



# Think Inside the Box(es)

---

Excel-Hosted Dashboards With  
Python Graphics

Lightning Talks – Ted Conway





# TODAY'S GAME PLAN

---

# 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

+

Filter files by name

/ notebooks /

Name	Last Modified
Intro.ipynb	a month ago
Lorenz.ipynb	a month ago
sqlite.ipynb	a month ago
Untitled.ipynb	2 minutes ago

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

JUPYTER HOSTED

2:53 PM 12/2/2022

▶ 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

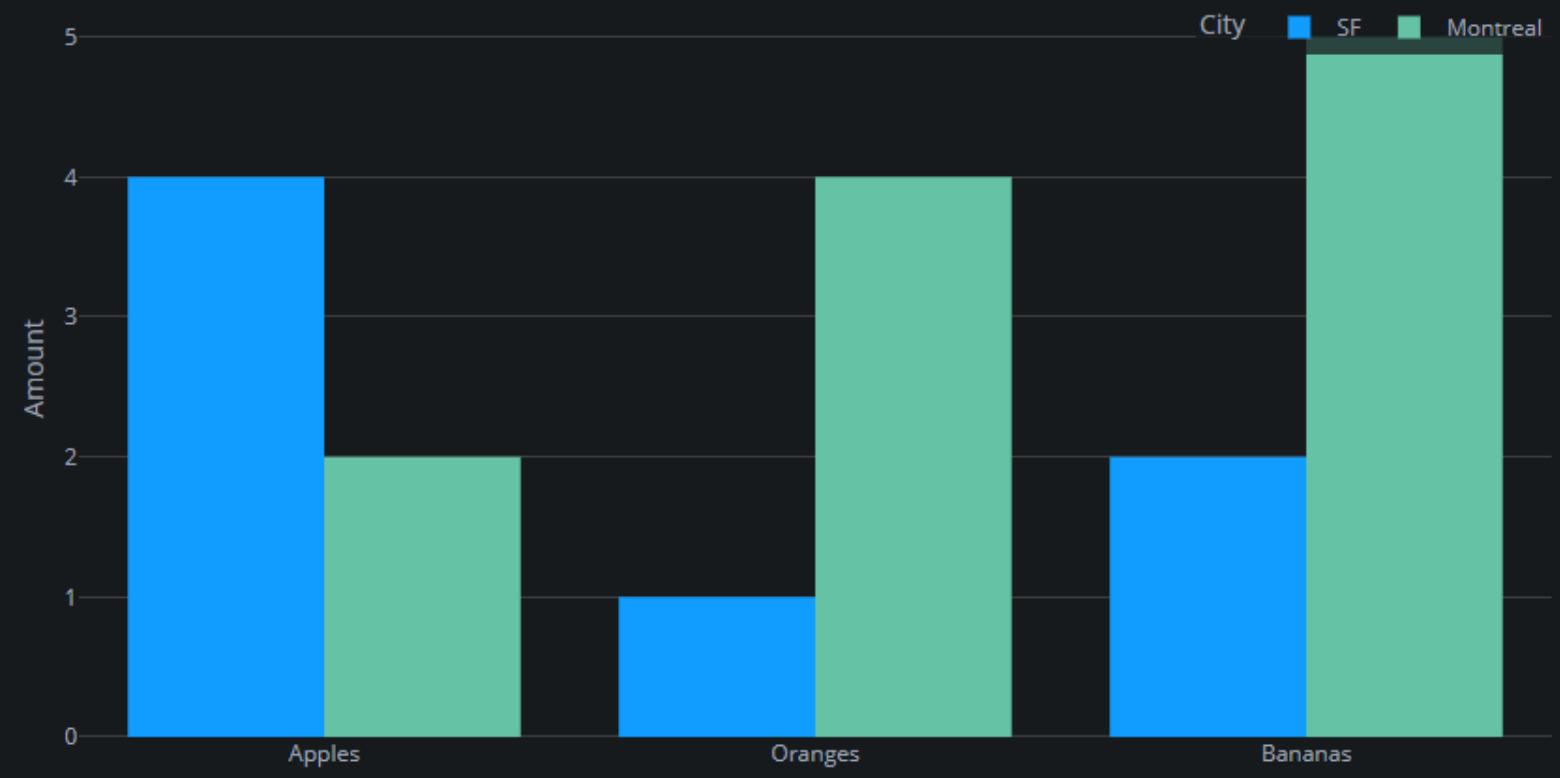
▶ Beyond the Basics

▶ 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



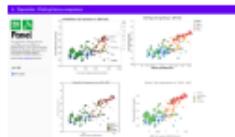
# Panel

## A high-level app and dashboarding solution for Python

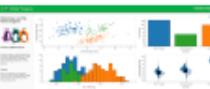
Attractors



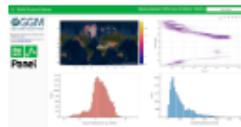
Gapminders



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

bill length mm

22

■

Wind

9:22 PM  
12/2/2022

QUESTION  
MARK  
IN  
EXCEL  
EXCEL

WHY  
NOT  
W

## 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

A black and white photograph capturing a moment of childhood play. In the foreground, a young girl with dark hair tied back is seen from behind, wearing a light-colored striped dress. She is climbing a metal jungle gym, her hands gripping a horizontal bar and her legs pulled up towards her. To her right, another child's legs are visible, also engaged with the equipment. In the background, a boy in a striped shirt and shorts is hanging from a higher level of the structure. A girl in a light dress stands further back, watching the scene. The playground is made of dark metal pipes and is set against a backdrop of trees and a fence.

# 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

A B C D E F

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()

1  
2  
3

XLheight

XLwidth

#PYTHON

Toolbar Buttons

① Draw Charts

② Delete Charts

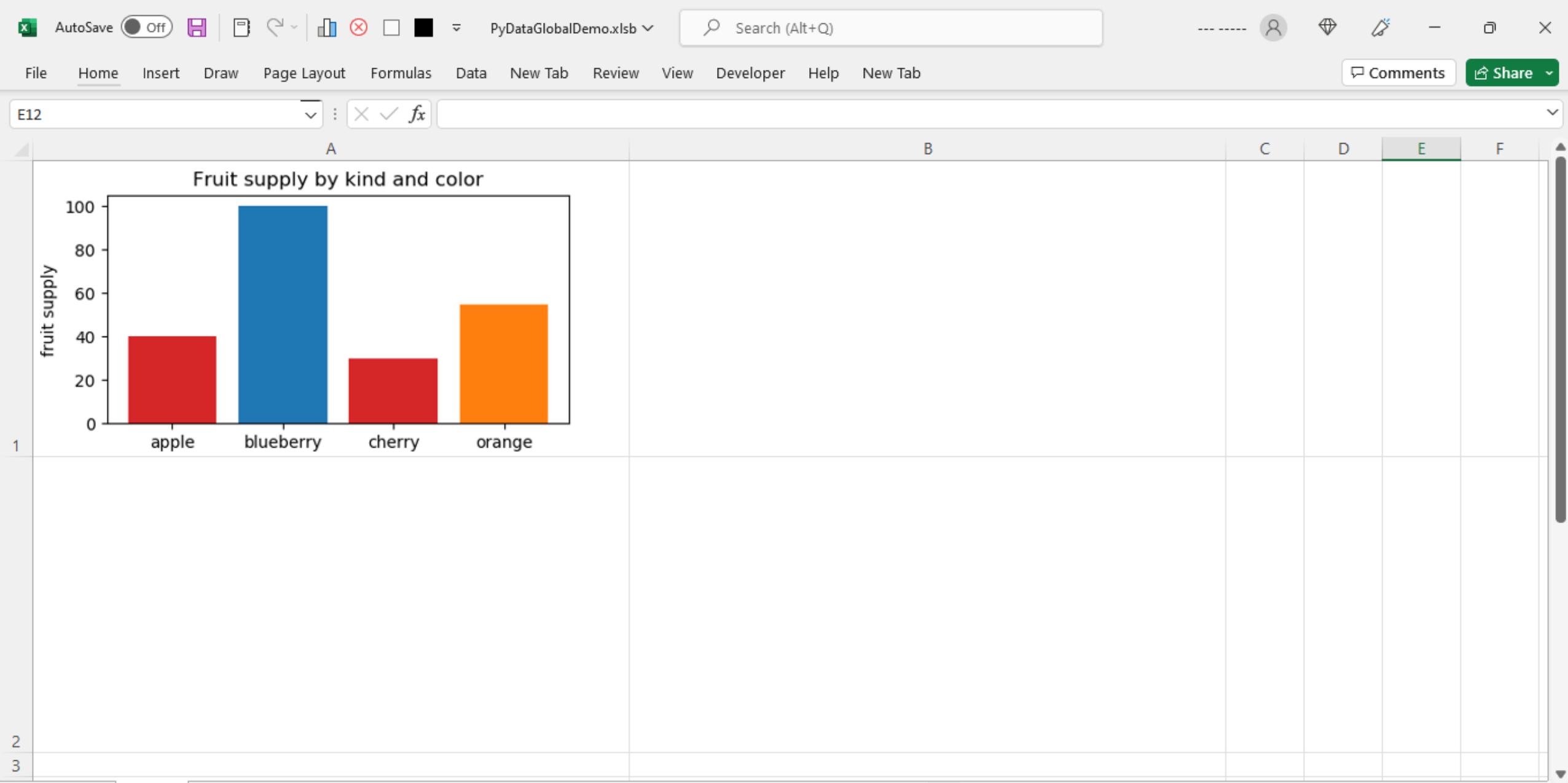
③ Show Charts

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

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()
```

```
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()
```

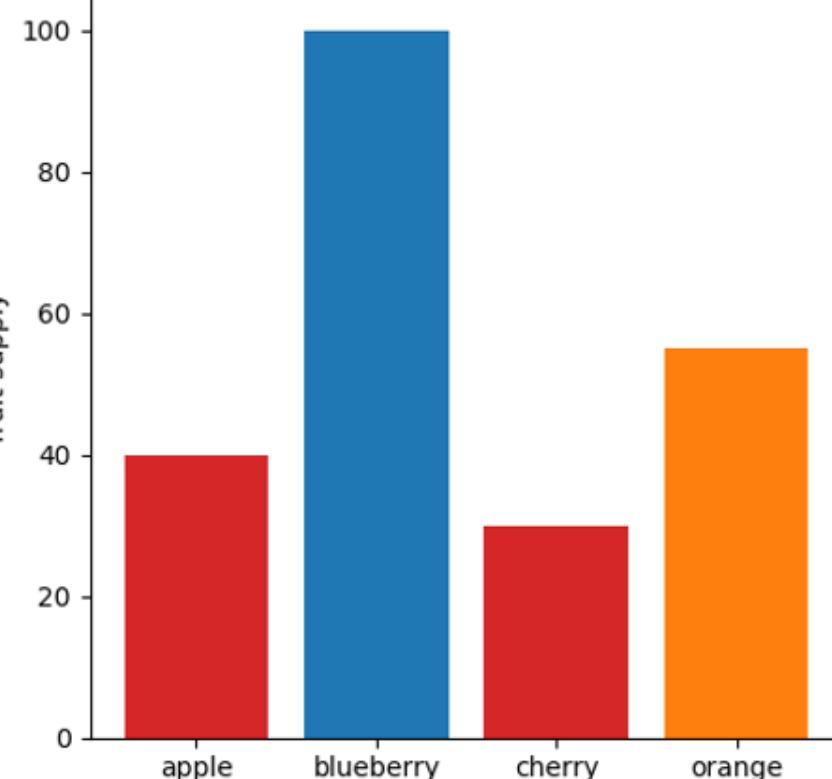
```
import matplotlib.pyplot as plt
# Use special variables XLwidth and XLheight to make image cell-sized
fig, ax = plt.subplots(figsize=(XLwidth/72., XLheight/72.))
fruits = ['apple', 'blueberry', 'cherry', 'orange']
counts = [40, 100, 30, 55]
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']
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

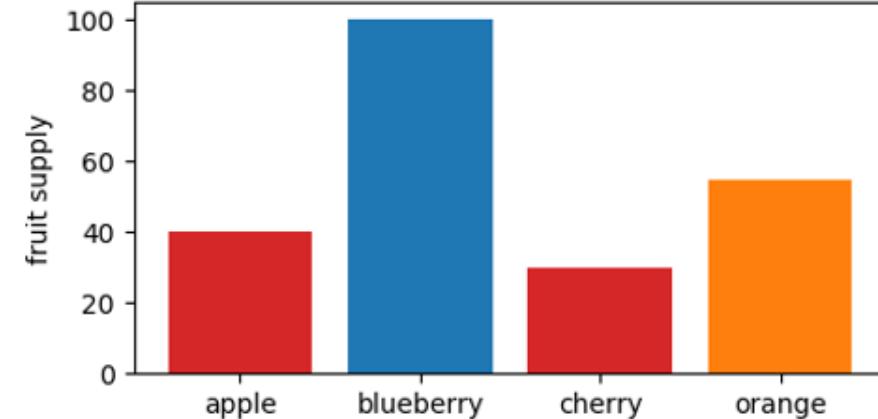
COPY CODE  
OR USE  
FORMULAS

CHARTS CAN  
SPAN CELLS

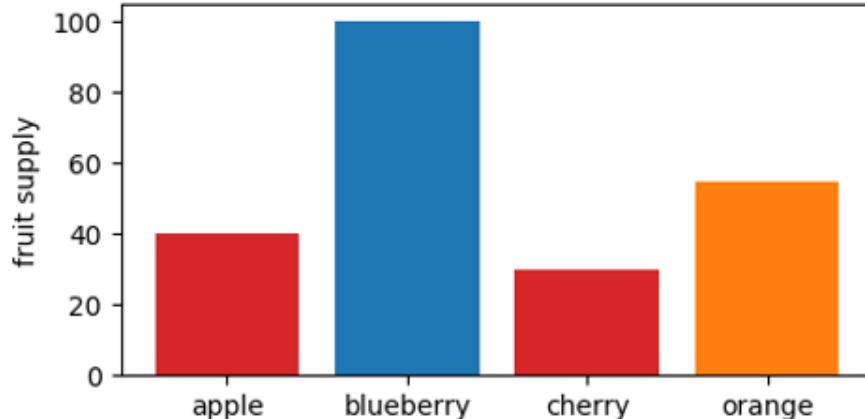
Fruit supply by kind and color



Fruit supply by kind and color

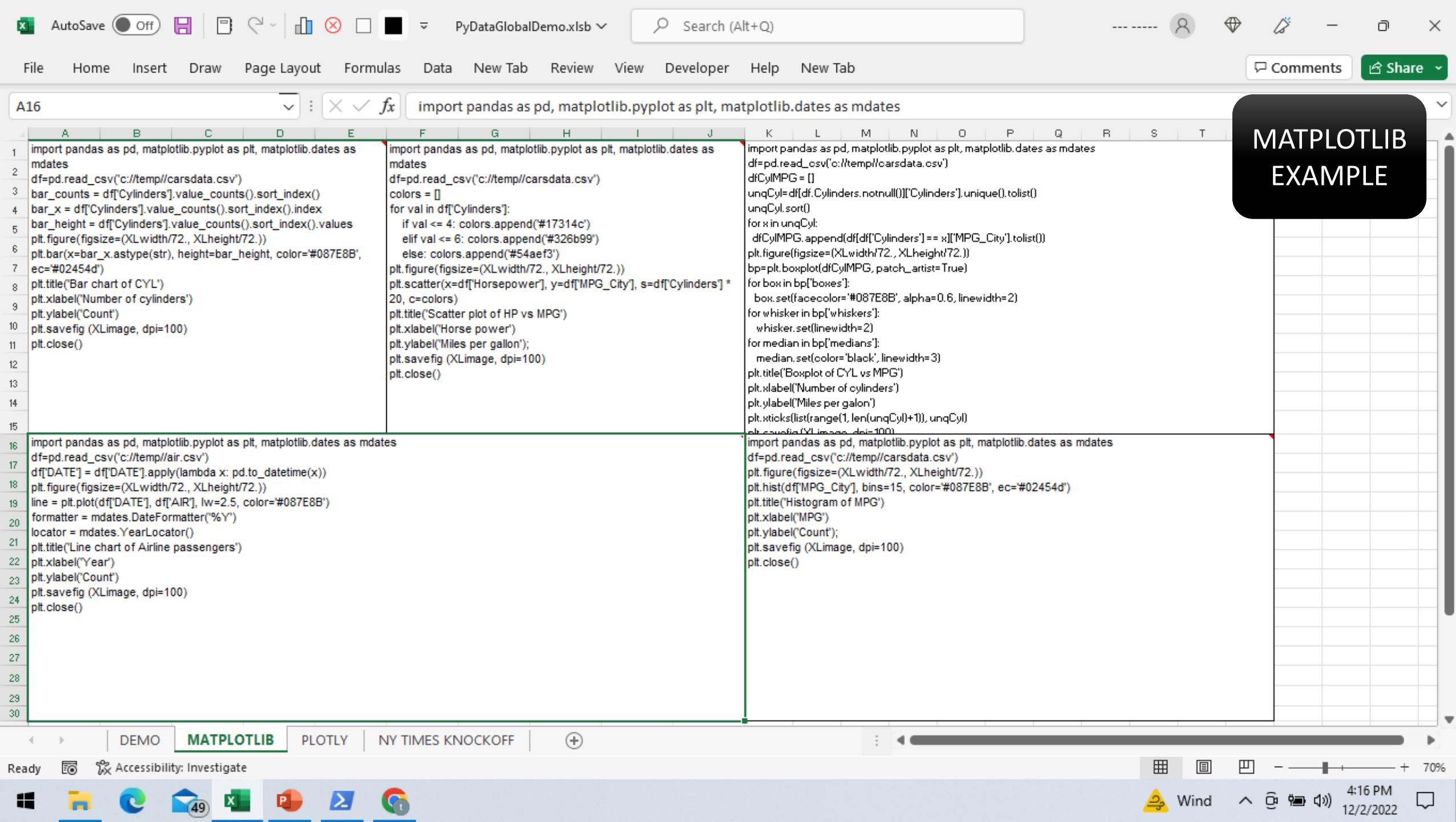


Fruit supply by kind and color



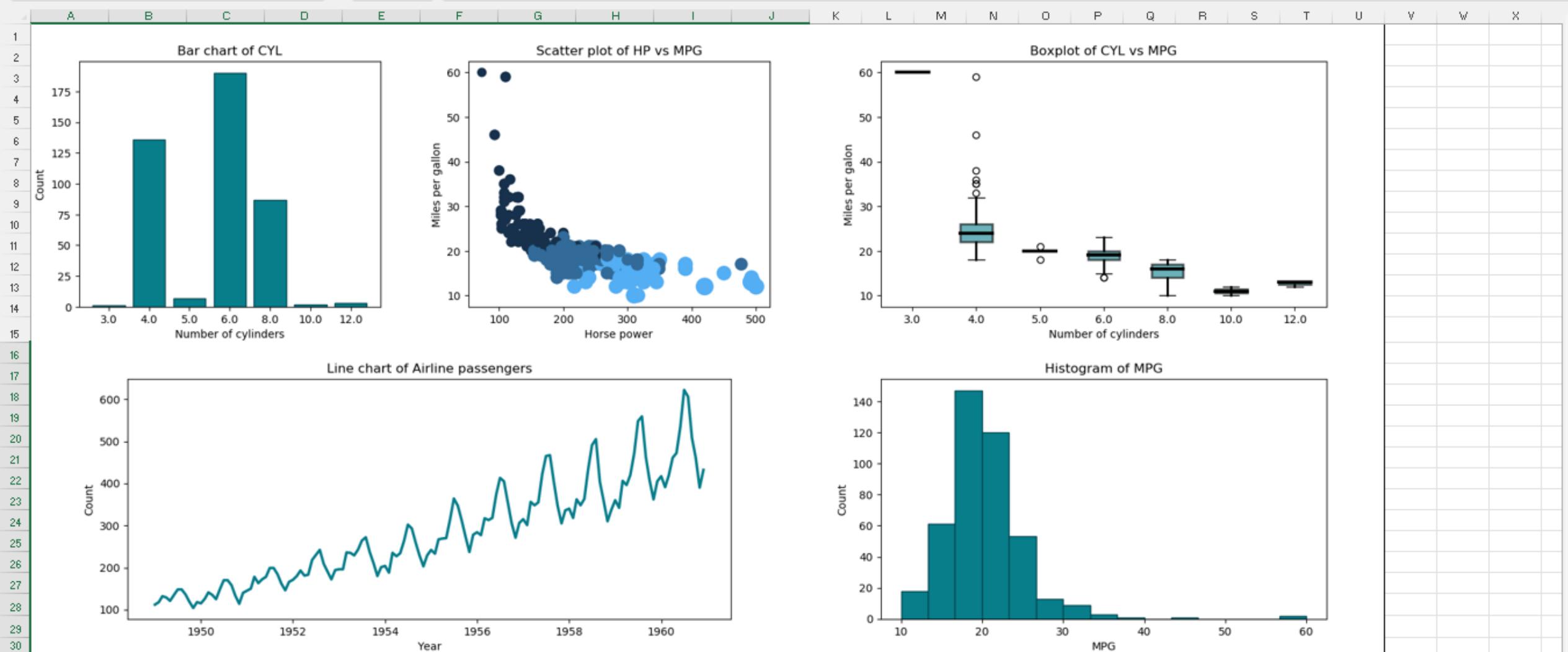
# DASHBOARDS





A16

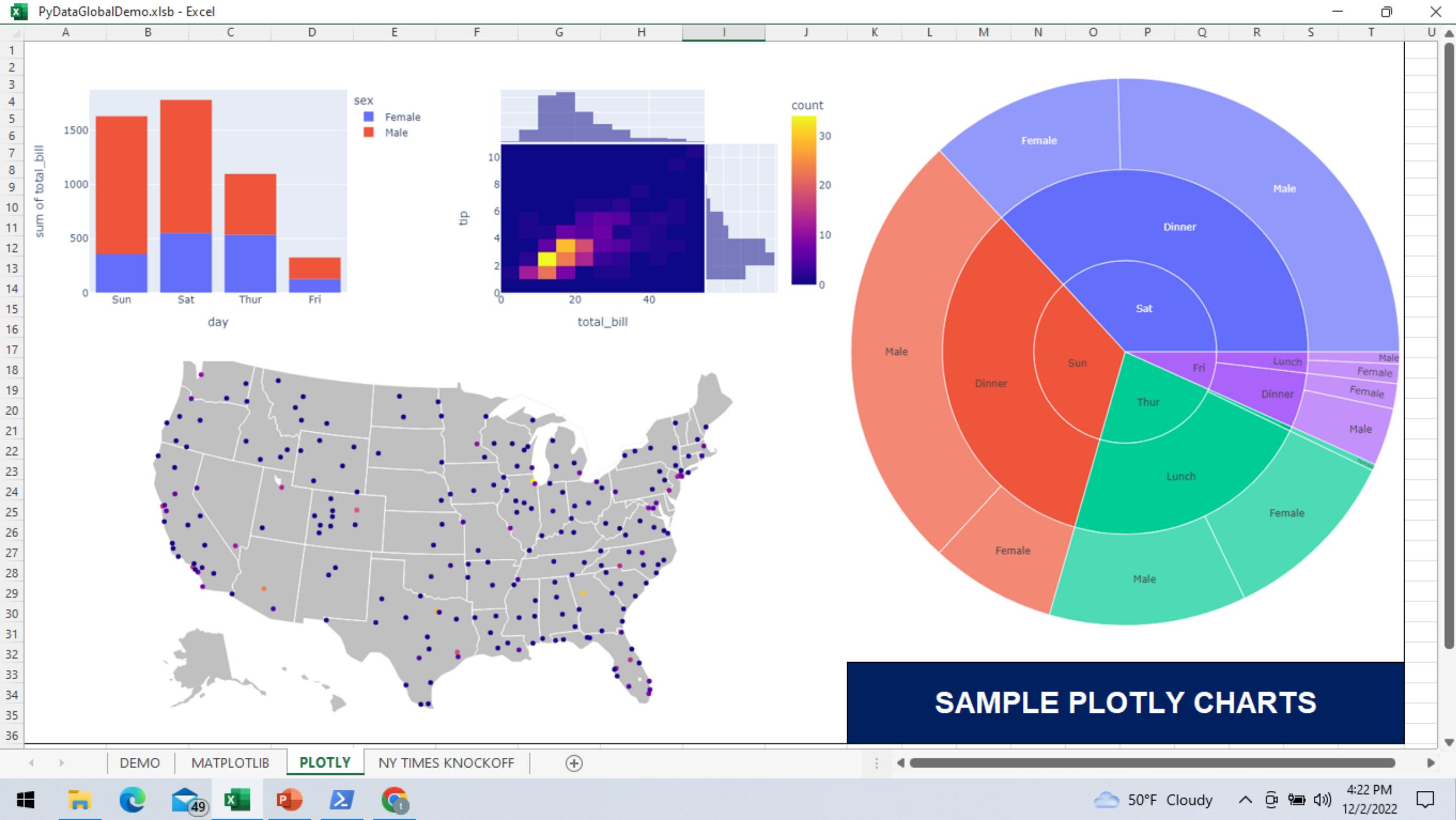
fx import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates



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
```

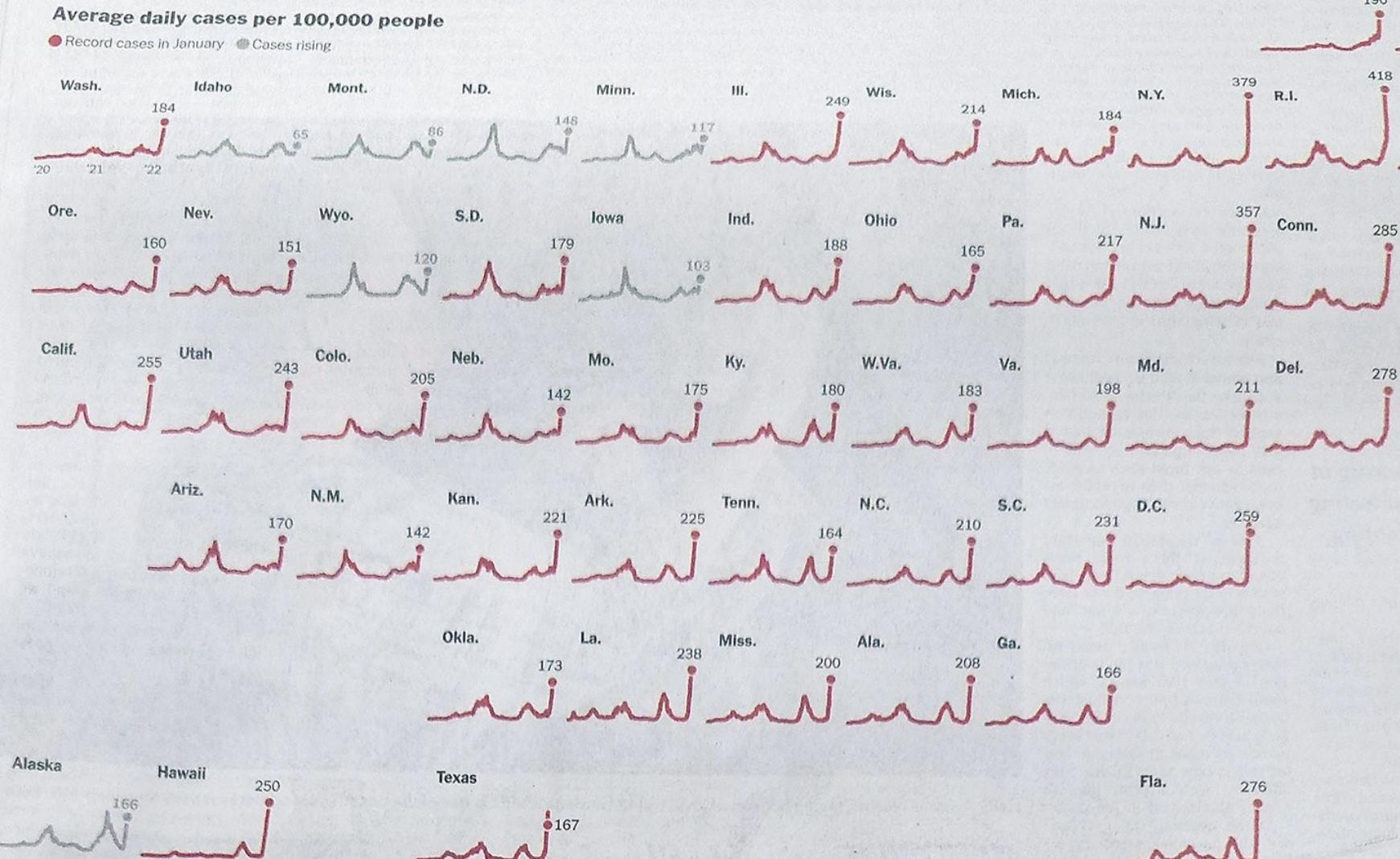
SAMPLE PLOTLY CHARTS



# What to Make of Those Soaring Covid Counts

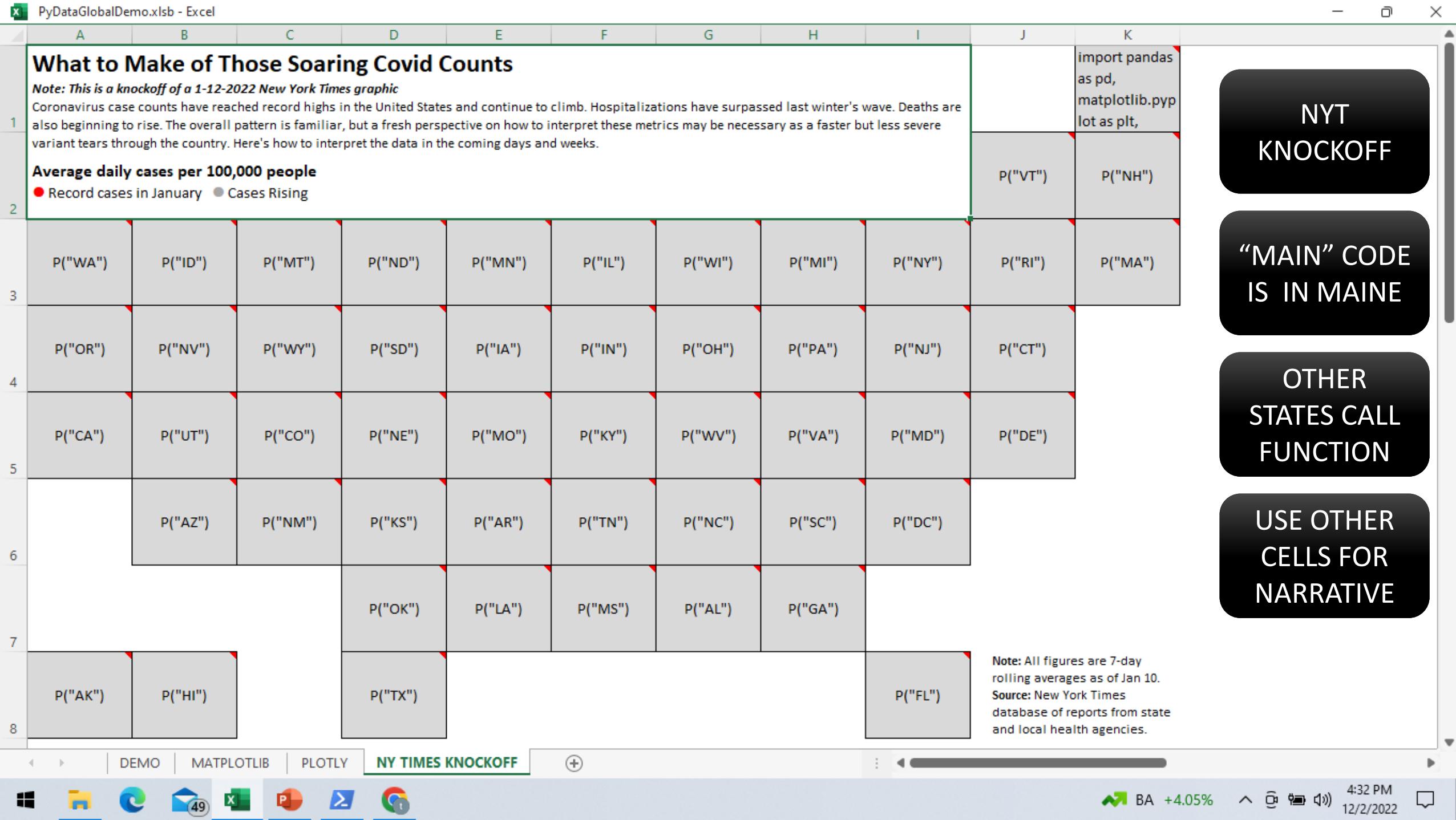
This article is by Lazario Gamio, Lisa Waananen Jones and Amy Schoenfeld Walker.

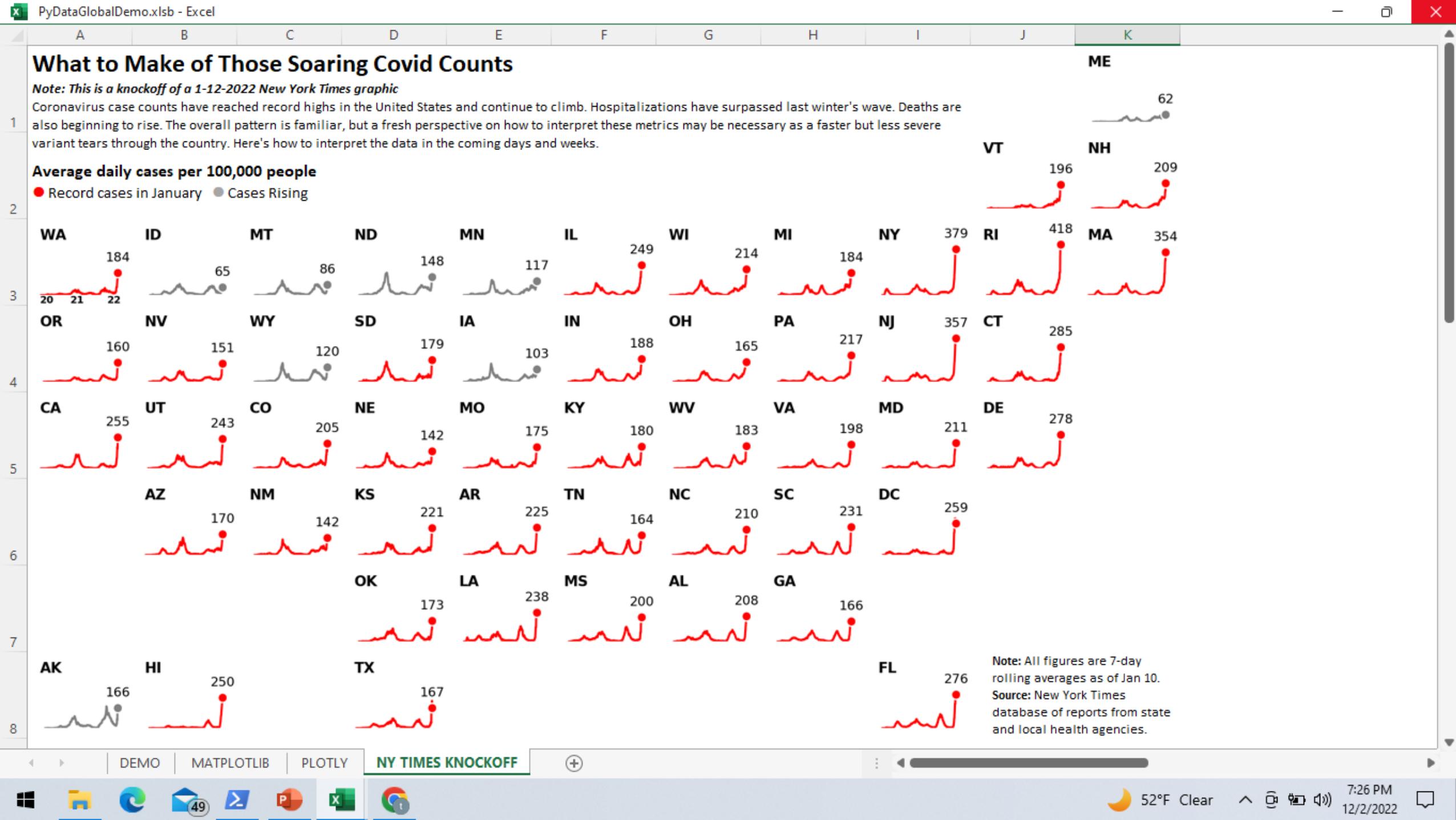
Coronavirus case counts have reached record highs in the United States and continue to climb. Hospitalizations have surpassed last winter's wave. Deaths are also beginning to rise. The overall pattern is familiar, but a fresh perspective on how to interpret these metrics may be necessary as a faster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

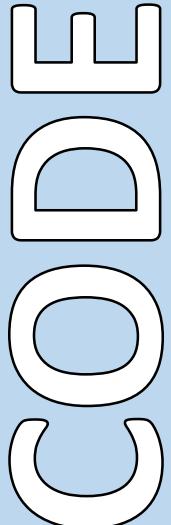


Note: All figures are seven-day rolling averages as of Jan. 10.  
Source: New York Times database of reports from state and local health agencies

THE NEW YORK TIMES







```
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
    # State code in upper left corner
    plt.text(0, 1, df["st"].iloc[-1], ha='left', va='top', transform=ax.transAxes, color='black', fontsize=10, weight="bold") # Ending number of cases over last point
    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') # Line plot of cases/100K of population,
    red/gray colors denote case trend
    line = plt.plot(df['date'], df['cases_avg_per_100k'], lw=1.5, color=df['trend'].iloc[-1]) # Marker surrounded by whitespace for ending #
of cases
    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)
    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.
>>>
```

AutoSave Off | File Home Insert Draw Page Layout Formulas Data New Tab Review View Developer Help New Tab Search (Alt+Q) Comments Share

A1 : import matplotlib.pyplot as plt

A	B	C	D	E	F
---	---	---	---	---	---

```
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

DEMO PLOTLY MATPLOTLIB NY TIMES KNOCKOFF

Ready Accessibility: Investigate





# STEP-BY-STEP



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



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

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

Comments Share

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.
>>>
```

A B C D E F G H I J K L M N O P Q R S T U V W X

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 Cylinders')
10 plt.xlabel('Number of cylinders')
11 plt.ylabel('Count')
12 plt.savefig (XLimage, dpi=100)
13 plt.close()

14

15 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
16 df=pd.read\_csv('c:/temp/air.csv')
17 df['DATE']=pd.date\_range(pd.datetime(0, 0, 0), pd.to\_datetime(x))
18 plt.figure(figsize=(X.width/72, X.height/72))
19 line = plt.plot(df['DATE'], df['ARR'], lw=2.5, color="#087E8B")
20 formatter = mdates.DateFormatter('%Y')
21 locator = mdates.YearLocator()
22 plt.title('Line chart of Airline passengers')
23 plt.xlabel('Year')
24 plt.ylabel('Count')
25 plt.savefig (XLimage, dpi=100)
26 plt.close()

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

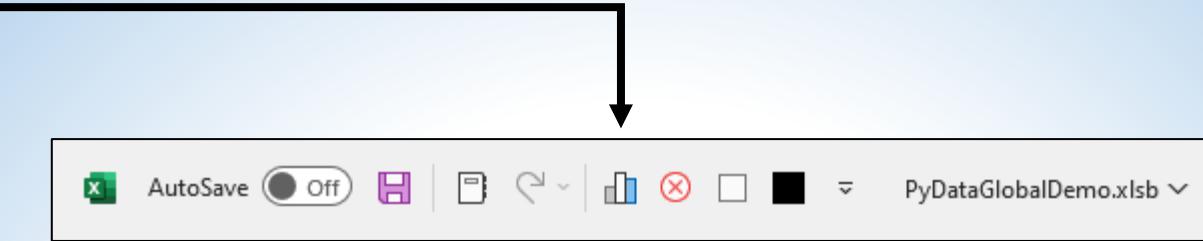
1284

1285

1286

1287

12



Excel toolbar button is pressed to run chart generation macro

PyDataGlobalDemo.xlsb

```
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 Cylinders')
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 plt.figure(figsize=(X.width/72., X.height/72.))
19 line = plt.plot(df['DATE'], df['ARR'], lw=2.5, color="#087E8B")
20 formatter = mdates.DateFormatter('%Y')
21 locator = mdates.YearLocator()
22 plt.title('Line chart of Airline passengers')
23 plt.xlabel('Year')
24 plt.ylabel('Count')
25 plt.savefig (XLimage, dpi=100)
26
27
28
29
30
```

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.
>>>
```

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

Comments Share

Search (Alt+Q)

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

Ready Accessibility: Investigate

MATPLOTLIB NY TIMES KNOCKOFF

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



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

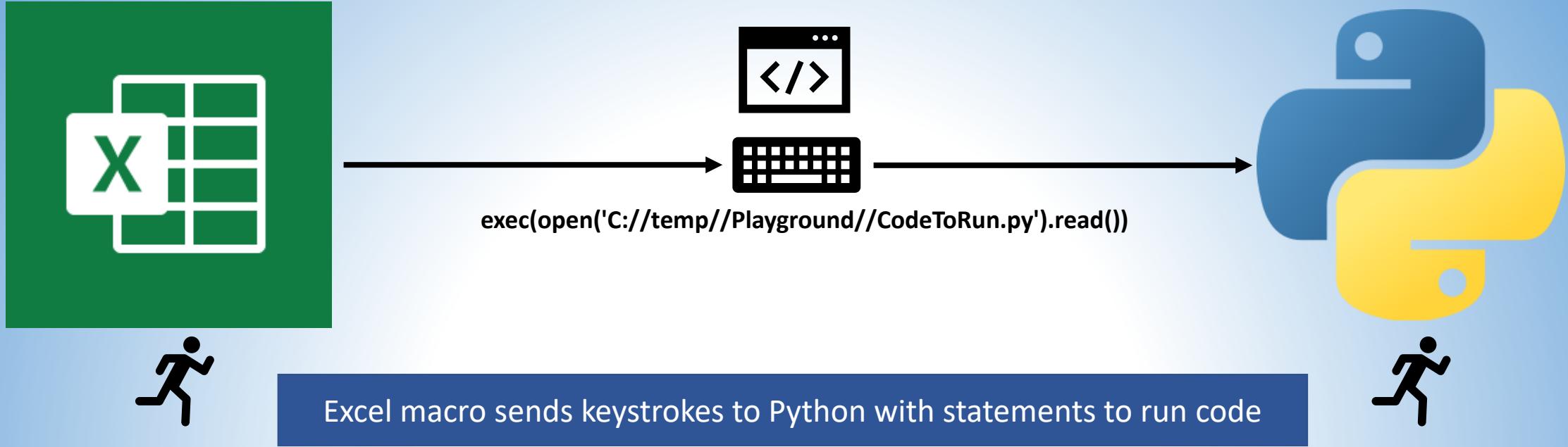
```
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
df=pd.read_csv('c:/temp/carsdata.csv')
bar_counts=df['Cylinders'].value_counts().sort_index()
bar_x=bar_counts.index
bar_height=bar_counts.sort_index().values
plt.figure(figsize=(Xwidth/72, Xheight/72))
plt.bar(x=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(XImage, dpi=100)
plt.close()

import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
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')
colors = []
for val in df['Cylinders']:
    if val <= 4: colors.append('#17314c')
    elif val <= 6: colors.append('#326b99')
    else: colors.append('#54aeff')
plt.figure(figsize=(Xwidth/72, Xheight/72))
bp=plt.boxplot(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/air.csv')
dt[DATE] = dt[DATE].apply(lambda x: pd.to_datetime(x))
plt.figure(figsize=(Xwidth/72, Xheight/72))
line = plt.plot(dt[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()

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
>>>
```





## 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()

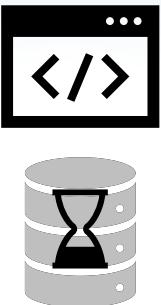
15
16 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
17 df=pd.read_csv('c:/temp/air.csv')
18 df['DATE'] = df['DATE'].apply(lambda x: pd.to_datetime(x))
19 plt.figure(figsize=(X.width/72, X.height/72))
20 line = plt.plot(df['DATE'], df['AIR'], lw=2.5, color="#087E8B")
21 formatter = 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 &amp; Excel waits for trigger file indicating complete

The screenshot shows a Microsoft Excel spreadsheet titled "PyDataGlobalDemo.xlsx" with a code editor open. The code editor contains two snippets of Python code:

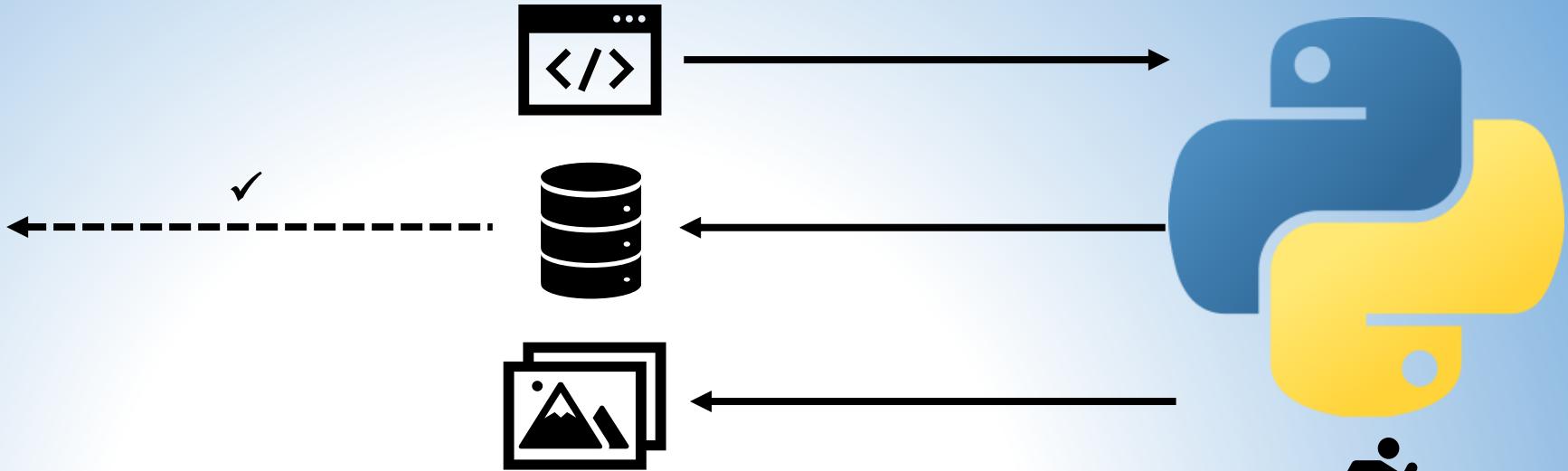
```
1 import pandas as pd, matplotlib.pyplot as plt, matplotlib.dates as mdates
2 df=pd.read_csv('c:/temp/carsdata.csv')
3 bar_counts = df['Cylinders'].value_counts().sort_index()
4 bar_x = df['Cylinders'].value_counts().sort_index().index
5 bar_height = df['Cylinders'].value_counts().sort_index().values
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.dateutil.parser.parse(df['DATE']).dt.date
19 df['DATE']=df['DATE'].dt.strftime('%Y-%m-%d')
20 locator = dates.YearLocator()
21 formatter = dates.DateFormatter('%Y')
22 plt.title('Line chart of Airline passengers')
23 plt.xlabel('Year')
24 plt.ylabel('Count')
25 plt.savefig (XLimage, dpi=100)
26
27
28
29
30
```

Below the code editor is a Windows PowerShell window showing the command being run:

```
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())
```

The Excel spreadsheet has tabs labeled "DEMO", "MATPLOTLIB", "PLOTTLY", and "NY TIMES KNOCKOFF". The status bar at the bottom shows "Ready", "Rain to stop", "12:35 AM 12/3/2022", and "70%". The taskbar at the bottom includes icons for File Explorer, Google Chrome, Microsoft Edge, and others.



Python code runs, producing output chart images and trigger file

```
AutoSave Off PyDataGlobalDemo.xlsx Search (Alt+Q) File Home Insert Draw Page Layout Formulas Data New Tab Review Developer Help New Tab
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.dateutil.parser.parse(df['DATE'])
19 df['DATE']=df['DATE'].dt.strftime('%Y-%m-%d')
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 plt.close()
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59 Select 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.
>>> exec(open('C:/temp/Playground//CodeToRun.py').read())
>>>

```



Pychart1-N.png



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

AutoSave (Off) PyDataGlobalDemo.xlsxv 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 12:47 AM 12/3/2022

Windows Earnings upcoming 12/3/2022

A black and white photograph of three young boys, two African American and one Caucasian, looking through Spencer binocular microscopes. They are all wearing plaid shirts. The background is slightly blurred.

A LOOK AT THE CODE

H  
I  
A  
B  
V  
E  
U  
C  
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                                 ' Delete existing trigger file (used to indicate Python is done)
    Kill pmdir & "CodeToRunDone.txt"
    On Error GoTo 0

    Sleep (1000)                                       ' Activate Python and run code

    AppActivate Sheets("Setup").Range("A15"), True
    Sleep (1000)                                       ' Activate Python and submit code via SendKeys function
    SendKeys "exec{()open{()}'" & vmdir & "CodeToRun.py'{()}.read{()}{}{ENTER}}", True;

    i = 0
    While (Dir$(pcdir & "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

    ' Sleep for 1 second
    ' Timeout after specified # of seconds
    ' After specified # of seconds, ask if user wants to continue or bail
End Sub
```

2  
I  
A  
B  
V  
E  
U  
C  
X  
E

```
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