



Think Inside the Box(es)

Excel-Hosted Dashboards With
Python Graphics

Lightning Talks – Ted Conway





TODAY'S GAME PLAN

AGENDA

Why Host Python Graphics in Excel Dashboards?

An Excel-Based Python Graphics Playground

A Quick Excel Dashboard Tour, Including a New York Times Data Viz Knockoff

A Step-by-Step Look at How Things Works



WHY EXCEL ?

JupyterLite

jupyter.org/try-jupyter/lab/

File Edit View Run Kernel Tabs Settings Help

Untitled.ipynb

[2]:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
fruits = ['apple', 'blueberry', 'cherry', 'orange']
counts = [40, 100, 30, 55]
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']
ax.bar(fruits, counts, color=bar_colors)
ax.set_ylabel('fruit supply')
ax.set_title('Fruit supply by kind and color')
plt.show()
```

Fruit supply by kind and color

Simple 0 \$ 2 Python (Pyodide) | Idle Mode: Command 2:53 PM 12/2/2022

JUPYTER HOSTED

▶ What's Dash?

▼ Dash Tutorial

Part 1. Installation

Part 2. Layout

Part 3. Basic Callbacks

Part 4. Interactive Graphing and Crossfiltering

Part 5. Sharing Data Between Callbacks

▶ Dash Callbacks

▶ Open Source Component Libraries

▶ Enterprise Component Libraries

▶ Creating Your Own Components

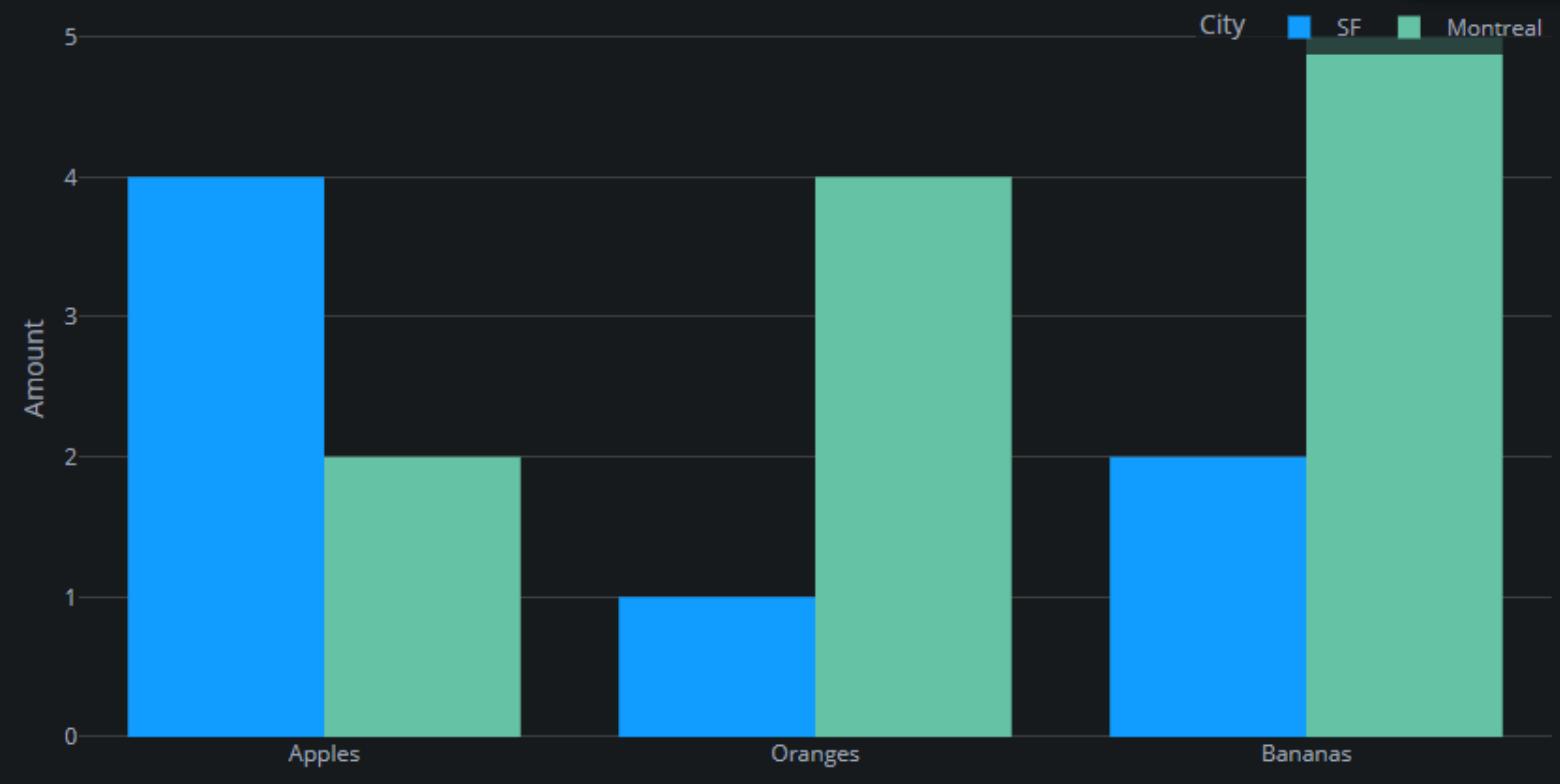
▶ Beyond the Basics

▶ Ecosystem Integration

▶ Production Capabilities

Hello Dash

Dash: A web application framework for your data.



DASH/PLOTLY
HOSTED

Feedback





Panel

Overview Getting Started User Guide Gallery Reference Gallery Developer Guide API Reference Releases F...

PANEL

Search the docs ...

Overview
Getting Started
User Guide
Gallery
Reference Gallery
Developer Guide
API Reference
Releases
FAQ
About

Usage
Sponsors



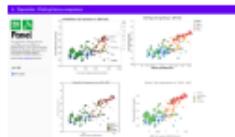
Panel

A high-level app and dashboarding solution for Python

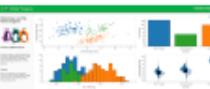
Attractors



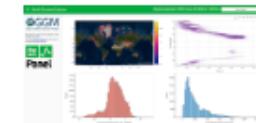
Gapminders



Penguins



Glaciers



Portfolio Optimizer



Panel is an [open-source](#) Python library that lets you create custom interactive web apps and dashboards by connecting user-defined widgets to plots, images, tables, or text.

Penguin K-Means Clustering

Slideshow

Output

Code

X

hill:length mm

22

■

Wind

9:22 PM
12/2/2022

WHY EXCEL

UPSIDE

- Ubiquitous, no install, simple, minimal training
- Python-generated graphics images **automatically sized and positioned** to match containing cells and shapes, and **inserted into Excel**
- Easily **organize and share** “live” Python code snippets in workbooks for learning or teaching
- Can create Excel-based **Python graphics dashboards**

CAVEAT

- Simplicity a feature and a bug – **won’t replace** other Open Source and commercial dataviz software



EXCEL “PLAYGROUND”

AutoSave Off | PyDataGlobalDemo.xlsxb | Search (Alt+Q) | ----- | Comments | Share

File Home Insert Draw Page Layout Formulas Data New Tab Review View Developer Help New Tab

A1 : import matplotlib.pyplot as plt

import matplotlib.pyplot as plt
Use special variables XLwidth and XLheight to make image cell-sized
fig, ax = plt.subplots(figsize=(XLwidth/72., XLheight/72.))
fruits = ['apple', 'blueberry', 'cherry', 'orange']
counts = [40, 100, 30, 55]
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']
ax.bar(fruits, counts, color=bar_colors)
ax.set_ylabel('fruit supply')
ax.set_title('Fruit supply by kind and color')
Instead of plt.show(), save image as name in special variable XLimage
plt.savefig (XLimage, dpi=100)
plt.close()

#PYTHON

XLheight

XLwidth

Toolbar Buttons

① Draw Charts

② Delete Charts

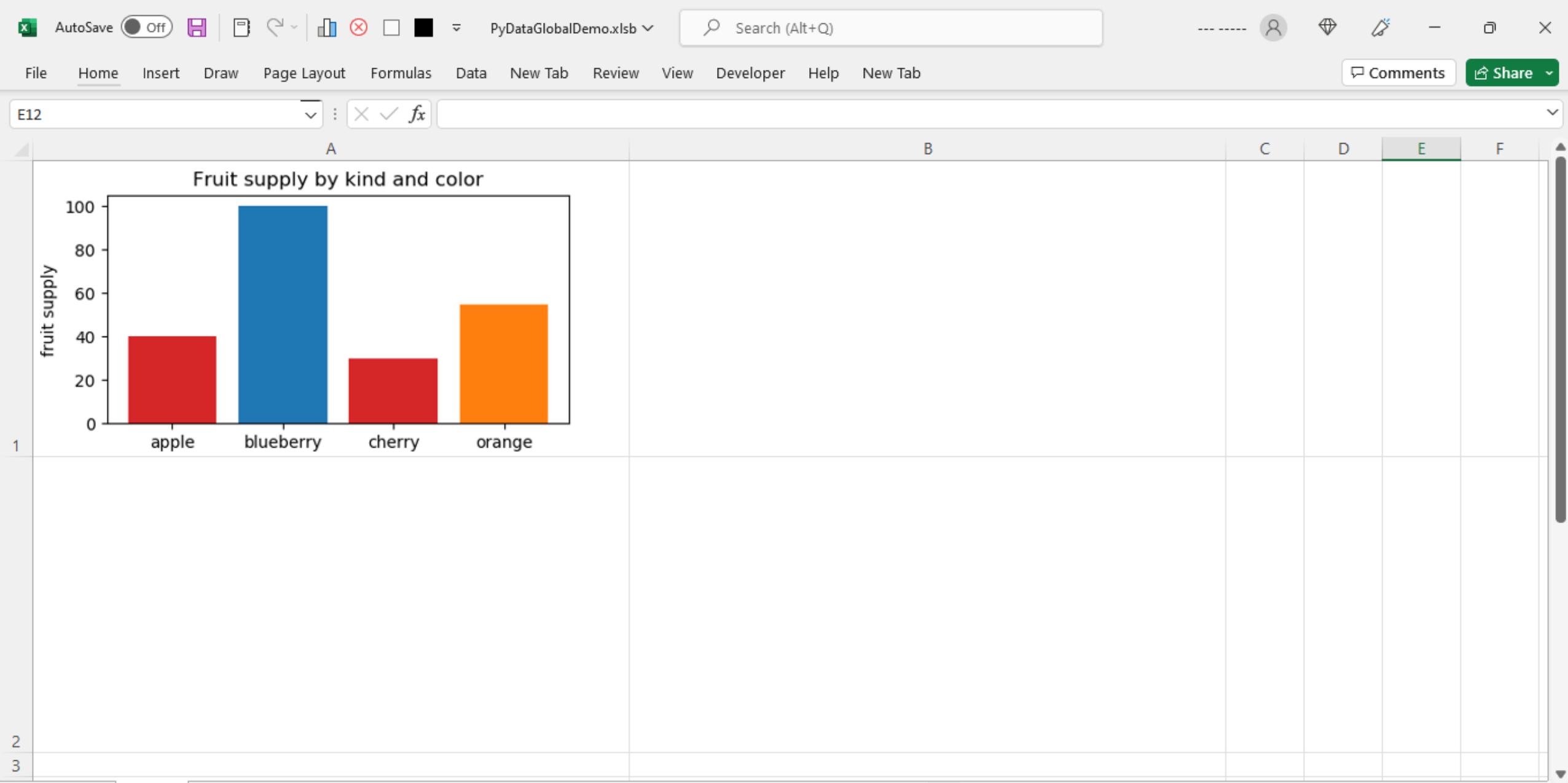
③ Show Charts

④ Hide Charts

DEMO PLOTLY MATPLOTLIB NY TIMES KNOCKOFF +

Cell A1 note by -----

Windows Taskbar: File (49), Excel, Powerpoint, 3:12 PM, 12/2/2022, 48°F Cloudy



- **Python chart-generating code** is placed in Excel cells tagged with comments starting with “#PYTHON”
- Clicking a button causes VBA macro to **write Python code** to a file, including statements that **create variables** with cell dimensions (**XLwidth**, **XLheight**) and generated filenames (**XLimage**)
- **SendKeys** is used to send an **exec** statement to Python, causing it to **import code & create images**
- VBA macro **waits for Python** to complete (checks for “trigger” file) and then **inserts images** over cells that contained code

B1

Name Box

A

```
import matplotlib.pyplot as plt  
# Use special variables XLwidth and XLheight to make image cell-sized  
fig, ax = plt.subplots(figsize=(XLwidth/72., XLheight/72.))  
fruits = ['apple', 'blueberry', 'cherry', 'orange']  
counts = [40, 100, 30, 55]  
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']  
ax.bar(fruits, counts, color=bar_colors)  
ax.set_ylabel('fruit supply')  
ax.set_title('Fruit supply by kind and color')  
# Instead of plt.show(), save image as name in special variable XLimage  
plt.savefig (XLimage, dpi=100)  
plt.close()
```

```
import matplotlib.pyplot as plt  
# Use special variables XLwidth and XLheight to make image cell-sized  
fig, ax = plt.subplots(figsize=(XLwidth/72., XLheight/72.))  
fruits = ['apple', 'blueberry', 'cherry', 'orange']  
counts = [40, 100, 30, 55]  
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']  
ax.bar(fruits, counts, color=bar_colors)  
ax.set_ylabel('fruit supply')  
ax.set_title('Fruit supply by kind and color')  
# Instead of plt.show(), save image as name in special variable XLimage  
plt.savefig (XLimage, dpi=100)  
plt.close()
```

```
import matplotlib.pyplot as plt  
# Use special variables XLwidth and XLheight to make image cell-sized  
fig, ax = plt.subplots(figsize=(XLwidth/72., XLheight/72.))  
fruits = ['apple', 'blueberry', 'cherry', 'orange']  
counts = [40, 100, 30, 55]  
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']  
ax.bar(fruits, counts, color=bar_colors)  
ax.set_ylabel('fruit supply')  
ax.set_title('Fruit supply by kind and color')  
# Instead of plt.show(), save image as name in special variable XLimage  
plt.savefig (XLimage, dpi=100)  
plt.close()
```

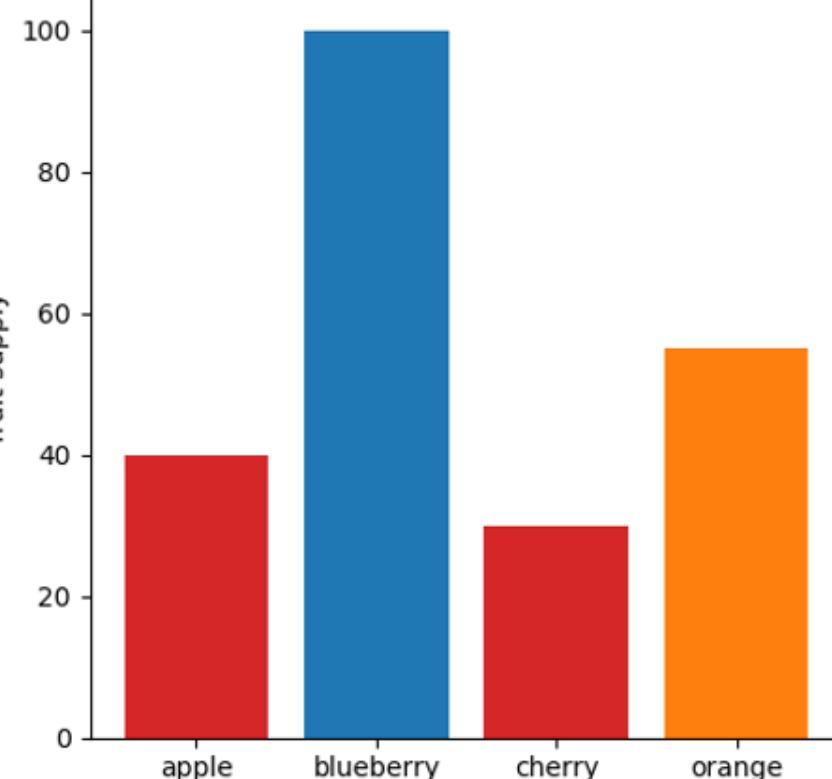
SAME CHART
X 3

COPY CODE
OR USE
FORMULAS

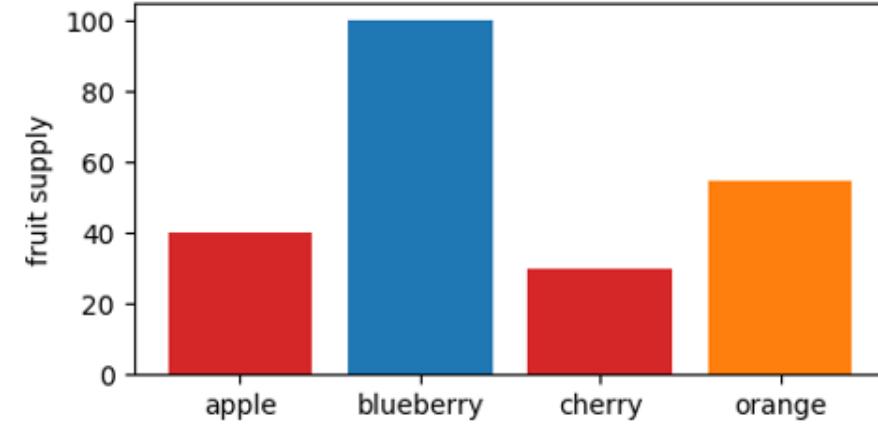
CHARTS CAN
SPAN CELLS

import matplotlib.pyplot as plt

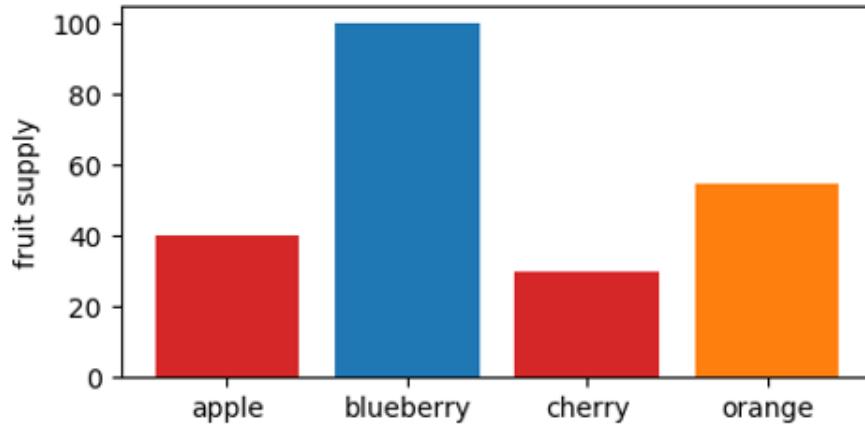
Fruit supply by kind and color



Fruit supply by kind and color



Fruit supply by kind and color



DEMO

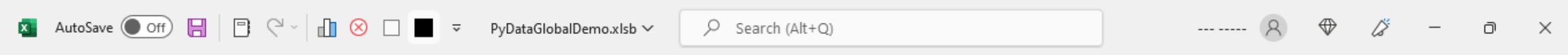
PLOTLY

MATPLOTLIB

NY TIMES KNOCKOFF

DASHBOARDS





File Home Insert Draw Page Layout Formulas Data New Tab Review View Developer Help New Tab

MATPLOTLIB EXAMPLE

```
A16 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
```

```
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as  
mdates  
df=pd.read_csv('c://temp//carsdata.csv')  
bar_counts = df['Cylinders'].value_counts().sort_index()  
bar_x = df['Cylinders'].value_counts().sort_index().index  
bar_height = df['Cylinders'].value_counts().sort_index().values  
plt.figure(figsize=(XLwidth/72., XLheight/72.))  
plt.bar(x=bar_x.astype(str), height=bar_height, color="#087E8B",  
ec="#02454d")  
plt.title('Bar chart of CYL')  
plt.xlabel('Number of cylinders')  
plt.ylabel('Count')  
plt.savefig (XLimage, dpi=100)  
plt.close()
```

```
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c://temp//carsdata.csv')
colors = []
for val in df['Cylinders']:
    if val <= 4: colors.append('#17314c')
    elif val <= 6: colors.append('#326b99')
    else: colors.append('#54aef3')
plt.figure(figsize=(XLwidth/72., XLheight/72.))
plt.scatter(x=df['Horsepower'], y=df['MPG_City'], s=df['Cylinders'] * 20, c=colors)
plt.title('Scatter plot of HP vs MPG')
plt.xlabel('Horse power')
plt.ylabel('Miles per gallon')
plt.savefig (XImage, dpi=100)
plt.close()
```

```

import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c://temp//carsdata.csv')
dfCylMPG = []
unqCyl=df[df.Cylinders.notnull()]['Cylinders'].unique().tolist()
unqCyl.sort()
for x in unqCyl:
    dfCylMPG.append(df[df['Cylinders'] == x]['MPG_City'].tolist())
plt.figure(figsize=(XLwidth/72, XLheight/72))
bp=plt.boxplot(dfCylMPG, patch_artist=True)
for box in bp['boxes']:
    box.set(facecolor='#087E8B', alpha=0.6, linewidth=2)
for whisker in bp['whiskers']:
    whisker.set(linewidth=2)
for median in bp['medians']:
    median.set(color='black', linewidth=3)
plt.title('Boxplot of CYL vs MPG')
plt.xlabel('Number of cylinders')
plt.ylabel('Miles per gallon')
plt.xticks(list(range(1, len(unqCyl)+1)), unqCyl)
plt.show()

```

```
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c://temp//air.csv')
df['DATE'] = df['DATE'].apply(lambda x: pd.to_datetime(x))
plt.figure(figsize=(XLwidth/72., XLheight/72.))
line = plt.plot(df['DATE'], df['AIR'], lw=2.5, color="#087E8B")
formatter = mdates.DateFormatter("%Y")
locator = mdates.YearLocator()
plt.title("Line chart of Airline passengers")
plt.xlabel('Year')
plt.ylabel('Count')
plt.savefig (XLimage, dpi=100)
plt.close()
```

```
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c://temp/carsdata.csv')
plt.figure(figsize=(XLwidth/72., XLheight/72.))
plt.hist(df['MPG_City'], bins=15, color="#087E8B", ec="#02454d")
plt.title('Histogram of MPG')
plt.xlabel('MPG')
plt.ylabel('Count');
plt.savefig (XImage, dpi=100)
plt.close()
```

◀ ▶ | DEMO | **MATPLOTLIB** | PLOTLY | NY TIMES KNOCKOFF

DEMO

matplotlib

PLOTLY

NY TIMES KNOCKOFF

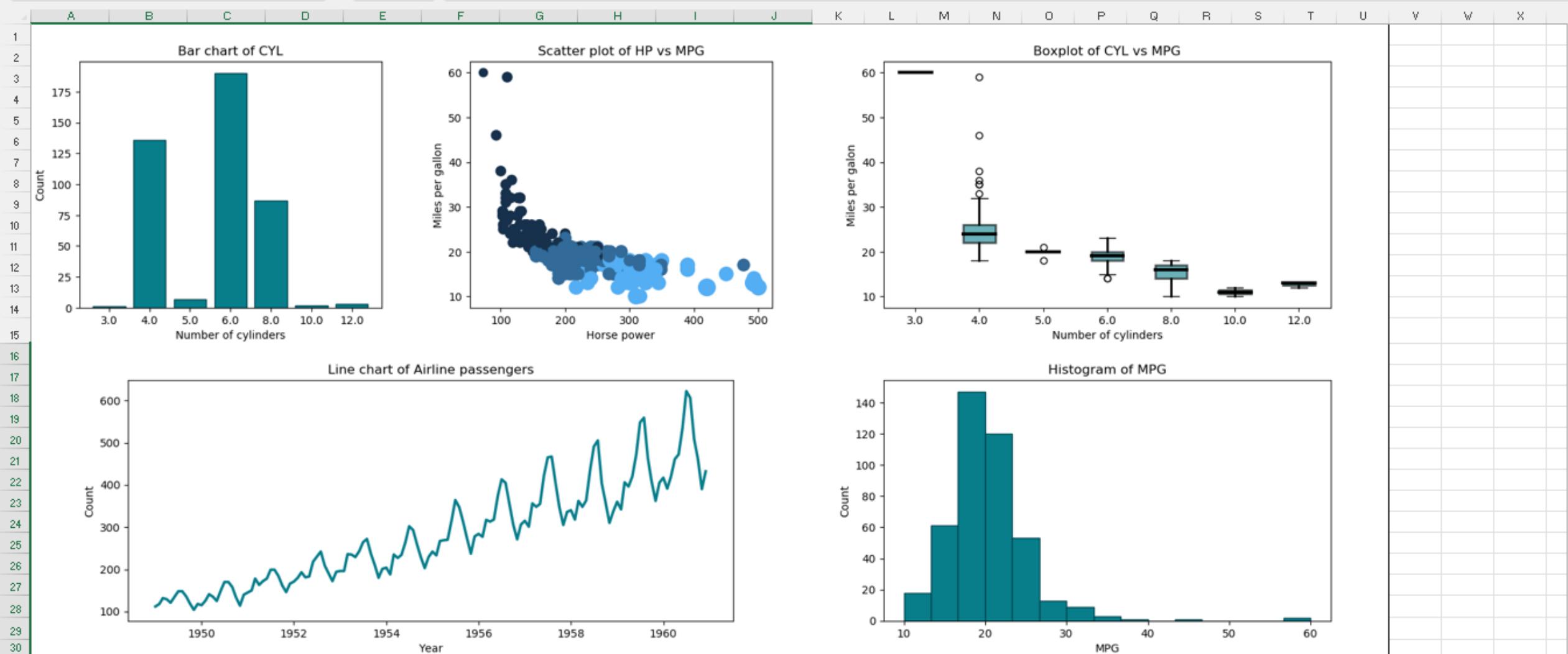
+

Ready Accessibility: Investigate



A16

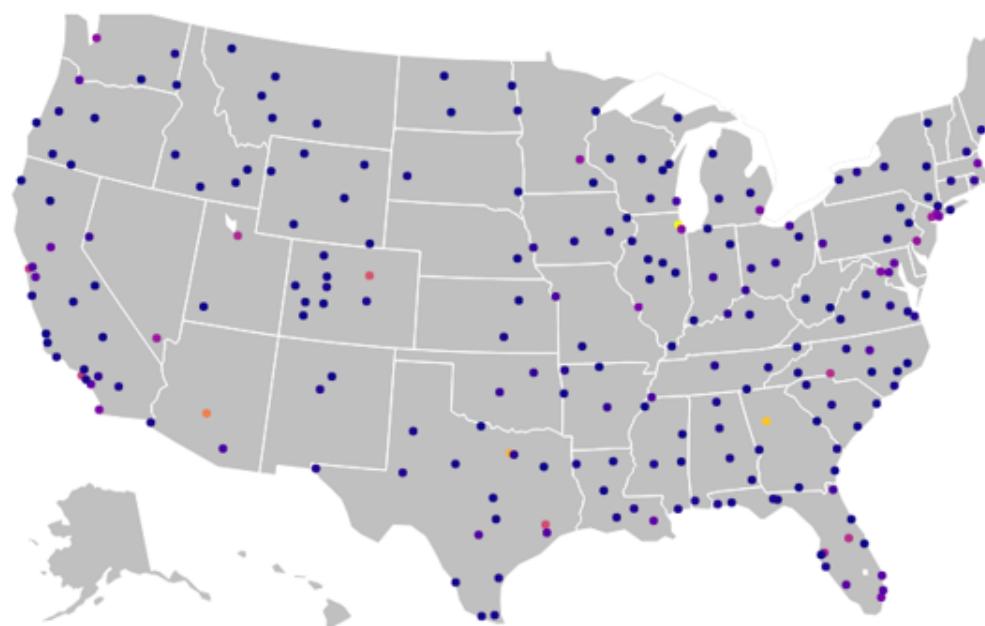
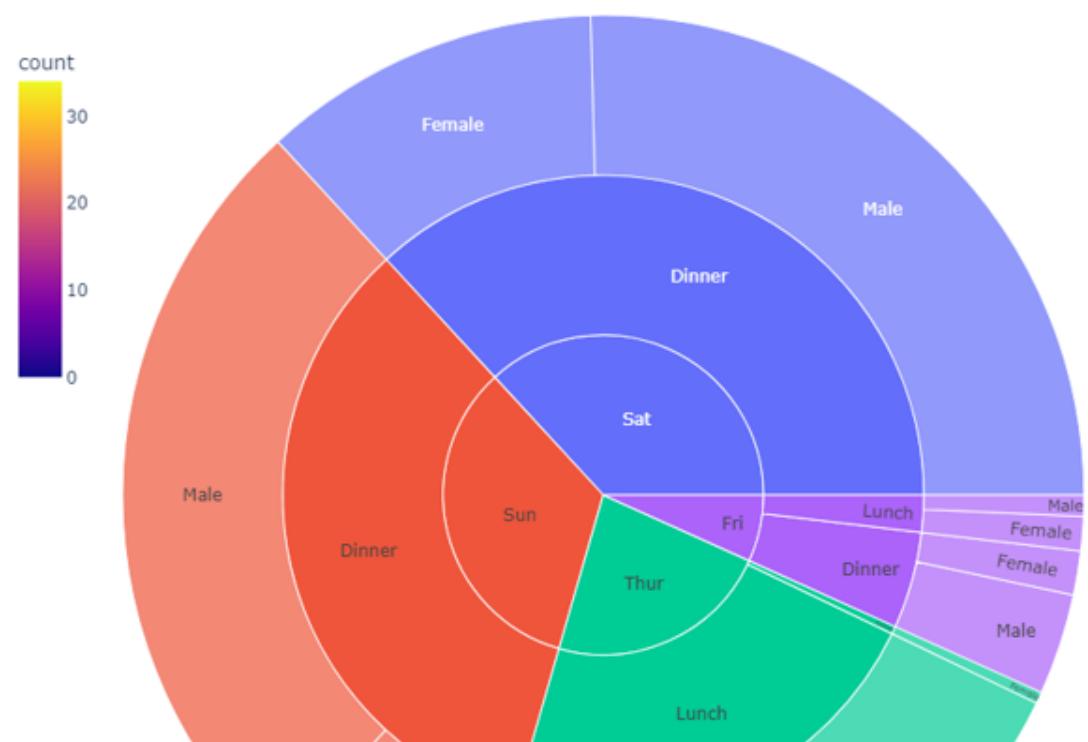
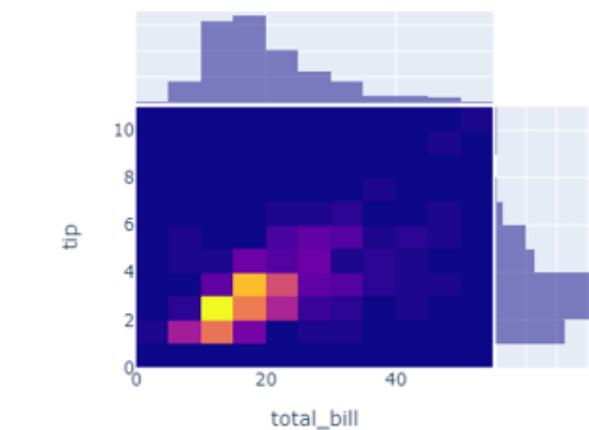
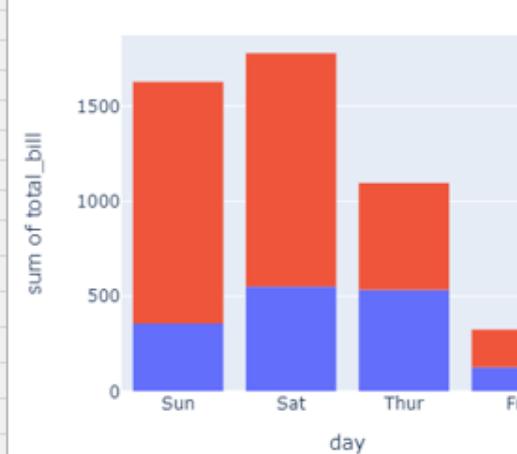
fx import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates



PyDataGlobalDemo.xlsb - Excel

```
1 import pandas as pd, plotly.express as px
2 df = px.data.tips()
3 fig = px.histogram(df, x="day", y="total_bill", color="sex")
4 fig.update_layout(autosize=False, width=XLwidth/72.*100,
5 height=XLheight/72.*100)
6 fig.write_image(XLimage)
7
8
9
10
11
12
13
14
15
16
17
18 import pandas as pd, plotly.express as px, plotly.graph_objects as go
19 #df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/2011_february_us_airport_traffic.csv')
20 df = pd.read_csv('c://temp//2011_february_us_airport_traffic.csv')
21 df['text'] = df['airport'] + " " + df['city'] + ', ' + df['state'] + " " + 'Arrivals: ' + df['cnt'].astype(str)
22 fig = go.Figure(data=go.Scattergeo(lon=df['long'], lat=df['lat'], text=df['text'], mode='markers', marker_color=df['cnt'], ))
23 fig.update_layout(
24     geo = dict(scope='usa', projection_type='albers usa', showland = True, landcolor="#C0C0C0", fitbounds="geojson"),
25     margin=dict(t=5, b=5, l=5, r=5), autosize=False, width=XLwidth/72.*100, height=XLheight/72.*100)
26 fig.write_image(XLimage)
27
28
29
30
31
32
33
34
35
36
```

SAMPLE PLOTLY CHARTS

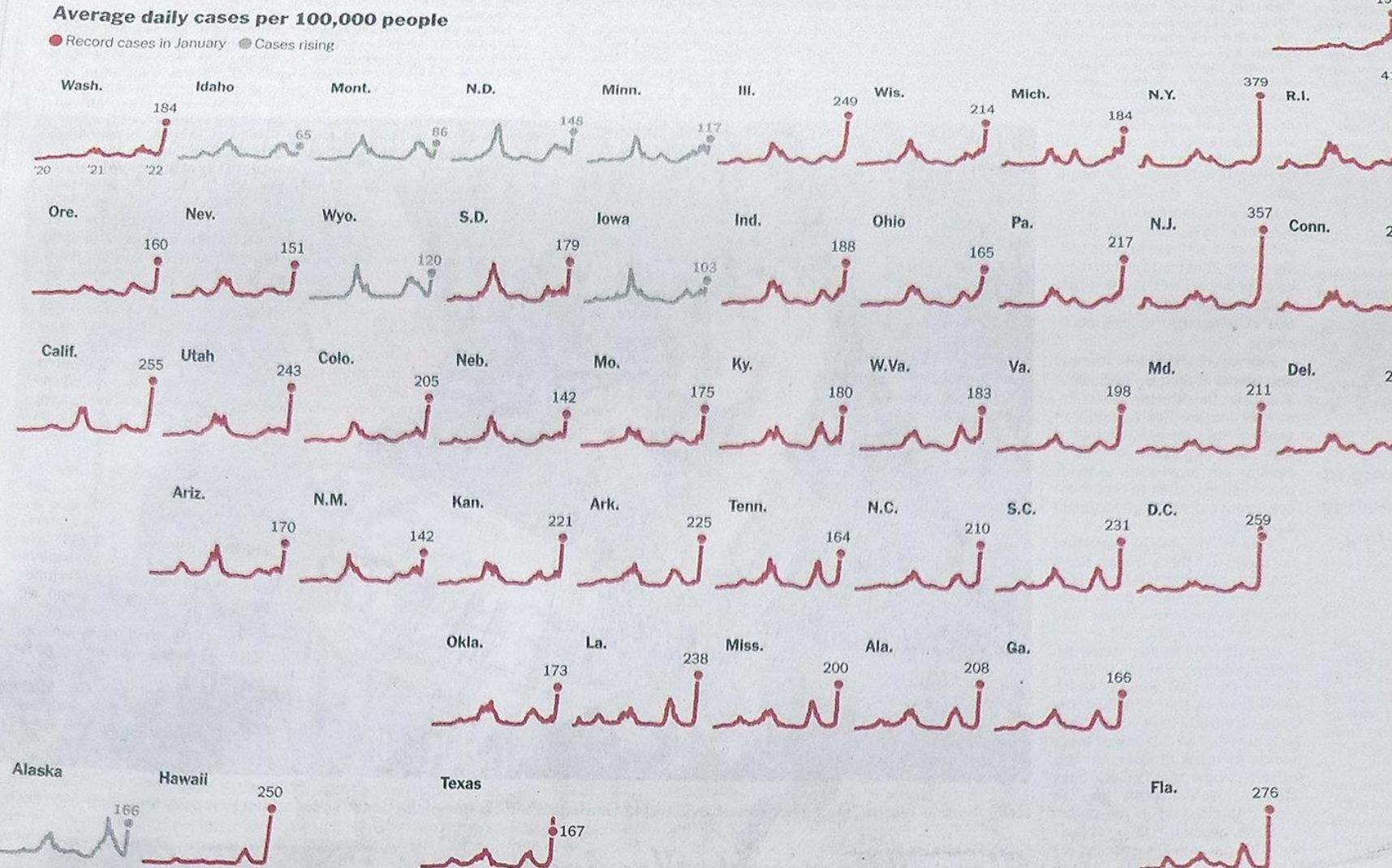


SAMPLE PLOTLY CHARTS

What to Make of Those Soaring Covid Counts

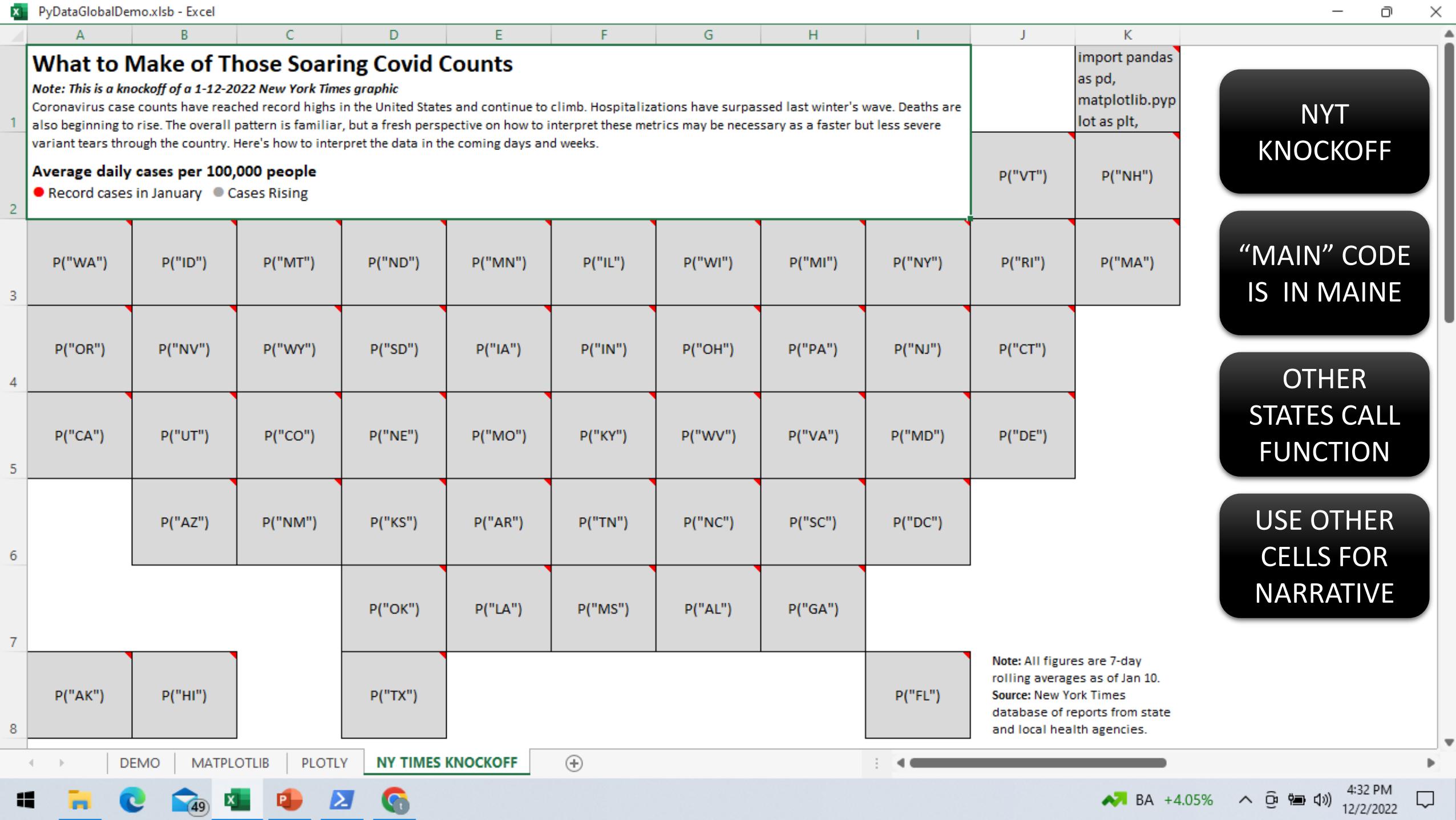
This article is by Lazario Gamio, Lisa Waananen Jones and Amy Schoenfeld Walker.

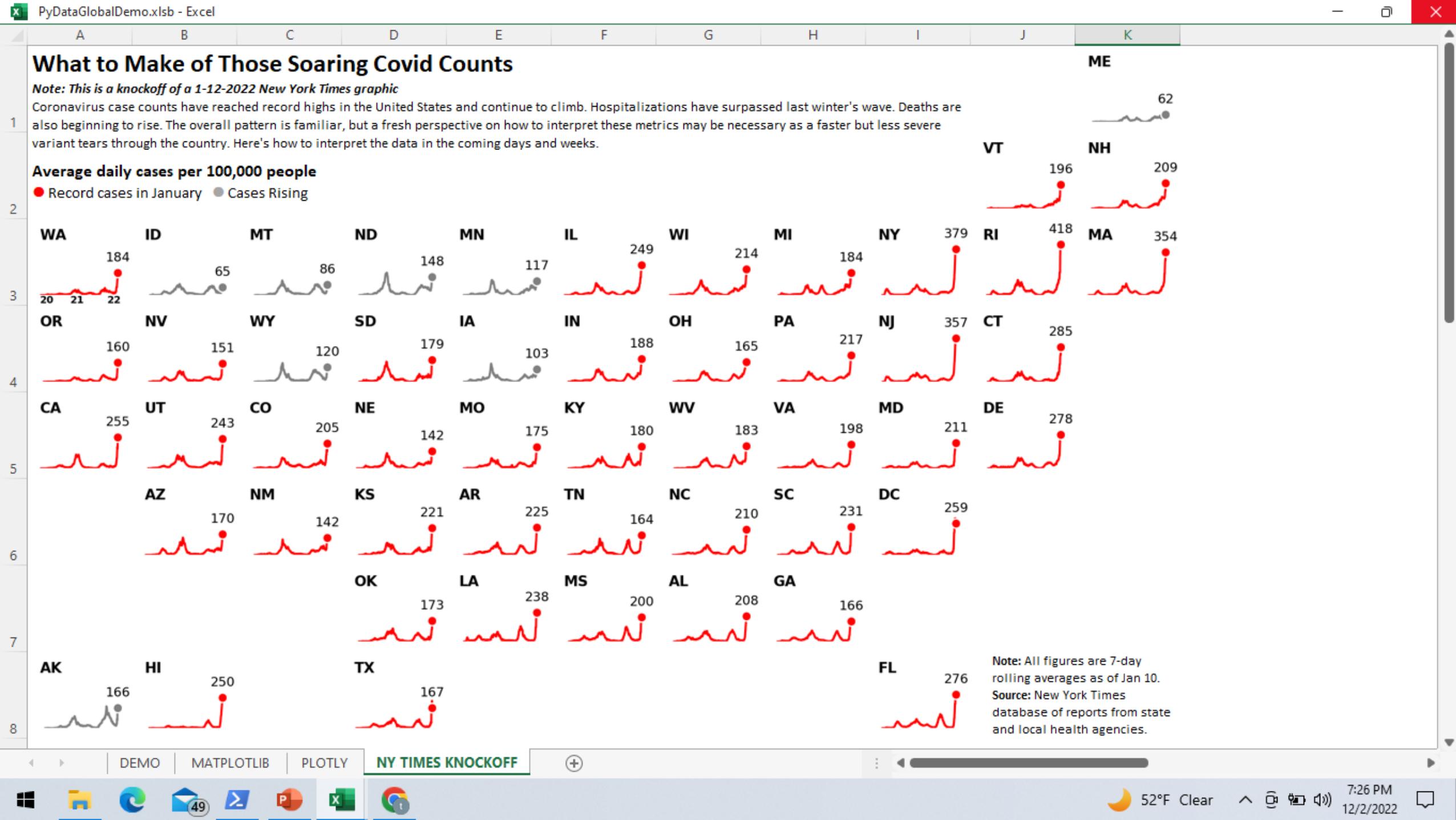
Coronavirus case counts have reached record highs in the United States and continue to climb. Hospitalizations have surpassed last winter's wave. Deaths are also beginning to rise. The overall pattern is familiar, but a fresh perspective on how to interpret these metrics may be necessary as a faster but less severe variant tears through the country. Here's how to interpret the data in the coming days and weeks.



Note: All figures are seven-day rolling averages as of Jan. 10.
Source: New York Times database of reports from state and local health agencies

THE NEW YORK TIMES





W
D
O
C
U

```
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates # Replicate 1-12-2-2022 NYT Covid Case Map
dfall=pd.read_csv('c://temp//nyt.csv') # Get NYT data
def P(stcode): # Function P plots chart for each state
    df=dfall[dfall["st"]==stcode] # Get daily case data for state
    df['date'] = df['date'].apply(lambda x: pd.to_datetime(x)) # Convert dates to datetimes
    plt.figure(figsize=(XLwidth/72., XLheight/72.)) # Make chart same size as Excel cell
    plt.axis('off') # Turn off axis
    ax = plt.gca() # Fix axes limits for all charts
    ax.set_xlim([datetime.date(2020, 1, 21), datetime.date(2022, 2, 10)]) # 2020-1-21 to 2022-2-10 report dates
    ax.set_ylim(ymin=0, ymax=550) # 0 to 550 incidents
    plt.text(0, 1, df["st"].iloc[-1], ha='left', va='top', transform=ax.transAxes, color='black', fontsize=10, weight="bold") # State code in upper left corner
    plt.text(x=df['date'].iloc[-1], y=df['cases_avg_per_100k'].iloc[-1]+75, s=df['EndingCasesX'].iloc[-1].astype(int).astype(str), color='black', fontsize=9, ha='center', va='bottom') # Ending number of cases over last point
    line = plt.plot(df['date'], df['cases_avg_per_100k'], lw=1.5, color=df['trend'].iloc[-1]) # Line plot of cases/100K of population,
    red/gray colors denote case trend
    plt.scatter(x=df['date'].iloc[-1], y=df['cases_avg_per_100k'].iloc[-1], s=44, facecolors=df['trend'].iloc[-1], edgecolors='white', zorder=10) # Marker surrounded by whitespace for ending #
of cases
    if df["st"].iloc[-1]=="WA": # Years of data eyecatcher for Washington State
        chart - 20/21/22
        plt.text(0, 0, "20", ha='left', va='top', transform=ax.transAxes, fontsize=7, weight="bold")
        plt.text(x=pd.to_datetime('2021-01-01'), y=1, s='21', ha='center', va='top', fontsize=7, weight="bold")
        plt.text(1, 0, "22", ha='right', va='top', transform=ax.transAxes, fontsize=7, weight="bold")
        plt.savefig (XLimage, dpi=200) # Save figure to unique filename
        plt.close() # Close file
P("ME") # Invoke plotting function for Maine
```

HOW IT WORKS



COMMUNICATION



Windows PowerShell

PS C:\Users\tedco> python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

AutoSave (Off) | PyDataGlobalDemo.xlsxb | Search (Alt+Q)

File Home Insert Draw Page Layout Formulas Data New Tab Review View Developer Help New Tab Comments Share

A1 : import matplotlib.pyplot as plt

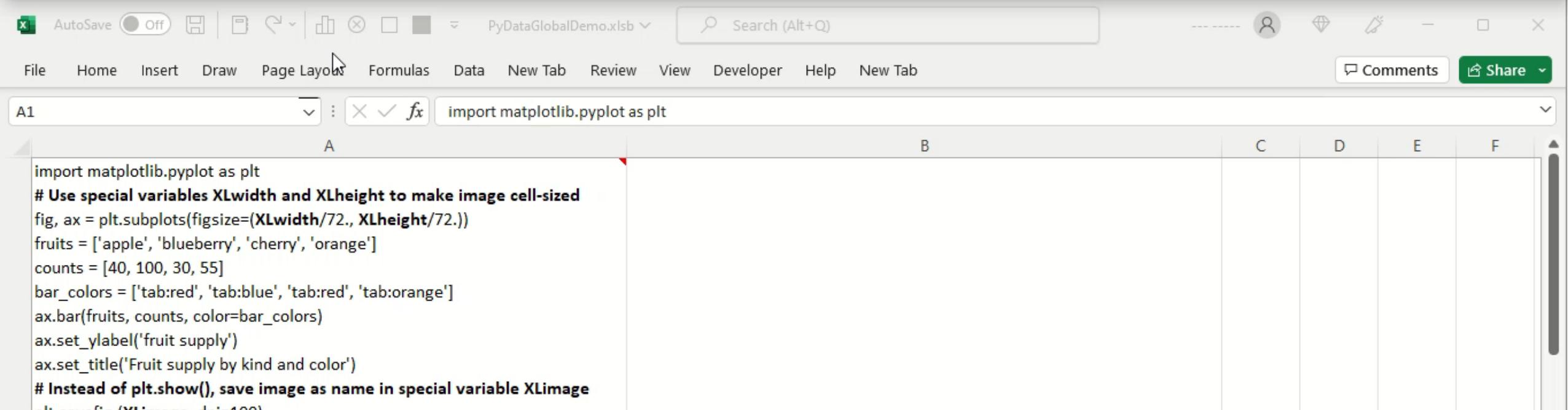
```
import matplotlib.pyplot as plt
# Use special variables XLwidth and XLheight to make image cell-sized
fig, ax = plt.subplots(figsize=(XLwidth/72., XLheight/72.))
fruits = ['apple', 'blueberry', 'cherry', 'orange']
counts = [40, 100, 30, 55]
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']
ax.bar(fruits, counts, color=bar_colors)
ax.set_ylabel('fruit supply')
ax.set_title('Fruit supply by kind and color')
# Instead of plt.show(), save image as name in special variable XLimage
plt.savefig (XLimage, dpi=100)
plt.close()
```

DEMO PLOTLY MATPLOTLIB NY TIMES KNOCKOFF +

Ready Accessibility: Investigate

48°F Cloudy 3:22 PM 12/2/2022

```
PS C:\Users\tedco> python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



The screenshot shows a Microsoft Excel spreadsheet titled "PyDataGlobalDemo.xlsx". In cell A1, there is Python code for generating a bar chart. The code imports matplotlib.pyplot, creates a figure and axis, defines fruit names and counts, sets colors for each bar, and saves the plot as an image. The code is as follows:

```
import matplotlib.pyplot as plt
# Use special variables XLwidth and XLheight to make image cell-sized
fig, ax = plt.subplots(figsize=(XLwidth/72., XLheight/72.))
fruits = ['apple', 'blueberry', 'cherry', 'orange']
counts = [40, 100, 30, 55]
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']
ax.bar(fruits, counts, color=bar_colors)
ax.set_ylabel('fruit supply')
ax.set_title('Fruit supply by kind and color')
# Instead of plt.show(), save image as name in special variable XLimage
plt.savefig (XLimage, dpi=100)
plt.close()
```

A RECORDED DEMO

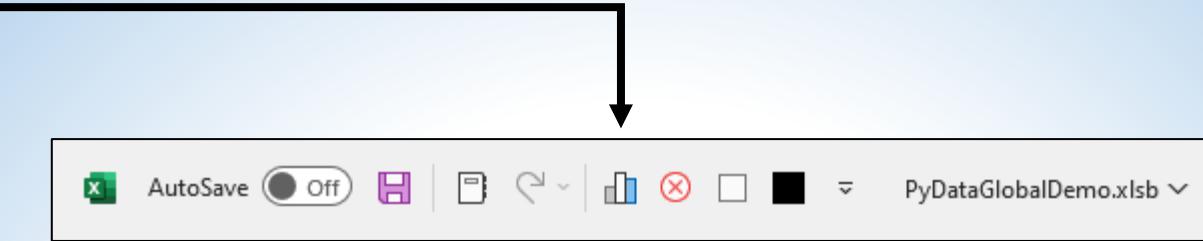


STEP-BY-STEP



Python chart-generating code is entered into image-sized Excel cells





Excel toolbar button is pressed to run chart generation macro



Windows PowerShell
PS C:\Users\stedco> python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

```
1 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
2 df=pd.read_csv('c:/temp/carsdata.csv')
3 bar_counts = df['Cylinders'].value_counts().sort_index()
4 bar_x = df['Cylinders'].value_counts().sort_index().index
5 bar_height = df['Cylinders'].value_counts().sort_index().values
6 plt.figure(figsize=(X.width/72., X.height/72.))
7 plt.bar(x=bar_x.astype(str), height=bar_height, color="#087E8B",
8 ec="#0245d4")
9 plt.title('Bar chart of CYL')
10 plt.xlabel('Number of cylinders')
11 plt.ylabel('Count')
12 plt.savefig (XLimage, dpi=100)
13 plt.close()
14
15
16 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
17 df=pd.read_csv('c:/temp/air.csv')
18 df['DATE']=pd.date_range(pd.datetime(1949, 1, 1), pd.to_datetime(x))
19 plt.figure(figsize=(X.width/72., X.height/72.))
20 line = plt.plot(df['DATE'], df['ARR'], lw=2.5, color="#087E8B")
21 formatter = mdates.DateFormatter('%Y')
22 locator = mdates.YearLocator()
23 plt.title('Line chart of Airline passengers')
24 plt.xlabel('Year')
25 plt.ylabel('Count')
26 plt.savefig (XLimage, dpi=100)
27
28
29
30
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c:/temp/carsdata.csv')
plt.figure(figsize=(X.width/72., X.height/72.))
df=df[df['Cylinders'].notnull()]['Cylinders'].unique().tolist()
unqCyl = []
for val in df['Cylinders']:
    if val <= 4: colors.append('#f7314c')
    elif val <= 6: colors.append('#32b899')
    else: colors.append('#54aeff')
20, c=colors
plt.figure(figsize=(X.width/72., X.height/72.))
bp=plt.boxplot(df['MPG'], patch_artist=True)
for box in bp['boxes']:
    box.set(facecolor="#087E8B", alpha=0.6, linewidth=2)
    for whisker in bp['whiskers']:
        whisker.set(linewidth=2)
        median = np.median(box['yrange'])
        median = np.median([box['ymin'], box['q3']])
        plt.axvline(median, color='black', linewidth=3)
    plt.outlierほplot(df['Cyl'], y=MPG)
    plt.xlabel('Number of cylinders')
    plt.ylabel('Miles per gallon')
    plt.xticks(np.arange(1, len(unqCyl)+1), unqCyl)
    plt.xscale('log')
    plt.show(X)

```

DEMO MATPLOTLIB PLOTLY NY TIMES KNOCKOFF

Rain to stop 12:35 AM 12/3/2022

Rain to stop 12:33 AM 12/3/2022



Excel macro writes Python code to file (incl. code to set image sizes)

The screenshot shows a dual-pane interface. The left pane is a Jupyter Notebook cell containing Python code for data visualization. The right pane is a Windows PowerShell window showing the execution of the code.

```
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c:/temp/carsdata.csv')
bar_counts=df['Cylinders'].value_counts().sort_index()
bar_x=df['Cylinders'].value_counts().sort_index().index
bar_height= df['Cylinders'].value_counts().sort_index().values
plt.figure(figsize=(Xwidth/72,Xheight/72))
plt.bar(x=x, y=bar_height, color="#087E8B",
ec="#0245ad")
plt.title('Bar chart of CYL')
plt.xlabel('Number of cylinders')
plt.ylabel('Count')
plt.savefig(XImage, dpi=100)
plt.close()

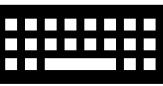
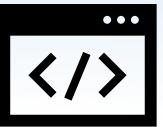
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c:/temp/carsdata.csv')
colors = []
for val in df['Cylinders']:
    if val <= 4: colors.append('#17314c')
    elif val <= 6: colors.append('#326b99')
    else: colors.append('#45aae3')
plt.figure(figsize=(Xwidth/72,Xheight/72))
dfMPG=df[df['Cylinders']==x][['MPG_City']].tolist()
plt.scatter(x=df['Cylinders'], y=df['MPG_City'], s=df['Cylinders'] * 20, c=colors)
plt.title('Scatter plot of HP vs MPG')
plt.xlabel('Horse power')
plt.ylabel('Miles per gallon')
plt.savefig(XImage, dpi=100)
plt.close()

import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df= pd.read_csv('c:/temp/air.csv')
df['DATE']=df['DATE'].apply(lambda x: pd.to_datetime(x))
plt.figure(figsize=(Xwidth/72,Xheight/72))
line = plt.plot(df['DATE'], df['AIRT'], lw=2.5, color="#087E8B")
formatter = mdates.DateFormatter('%Y')
locator = mdates.YearLocator()
plt.title('Line chart of Airline passengers')
plt.xlabel('Year')
plt.ylabel('Count')
plt.savefig(XImage, dpi=100)
plt.close()

import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c:/temp/carsdata.csv')
plt.figure(figsize=(Xwidth/72,Xheight/72))
plt.hist(df['MPG_City'], bins=15, color="#087E8B", ec="#0245ad")
plt.title('Histogram of MPG')
plt.xlabel('MPG')
plt.ylabel('Count')
plt.savefig(XImage, dpi=100)
plt.close()

PS C:\Users\stedco> python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```





```
exec(open('C://temp//Playground//CodeToRun.py').read())
```



Excel macro sends keystrokes to Python with statements to run code

The screenshot shows a dual-monitor setup. The left monitor displays a Microsoft Word document titled "PyDataGlobalDemo.xlsx" containing Python code for data visualization using pandas and matplotlib. The right monitor displays a Windows PowerShell window with the command "python" running, showing the Python version and a file reading operation.

```
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates

1 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as
2 mdates
3 df=pd.read_csv('c:/temp/carsdata.csv')
4 bar_counts = df['Cylinders'].value_counts().sort_index()
5 bar_x = df['Cylinders'].value_counts().sort_index().index
6 bar_height = df['Cylinders'].value_counts().sort_index().values
7 plt.figure(figsize=(X.width/72, X.height/72))
8 plt.bar(x=bar_x.astype(str), height=bar_height, color="#087E8B",
9 ec="#0245d4")
10 plt.title('Bar chart of CYL')
11 plt.xlabel('Number of cylinders')
12 plt.ylabel('Count')
13 plt.savefig(XImage, dpi=100)
14 plt.close()

15
16 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
17 df=pd.read_csv('c:/temp/air.csv')
18 df['DATE'] = df['DATE'].apply(lambda x: pd.to_datetime(x))
19 plt.figure(figsize=(X.width/72, X.height/72))
20 line = plt.plot(df['DATE'], df['AIR'], lw=2.5, color="#087E8B")
21 formatter = mdates.DateFormatter('%Y')
22 locator = mdates.YearLocator()
23 plt.title('Line chart of Airline passengers')
24 plt.xlabel('Year')
25 plt.ylabel('Count')
26 plt.savefig(XImage, dpi=100)
27 plt.close()

28
29
30
```

```
PS C:\Users\tedco> python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exec(open('C://temp//playground//CodeToRun.py').read())
```



CodeToRunDone.txt



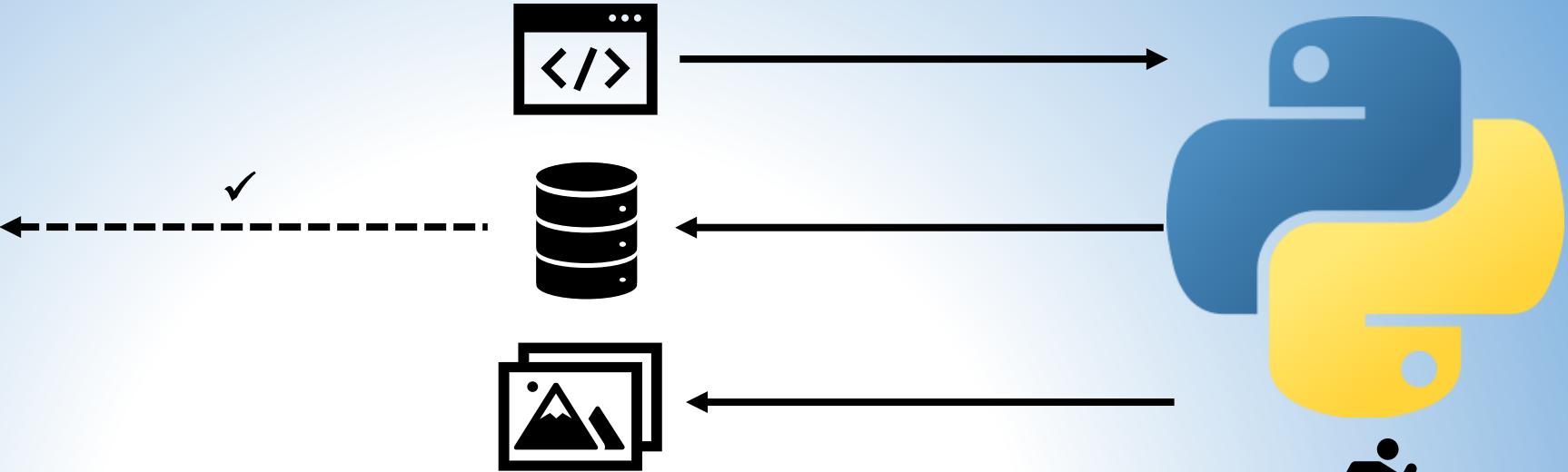
Python script starts & Excel waits for trigger file indicating complete

The screenshot shows a Windows desktop environment. In the center, there is a blue rectangular overlay containing the text "Python script starts & Excel waits for trigger file indicating complete". Below this overlay, there are three windows:

- Excel Window:** The title bar says "PyDataGlobalDemo.xlsx". The spreadsheet contains several rows of code, primarily using the Matplotlib library to generate plots from CSV files. The code includes imports for pandas, matplotlib.pyplot, and matplotlib.dates, along with various plot types like bar charts and scatter plots.
- Windows PowerShell Window:** The title bar says "PS C:\Users\stedco>". It shows the command "python" being run, followed by the output of a Python script that reads a file named "CodeToRunDone.txt".
- Code Editor Window:** The title bar says "MATPLOTLIB". It displays the same code as the Excel sheet, which is used to generate the plots shown in the Excel spreadsheet.

The taskbar at the bottom shows icons for various applications including File Explorer, Edge, and Powerpoint. The system tray indicates the date and time as 12/3/2022 and 12:51 AM, with a weather status of 50°F Clear.

```
1 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
2 df=pd.read_csv('c:/temp/carsdata.csv')
3 bar_counts = df['Cylinders'].value_counts().sort_index()
4 bar_x = df['Cylinders'].value_counts().sort_index().index
5 bar_height = df['Cylinders'].value_counts().sort_index().values
6 plt.figure(figsize=(X.width/72, X.height/72))
7 plt.bar(x=bar_x.astype(str), height=bar_height, color="#087E8B",
8 ec="#0245d4")
9 plt.title('Bar chart of CYL')
10 plt.xlabel('Number of cylinders')
11 plt.ylabel('Count')
12 plt.savefig (XLimage, dpi=100)
13 plt.close()
14
15
16 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
17 df=pd.read_csv('c:/temp/air.csv')
18 df['DATE']=pd.date_range(pd.datetime(0, pd.to_datetime(x)))
19 plt.figure(figsize=(X.width/72, X.height/72))
20 line = plt.plot(df['DATE'], df['ARR'], lw=2.5, color="#087E8B")
21 formatter = mdates.DateFormatter('%Y')
22 locator = mdates.YearLocator()
23 plt.title('Line chart of Airline passengers')
24 plt.xlabel('Year')
25 plt.ylabel('Count')
26 plt.savefig (XLimage, dpi=100)
27
28
29
30
```



Python code runs, producing output chart images and trigger file

The screenshot displays a dual-monitor setup. The left monitor shows a Jupyter Notebook interface with two code cells. The first cell contains code for creating a bar chart of car cylinder counts, and the second cell contains code for creating a scatter plot of HP vs MPG. The right monitor shows a Windows PowerShell window running Python 3.7.1, which is executing a script named 'CodeToRun.py'.

```
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c:/temp/carsdata.csv')
bar_counts = df['Cylinders'].value_counts().sort_index()
bar_x = df['Cylinders'].value_counts().sort_index().index
bar_height = df['Cylinders'].value_counts().sort_index().values
plt.figure(figsize=(X.width/72, X.height/72))
plt.bar(x=x.astype(str), height=bar_height, color="#087E8B", ec="#0245d4")
plt.title('Bar chart of CYL')
plt.xlabel('Number of cylinders')
plt.ylabel('Count')
plt.savefig(XImage, dpi=100)
plt.close()

import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c:/temp/air.csv')
df['DATE'] = df['DATE'].apply(lambda x: pd.to_datetime(x))
line = plt.plot(df['DATE'], df['AIR'], lw=2.5, color="#087E8B")
formatter = mdates.DateFormatter('%Y')
locator = mdates.YearLocator()
plt.title('Line chart of Airline passengers')
plt.xlabel('Year')
plt.ylabel('Count')
plt.savefig(XImage, dpi=100)
plt.close()
```

```
PS C:\Users\tedco> python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exec(open('C://temp//playground//CodeToRun.py').read())
>>> -
```



Pychart1-N.png



Excel resumes, retrieves images & places them into cells over code

AutoSave (Off) PyDataGlobalDemo.xlsx Search (Alt+Q)

File Home Insert Draw Page Layout Formulas Data New Tab Review View Developer Help New Tab

fx import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates

Select Windows PowerShell
PS C:\Users\stedco> python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exec(open('C:/temp/Playground//CodeToRun.py').read())
>>>

Bar chart of CYL

Scatter plot of HP vs MPG

Boxplot of CYL vs MPG

Line chart of Airline passengers

Histogram of MPG

Count Miles per gallon Number of cylinders

Count Year

Count MPG

DEMO MATPLOTLIB PLOTLTY NY TIMES KNOCKOFF

Rain to stop 12:43 AM 12/3/2022

Windows 10 Taskbar: Earnings upcoming, Rain to stop, Accessibility: Investigate, etc.



A LOOK AT THE CODE

H
I
A
B
V
E
C
E
X
E

```
'==> Use Windows 32 or 64-bit routines to sleep for # of milliseconds
#If VBA7 Then
    Public Declare PtrSafe Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As LongPtr) ' Sleep funcion (msec) for 64 bit systems
#Else
    Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)           ' Sleep funcion (msec) for 32 bit systems
#End If

'==> Create Excel charts by calling Python with cell width/height & code from each tagged comment/cell
Public Sub DrawCharts()

    DeleteCharts                                         ' Get rid of any existing charts

    pmdir = Sheets("Setup").Range("A2")
    vmdir = Sheets("Setup").Range("A4")
    Open pmdir & "CodeToRun.py" For Output As #1
    i = 0
    For Each cmt In ActiveSheet.Comments
        If UCase(Left(cmt.Text, 7)) = "#PYTHON" Then
            i = i + 1
            Print #1, "XLwidth=" & cmt.Parent.MergeArea.Width & Constants.vbCrLf &
                      "XLheight=" & cmt.Parent.MergeArea.Height & Constants.vbCrLf &
                      "XLimage=" & vmdir & "Pythonchart" & i & "." & Sheets("Setup").Range("A9") & "''" & Constants.vbCrLf &
                      Replace(cmt.Parent.Value, Chr(10), vbCrLf) & Constants.vbCrLf & cmt.Text
            Debug.Print cmt.Parent.Value
            On Error Resume Next
            Kill pmdir & "Pythonchart" & i & "." & Sheets("Setup").Range("A9")
            On Error GoTo 0
        End If
    Next cmt
    imagesexpected = i                                     ' Save number of images expected
    Print #1, "Xlf=open('" & vmdir & "CodeToRunDone.txt','w')" & Constants.vbCrLf &
              "Xlf.write('Done')" & Constants.vbCrLf &
              "Xlf.close()"
    Close #1

    On Error Resume Next
    Kill pmdir & "CodeToRunDone.txt"                     ' Delete existing trigger file (used to indicate Python is done)
    On Error GoTo 0

    Sleep (1000)                                         ' Activate Python and run code

    AppActivate Sheets("Setup").Range("A15"), True
    Sleep (1000)                                         ' Sleep for 1 second
    SendKeys "exec{()open{()}'" & vmdir & "CodeToRun.py'{()}.read{()}{}{ENTER}}", True;

    i = 0
    While (Dir$(pmdir & "CodeToRunDone.txt") = "")      ' Wait for Python code to finish
        Sleep (1000)
        i = i + 1
        If i > Sheets("Setup").Range("A13") Then
            w = MsgBox("Still unfinished, wait?", vbYesNo)
            If w = vbNo Then Exit Sub
            i = 0
        End If
    Wend
    ' Timeout after specified # of seconds
    ' After specified # of seconds, ask if user wants to continue or bail

```

EXCEL VBA

```
i = 0
For Each cmt In ActiveSheet.Comments
    If UCASE(Left(cmt.Text, 7)) = "#PYTHON" Then
        i = i + 1
        If Dir$(pcdir & "Pythonchart" & i & "." & Sheets("Setup").Range("A9")) <> "" Then
            Set newshape = ActiveSheet.Shapes.AddPicture(pcdir & "Pythonchart" & i & "." & Sheets("Setup").Range("A9"), False, True, _
                cmt.Parent.Left, cmt.Parent.Top, cmt.Parent.MergeArea.Width, cmt.Parent.MergeArea.Height)
            newshape.AlternativeText = "*Python;"
            imagesinserted = imagesinserted + 1
        End If
    End If
Next cmt
wbname = Split(ActiveWorkbook.Name, ".")
AppActivate wbname(LBound(wbname)) ' "Excel"
' Make sure we got expected # of images, issue error if not
If imagesexpected <> imagesinserted And Left(UCASE(Sheets("Setup").Range("A17")), 1) = "Y" Then
    i = MsgBox("CHECK Python LOG: ONLY " & imagesinserted & " OF " & imagesexpected & " IMAGES INSERTED", vbOKOnly, "CODE PLAYGROUND ALERT")
End If

End Sub

'==> Delete any existing Python-generated charts
Sub DeleteCharts()
C = ActiveSheet.Shapes.Count
For i = C To 1 Step -1
    If ActiveSheet.Shapes(i).AlternativeText = "*Python;" Then ActiveSheet.Shapes(i).Delete
Next
End Sub

'==> Hide any existing Python-generated charts
Sub HideCharts()
C = ActiveSheet.Shapes.Count
For i = C To 1 Step -1
    If ActiveSheet.Shapes(i).AlternativeText = "*Python;" Then ActiveSheet.Shapes(i).Visible = False
Next
End Sub

'==> Show any existing Python-generated charts
Sub ShowCharts()
C = ActiveSheet.Shapes.Count
For i = C To 1 Step -1
    If ActiveSheet.Shapes(i).AlternativeText = "*Python;" Then ActiveSheet.Shapes(i).Visible = True
Next
End Sub
```

thinkinsidethebox/README.md  + 

github.com/tedconway/thinkinsidethebox/blob/main/README.md

 Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

 tedconway / thinkinsidethebox Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

 main  thinkinsidethebox / README.md Go to file ...

 tedconway Update README.md Latest commit 634d299 43 seconds ago  History

 1 contributor

COMING SOON!

2 lines (2 sloc) | 141 Bytes   Raw Blame   

thinkinsidethebox

PyData Global 2022 Lightning Talk Materials Coming Soon! ([Abstract](#))

“All models are wrong, but some are useful.”

– George Box

“All VBA tricks are stupid, but some are useful.”

– Ted Conway



QUESTIONS ?

Thank you!

Ted Conway
ted.j.conway@gmail.com
@vivasasvegas