .py file input
 - .r file input



Save Work

- Save history
 - ipnyb file
- Save code in single cell
 - .r file output
 - .py file output



The screenshot shows a Jupyter Notebook interface. At the top, there is a toolbar with icons for running, saving, and other actions. A 'Save' button is highlighted. Below the toolbar, the text 'Python 3' and 'Beta version' are visible. The main area contains a code cell with the following Python code:

```
In [1]: %matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.patches import Ellipse

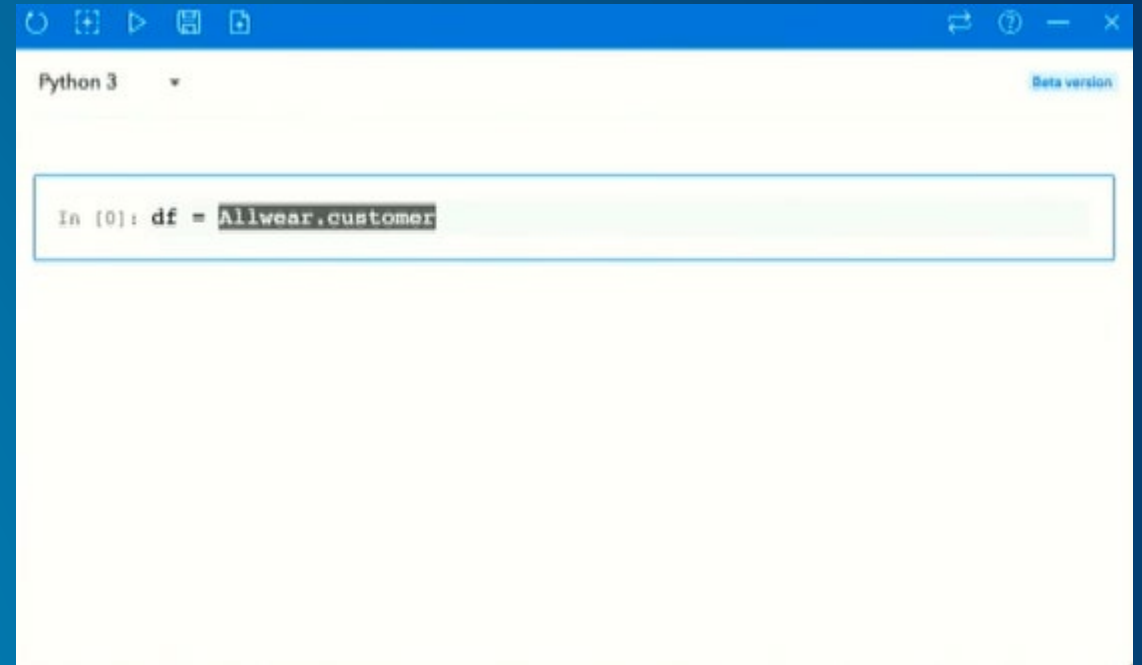
NUM = 250

ells = [Ellipse(xy=np.random.rand(2) * 10,
                width=np.random.rand(), height=np.random.rand(),
                angle=np.random.rand() * 360)
        for i in range(NUM)]

fig, ax = plt.subplots(subplot_kw={'aspect': 'equal'})
for e in ells:
    ax.add_artist(e)
    e.set_clip_box(ax.bbox)
    e.set_alpha(np.random.rand())
    e.set_facecolor(np.random.rand(3))
```

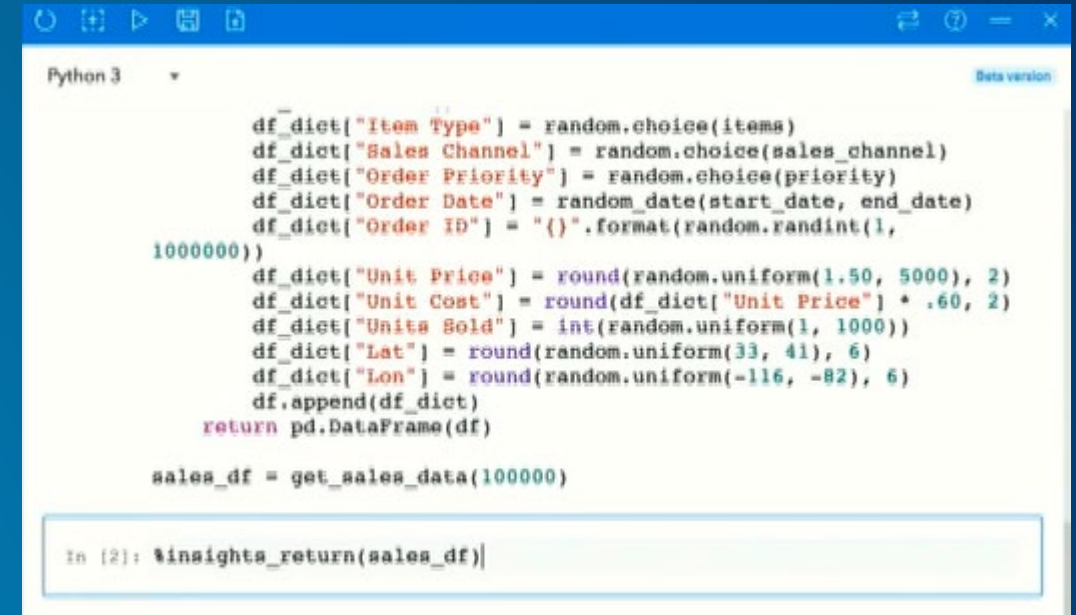
Using Insights Data

- Choose data from Data Pane
- Drag-and-drop data into Console
- Create data frame from dropped data



Data from Console to Data Pane

- Use Insights magic command
 - `%insights_return(data_frame)`
 - Takes R data frame or pandas Data Frame
 - Ctrl + q for shortcut
 - Must be run on single line in single cell



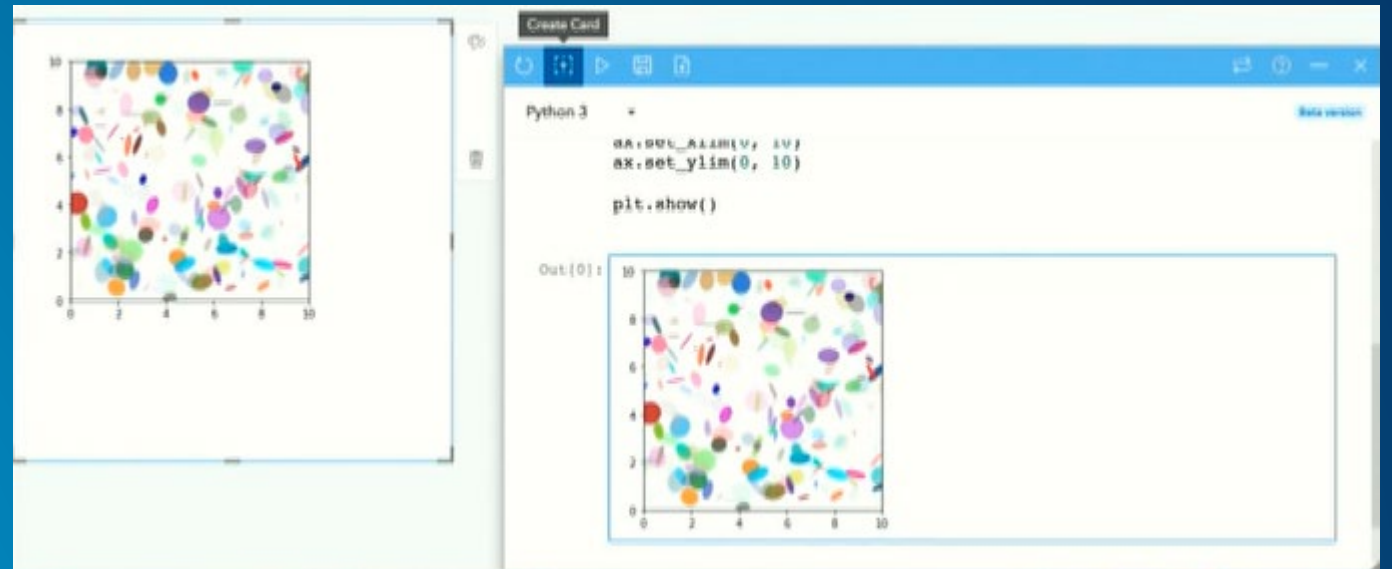
```
Python 3 Beta version  
  
df_dict["Item Type"] = random.choice(items)  
df_dict["Sales Channel"] = random.choice(sales_channel)  
df_dict["Order Priority"] = random.choice(priority)  
df_dict["Order Date"] = random_date(start_date, end_date)  
df_dict["Order ID"] = "{}".format(random.randint(1,  
1000000))  
df_dict["Unit Price"] = round(random.uniform(1.50, 5000), 2)  
df_dict["Unit Cost"] = round(df_dict["Unit Price"] * .60, 2)  
df_dict["Units Sold"] = int(random.uniform(1, 1000))  
df_dict["Lat"] = round(random.uniform(33, 41), 6)  
df_dict["Lon"] = round(random.uniform(-116, -82), 6)  
df.append(df_dict)  
return pd.DataFrame(df)  
  
sales_df = get_sales_data(100000)  
  
In [2]: %insights_return(sales_df)
```

Shortcuts

- **Ctrl + q**
 - Add %insights_return magic command to cell
- **Ctrl + /**
 - comment line
- **Ctrl + Spacebar**
 - intellisense
- **Shift + Enter /return**
 - execute cell
- **Shift + Up Arrow or Down Arrow**
 - access history of executed cells

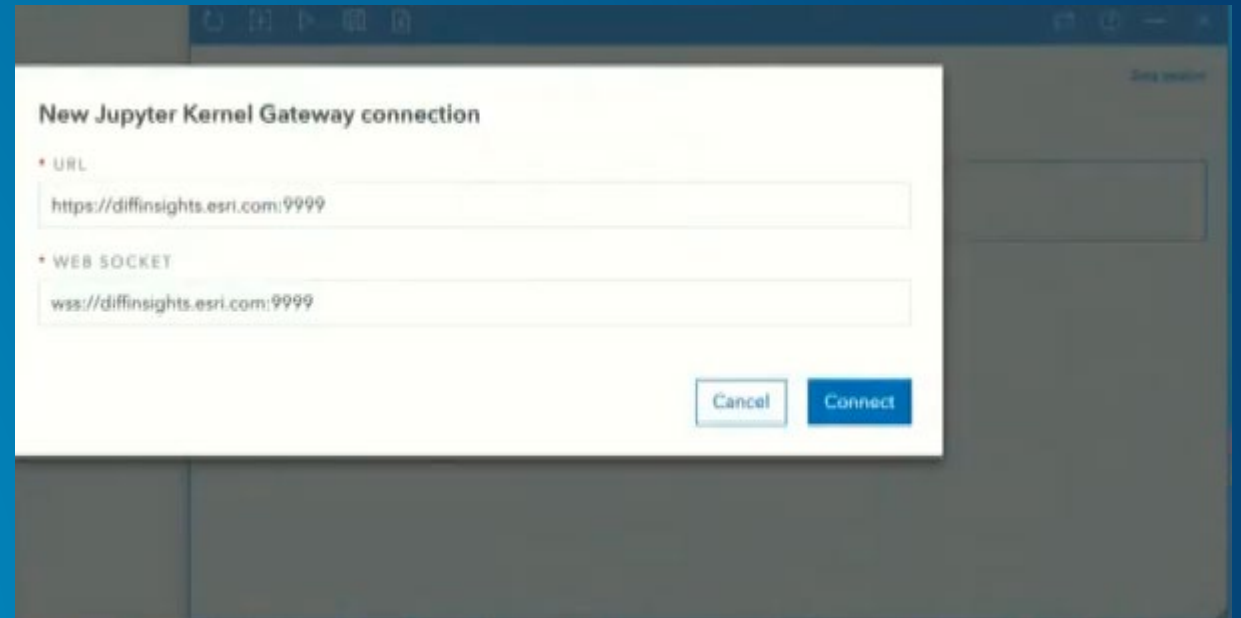
Create Custom cards

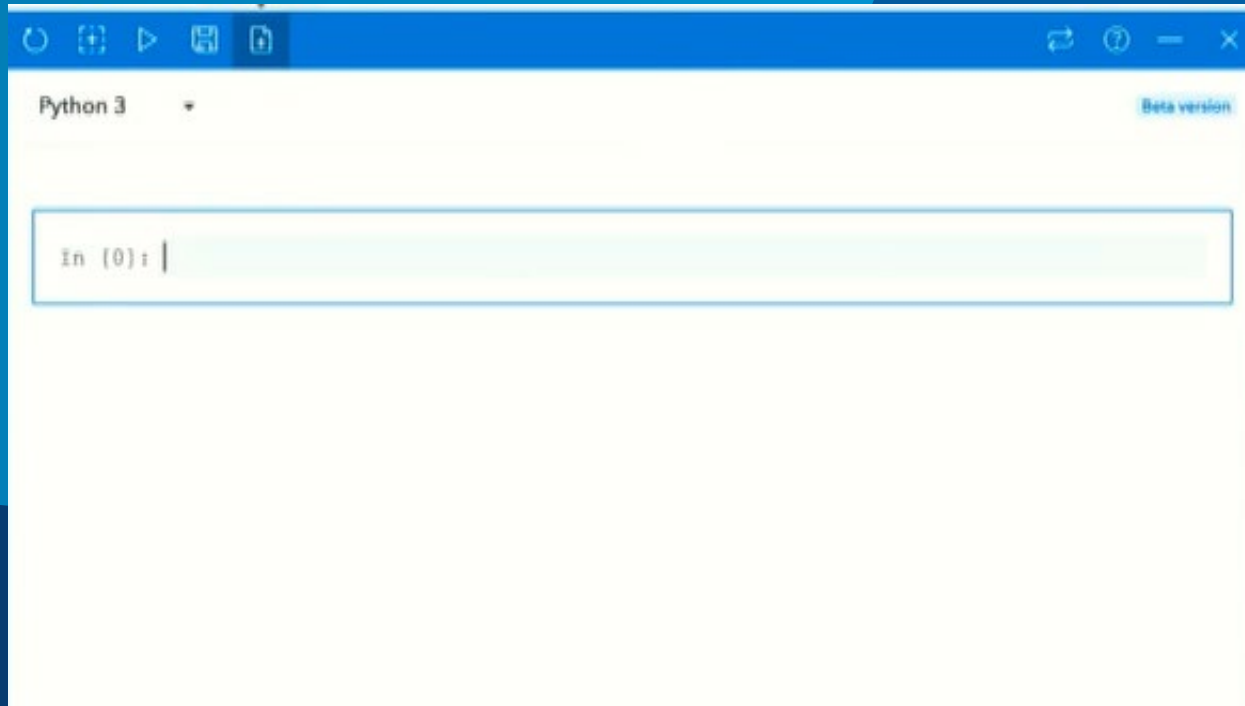
- Custom visualisations
- Select output result cells
- Click create Card button
- Custom Card added



Switch Gateway

- Click Switch Jupyter Kernel Gateway button
- Connect to different Jupyter Kernel Gateway
 - <https://myurl.com:9999>
 - <wss://myurl.com:9999>
- Click Connect





Insights Scripting

Introduction

Setting up the Gateway

Required components

- Jupyter Kernel Gateway
 - Python Kernel
 - R Kernel
- Install and configuring using
 - Docker
 - Anaconda
- TLS/SSL configuration

Outlined on Github

- [Insights scripting guide – Github](#)

📁 Docker	Update docker file comments	9 days ago
📁 Graphics	Initial commit	7 months ago
📁 Py	Update README.md	3 months ago
📁 R	Initial commit	7 months ago
📄 .gitignore	Initial commit	7 months ago
📄 LICENSE	Initial commit	7 months ago
📄 README.md	Remove beta restrictions	4 days ago

📖 README.md

ArcGIS Insights scripting guide

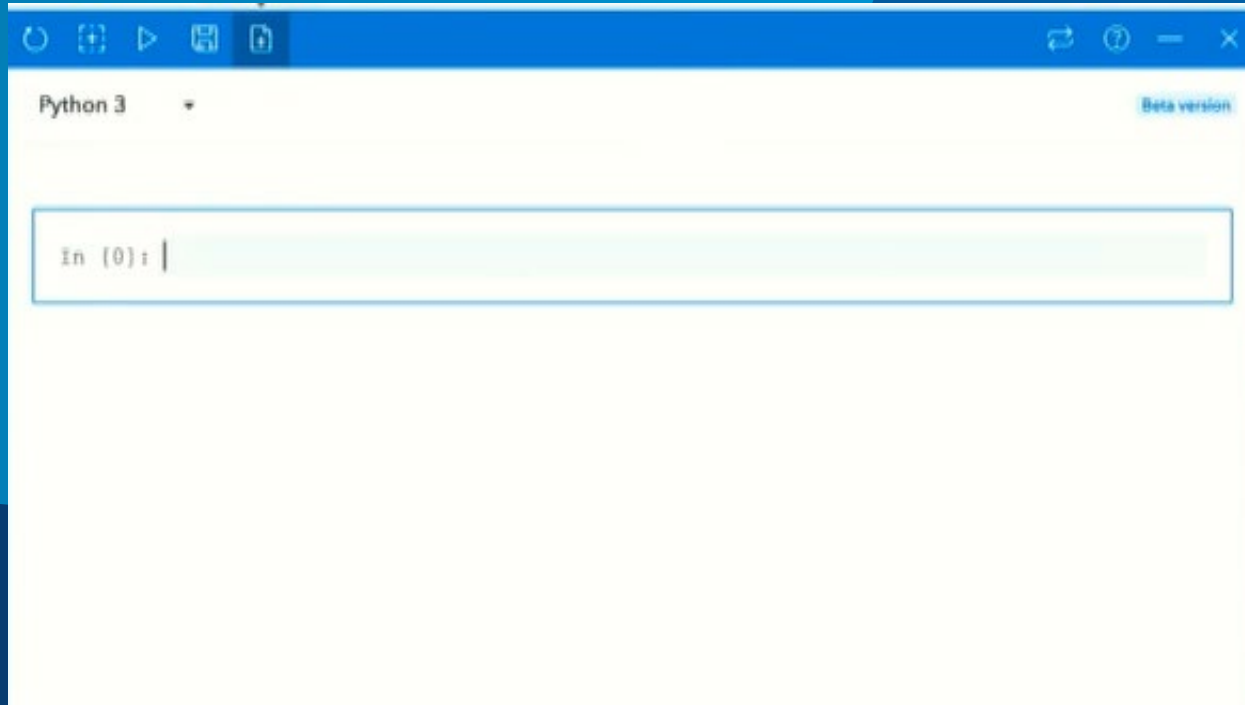
Welcome to the ArcGIS Insights scripting guide.

This repo contains information on getting started with scripting in ArcGIS Insights. This feature enables the execution of Python and R code using a bring-your-own scripting environment. Among many topics, we outline how to deploy a Jupyter Kernel Gateway, how to connect to a Jupyter Kernel Gateway, and tips and tricks for using the ArcGIS Insights scripting environment with other ArcGIS Insights features. In addition, the repo serves as a place to find and share useful Python and R scripts and creates a community around those who prefer to write code to advance data science and knowledge sharing.

We welcome any contributions for improving this guide. If you find a bug or would like to report a problem in the ArcGIS Insights scripting environment, please log an issue in this repo.

Overview

- Learn how to configure a Jupyter Kernel Gateway (this is a requirement for using the ArcGIS Insights scripting environment).
- Pick a [Windows](#) or [macOS](#) machine, such as a desktop or laptop, to host your Jupyter Kernel Gateway.
- Generate Transport Layer Security (TLS) or Secure Sockets Layer (SSL) certificates.
- Follow the guide for installing and deploying a Jupyter Kernel Gateway.
- Learn tips about keyboard shortcuts and important scripting options.
- Search for Python and R code.
- Contribute Python and R code.
- Read ArcGIS Insights use cases and product documentation.
- Try ArcGIS Insights in Enterprise.



Insights Scripting

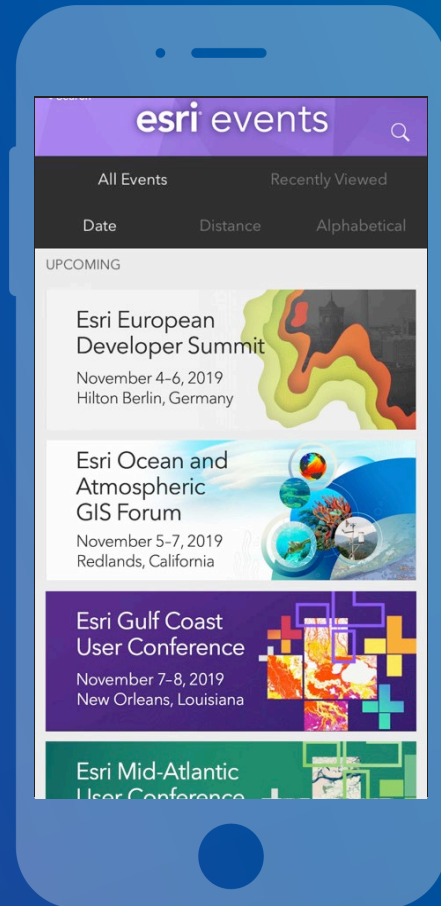
Data management with Python

Conclusion

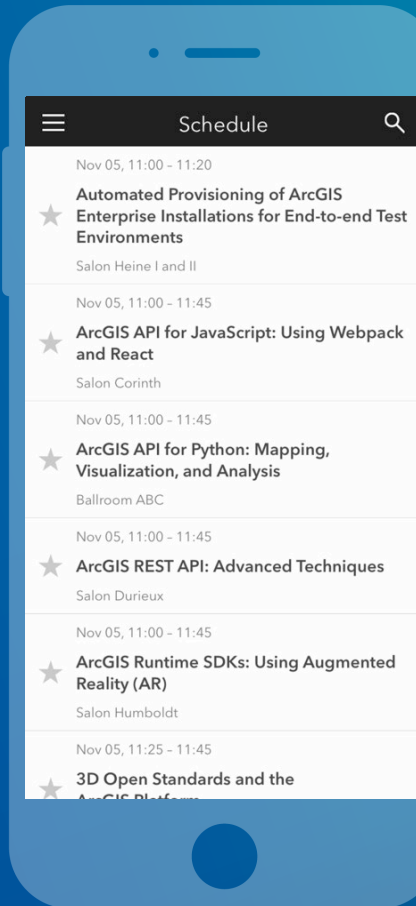
- Leveraging the existing environment
- Write your own code
- Import your own script
- Notebook Server will be the default kernel (you can bring notebooks)
- Available in ArcGIS Enterprise with Insights 3.2
- In the future for ArcGIS Online (no date)
- Ask questions on the Github page → place to collaborate
- Take into account your advice to improve the roadmap

Please Take Our Survey on the App

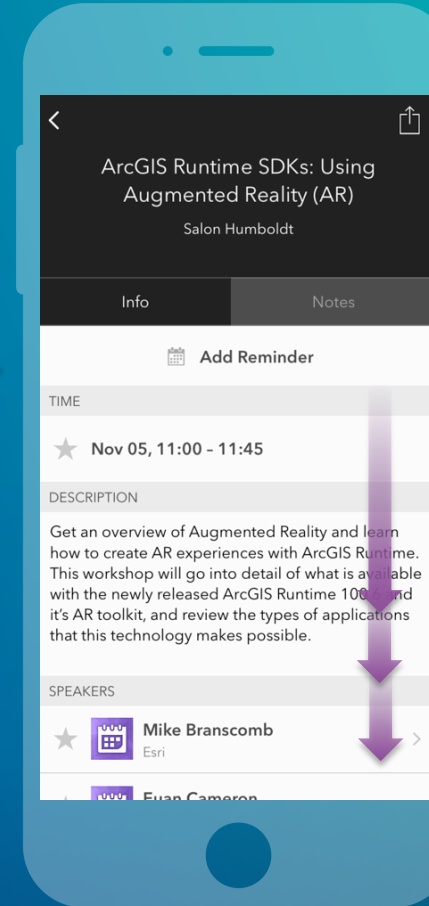
Download the Esri Events app and find your event



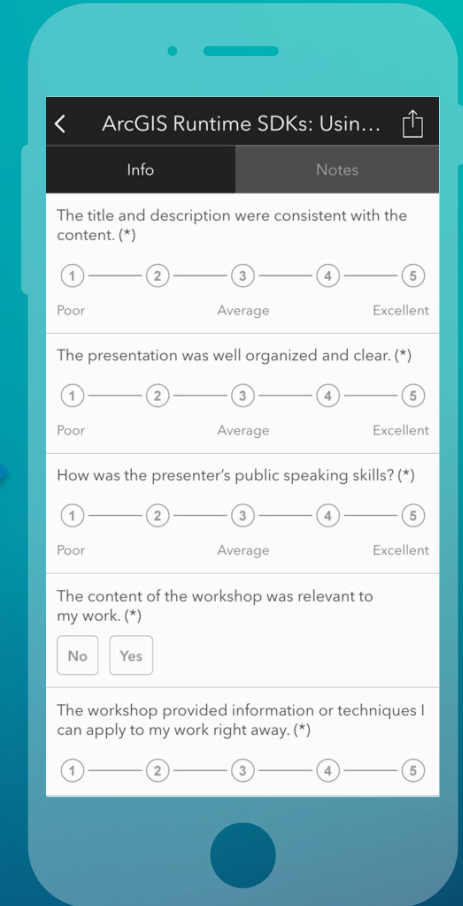
Select the session you attended



Scroll down to find the feedback section



Complete answers and select "Submit"





esri

THE
SCIENCE
OF
WHERE