

From multi-format CAD (BIM) data into a structured format 😊

```
 1 import os, subprocess  
 2  
 3 # Folder where the DDC converter is located  
 4 path_conv = r'C:\DDC_Revit_Community\datadrivenlibs\\'  
 5 # Path address RVT | IFC | DWG project are located  
 6 file_path = r'C:\DDC\rstadvanced_sample.rvt'  
 7  
 8 # Conversion of one RVT project  
 9 process = subprocess.Popen([os.path.join(path_conv,  
10 'RvtExporter.exe'), file_path], cwd=path_conv)  
11  
12 print("DDC Conversion process finished")
```

DATA CONVERSION TO OPEN FORMATS



conversion in just 4
lines of code

data-driven
construction.io

```

1 # RVT | IFC | DWG project file name in XLSX format
2 output_file = file_path[:-4] + "_rvt.xlsx"
3 # Read the converted Excel file
4 df = pd.read_excel(output_file)
5 # Update column names to remove storage type in parameter
6 df.columns = [col.split(' : ')[0] for col in df.columns]

```

two-dimensional
project data

data-driven
construction.io

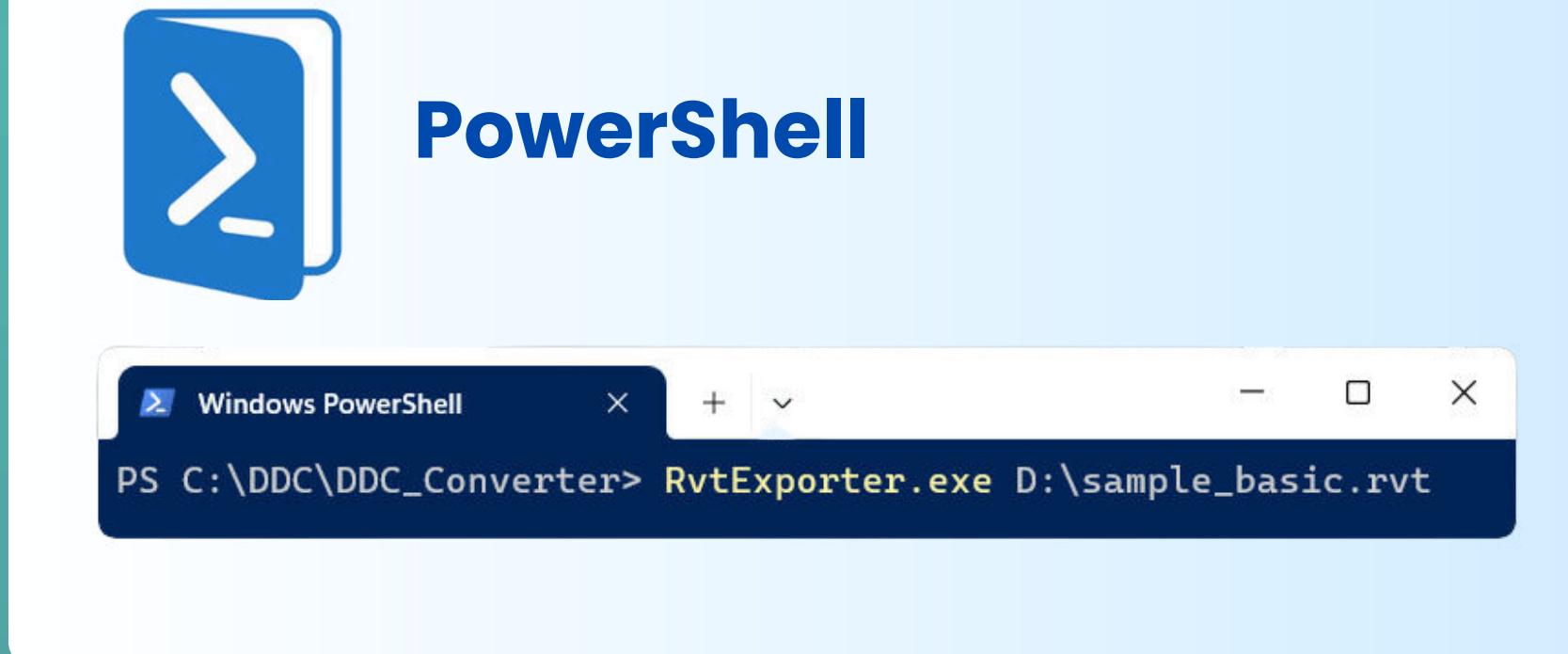
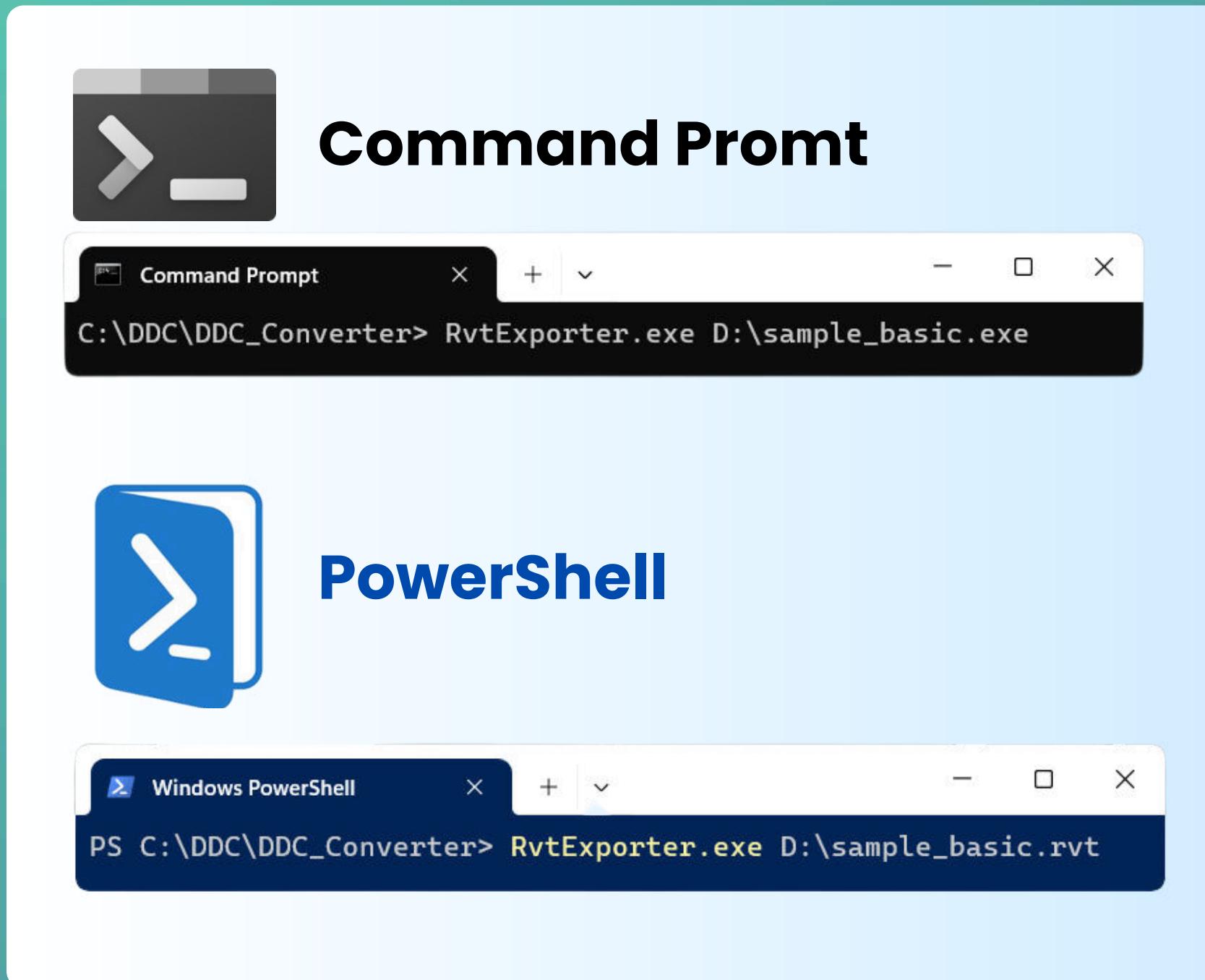
🚀 Structured format is ideal
for analytics, visualization
and automation

STRUCTURED AS DATA

ID	Name	Category	Family Name	Height	BoundingBoxMin_X	BoundingBoxMin_Y	BoundingBoxMin_Z	Level
431144	Single-Flush	OST_Doors	Single-Flush	6.88976378	20.1503	-10.438	9.84252	Level 1
431198	Single-Flush	OST_Doors	Single Window	6.88976378	13.2281	-1.1207	9.84252	Level 2
457479	Single Window	OST_Windows		8.858267717	-11.434	-11.985	9.80971	Level 2
485432	Single Window	OST_Windows	Single Window	8.858267717	-11.434	4.25986	9.80971	Level 2
490150	Single-Flush	OST_Doors	Single-Flush	6.88976378	-1.5748	-2.9565	-1E-16	
493697	Basic Wall	OST_Walls	Basic Wall	-38.15	-38.15	20.1656	-4.9213	Level 1
497540	Basic Wall	OST_Walls	Basic Wall		-4.5212	-0.0708	9.84252	Level 1
								Level 1

Converter

terminal version



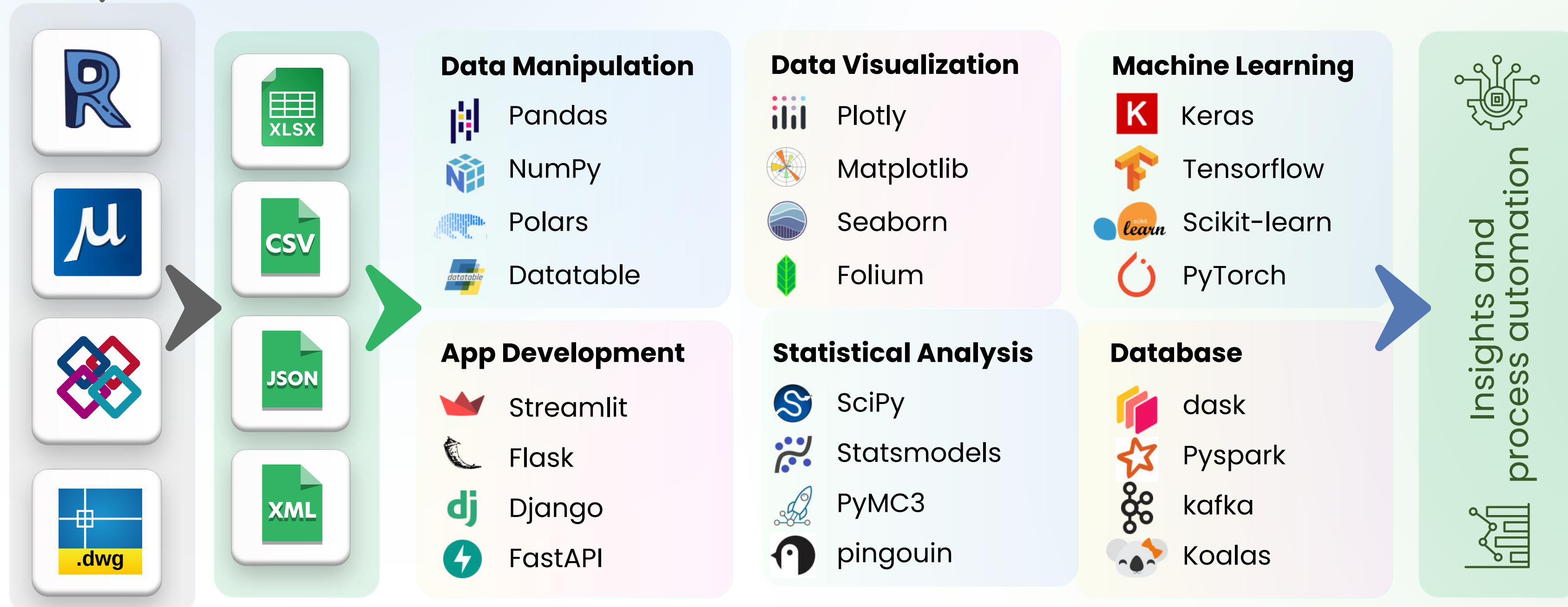
Hundreds of applications allow you to embed the conversion process into your use cases



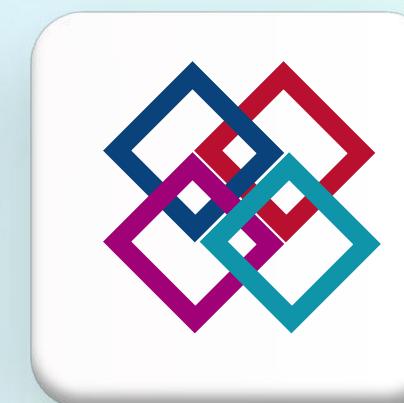
Life Is Short, Use Python

to work with construction project data

data-driven
construction.io

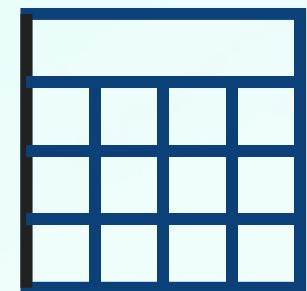


easy to learn, easy to develop



STRUCTURED DATA

AS



The diagram illustrates a Pandas DataFrame with various annotations:

- Column names:** Points to the column headers: ID, Name, Category, Family Name, Height, BoundingBoxMin_X, BoundingBoxMin_Y, BoundingBoxMin_Z, and Level.
- Index label:** Points to the index label "431144" in the first row.
- Index axis = 0:** Points to the index axis, labeled "axis = 0".
- Missing value:** Points to an empty cell in the "Family Name" column for the second row.
- Data:** Points to the value "8.858267717" in the "Height" column for the third row.
- Level:** Points to the "Level" column, which contains values like "Level 1" and "Level 2".
- Level 1:** Points to the rows where the "Level" column value is "Level 1".
- Level 2:** Points to the rows where the "Level" column value is "Level 2".

ID	Name	Category	Family Name	Height	BoundingBoxMin_X	BoundingBoxMin_Y	BoundingBoxMin_Z	Level
431144	Single-Flush	OST_Doors	Single-Flush	6.88976378	20.1503	-10.438	9.84252	Level 1
431198	Single-Flush	OST_Doors		6.88976378	13.2281	-1.1207	9.84252	Level 2
457479	Single Window	OST_Windows	Single Window	8.858267717	-11.434	-11.985	9.80971	Level 2
485432	Single Window	OST_Windows	Single Window	8.858267717	-11.434	4.25986	9.80971	Level 2
490150	Single-Flush	OST_Doors	Single-Flush	6.88976378	-1.5748	-2.9565	-1E-16	Level 1
493697	Basic Wall	OST_Walls	Basic Wall		-38.15	20.1656	-4.9213	Level 1
497540	Basic Wall	OST_Walls	Basic Wall		-4.5212	-0.0708	9.84252	Level 1



STRUCTURED DATA

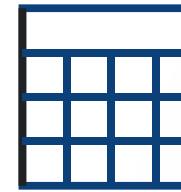
ID	Name	Category	version	proj	site	Parent	ObjectType
2	34_0001	IfcProject	IFC2X3				ÿ
3	38274	Default					
4	36_ÿ	IfcBuilding	IFC2X3	0001	Default	Default	ÿ
5	39_Level	IfcBuildingStorey	IFC2X3	0001	Default	Level 1	Basic Wall:Exterior - Brick on B
6	3797_Basic_Wall:Exterior - Brick on Block:1380	IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Exterior - Brick on B
7	3999_Basic_Wall:Exterior - Brick on Block:1381	IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Exterior - Brick on B
8	4043_Basic_Wall:Exterior - Brick on Block:1382	IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Exterior - Brick on B
9	4087_Basic_Wall:Exterior - Brick on Block:1383	IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Exterior - Brick on B
10	4131_Basic_Wall:Interior - Partition (92mm Stu IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Interior - Partition (92mm Stu IfcWallStandardC	
11	4219_Basic_Wall:Interior - Partition (92mm Stu IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Interior - Partition (92mm Stu IfcWallStandardC	
12	4287_Basic_Wall:Party_Wall - CMU Residential IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Party_Wall - CMU Residential IfcWallStandardC	
13	4399_Basic_Wall:Party_Wall - CMU Residential IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Party_Wall - CMU Residential IfcWallStandardC	
14	4465_Basic_Wall:Party_Wall - CMU Residential IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Party_Wall - CMU Residential IfcWallStandardC	
15	4508_Basic_Wall:Interior - Partition (92mm Stu IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Interior - Partition (92mm Stu IfcWallStandardC	
16	4553_Basic_Wall:Interior - Partition (92mm Stu IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Interior - Partition (92mm Stu IfcWallStandardC	
17	4598_Basic_Wall:Interior - Partition (92mm Stu IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Interior - Partition (92mm Stu IfcWallStandardC	
18	5165_Floor:127mm_Slab on Grade:141232	IfcSlab	IFC2X3	0001	Default	Level 1	Floor:127mm_Slab on Grade:141232
19	5267_Floor:127mm_Slab on Grade:141201	IfcSlab	IFC2X3	0001	Default	Level 1	Floor:127mm_Slab on Grade:141201
20	5642_Basic_Wall:Interior - Partition (92mm Stu IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Interior - Partition (92mm Stu IfcWallStandardC	
21	5903_Basic_Wall:Interior - Partition (92mm Stu IfcWallStandardC	IFC2X3	0001	Default	Level 1	Basic Wall:Interior - Partition (92mm Stu IfcWallStandardC	
22	6426_M_Fixed:4835mm x 2420mm:4835mm x IfcWindow	IFC2X3	0001	Default	Level 1	M_Fixed:4835mm x 2420mm:4835mm x IfcWindow	
23	6531_M_Fixed:4835mm x 2420mm:4835mm x IfcWindow	IFC2X3	0001	Default	Level 1	M_Fixed:4835mm x 2420mm:4835mm x IfcWindow	
24	6652_M_Single-Flush:1250mm x 2010mm:125IfcDoor	IFC2X3	0001	Default	Level 1	M_Single-Flush:1250mm x 2010mm:125IfcDoor	
25	6757_M_Single-Flush:1250mm x 2010mm:125IfcDoor	IFC2X3	0001	Default	Level 1	M_Single-Flush:1250mm x 2010mm:125IfcDoor	
26	6921_M_Fixed:750mm x 2200mm:750mm x 22 IfcWindow	IFC2X3	0001	Default	Level 1	M_Fixed:750mm x 2200mm:750mm x 22 IfcWindow	
27	7029_M_Elbow:750mm x 2200mm:750mm x 22 IfcWindow	IFC2X3	0001	Default	Level 1	M_Elbow:750mm x 2200mm:750mm x 22 IfcWindow	



STRUCTURED DATA

ID	Name	Category	Design	IfcGUID	Type IfcGUID	Family and Type
29	198363_Single_Window	OST_Windows	None	3Lx0gNe59vvExhbv0Bfj2		
30	198367_Basic_Wall	OST_Walls	None	3Lx0gNe59vvExhbv0Bfj3		
31	198369_Finishes - Interior - Plaster	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj2		
32	198370_Wood - Stud Layer	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
33	198372_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
34	198373_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
35	198374_Finishes - Exterior - Timbr	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
36	198365_Window - PVC Coating	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj7		
37	198366_Single_Window	OST_Windows	None	3Lx0gNe59vvExhbv0Bfj2		
38	198367_Basic_Wall	OST_Walls	None	3Lx0gNe59vvExhbv0Bfj3		
39	198368_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj2		
40	198369_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj3		
41	198370_Wood - Stud Layer	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
42	198372_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
43	198373_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
44	198374_Finishes - Exterior - Timbr	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
45	198365_Window - PVC Coating	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj7		
46	198366_Single_Window	OST_Windows	None	3Lx0gNe59vvExhbv0Bfj2		
47	198367_Basic_Wall	OST_Walls	None	3Lx0gNe59vvExhbv0Bfj3		
48	198368_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj2		
49	198369_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj3		
50	198370_Wood - Stud Layer	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
51	198372_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
52	198373_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
53	198374_Finishes - Exterior - Timbr	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
54	198365_Window - PVC Coating	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj7		
55	198366_Single_Window	OST_Windows	None	3Lx0gNe59vvExhbv0Bfj2		
56	198367_Basic_Wall	OST_Walls	None	3Lx0gNe59vvExhbv0Bfj3		
57	198368_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj2		
58	198369_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj3		
59	198370_Wood - Stud Layer	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
60	198372_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
61	198373_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
62	198374_Finishes - Exterior - Timbr	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
63	198365_Window - PVC Coating	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj7		
64	198366_Single_Window	OST_Windows	None	3Lx0gNe59vvExhbv0Bfj2		
65	198367_Basic_Wall	OST_Walls	None	3Lx0gNe59vvExhbv0Bfj3		
66	198368_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj2		
67	198369_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj3		
68	198370_Wood - Stud Layer	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
69	198372_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
70	198373_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
71	198374_Finishes - Exterior - Timbr	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
72	198365_Window - PVC Coating	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj7		
73	198366_Single_Window	OST_Windows	None	3Lx0gNe59vvExhbv0Bfj2		
74	198367_Basic_Wall	OST_Walls	None	3Lx0gNe59vvExhbv0Bfj3		
75	198368_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj2		
76	198369_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj3		
77	198370_Wood - Stud Layer	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
78	198372_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
79	198373_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
80	198374_Finishes - Exterior - Timbr	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
81	198365_Window - PVC Coating	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj7		
82	198366_Single_Window	OST_Windows	None	3Lx0gNe59vvExhbv0Bfj2		
83	198367_Basic_Wall	OST_Walls	None	3Lx0gNe59vvExhbv0Bfj3		
84	198368_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj2		
85	198369_Finishes - Interior - Plaste	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj3		
86	198370_Wood - Stud Layer	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
87	198372_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
88	198373_Structure - Timber Insulat	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
89	198374_Finishes - Exterior - Timbr	OST_Materials	None	3Lx0gNe59vvExhbv0Bfj1		
90	198365_Window - PVC Coating	OST				

STRUCTURED DATA



Pandas: The leading library for data manipulation and a key tool for building pipelines



8811040

Number of downloads of the Pandas Pipeline library each day



70%

Data engineers [using](#) Pandas Pipeline as their primary tool



200k

Questions on Stack Overflow [tagged](#) with Pandas Pipeline



LOAD

Input

Importing Revit and IFC data.py

```
1 # Importing data for processing
2
3 import pandas as pd
4 df = pd.read_csv('C:\Revit_Sample.csv')
```

Output

	Id	Category	Type	Length	Volume
0	12577	Wall	Wall WD100	3200	1.0
1	15889	Wall	Wall STB 200	5400	6.0
2	76554	Door	Glazed Back Door	1300	0.3
3	74456	Window	Window 1700w	1700	0.5



FILTER

Input

Filtering data in Revit and IFC projects.py

```
1 # Whether each element contains the values
2
3 df[df['Category'].isin(['Wall', 'Window'])]
```

Output

	Id	Category	Type	Length	Volume
0	12577	Wall	Wall WD100	3200	1.0
1	15889	Wall	Wall STB 200	5400	6.0
3	74456	Window	Window 1700w	1700	0.5



GROUP

Input

GroupBy Revit IFC.py

```
1 # Grouping a Revit or IFC project by parameters
2
3 df.groupby('Category')['Volume', 'Length'].sum()
```

Output

Category	Volume	Length
Door	0.3	1300
Wall	7.0	8600
Window	0.5	1700

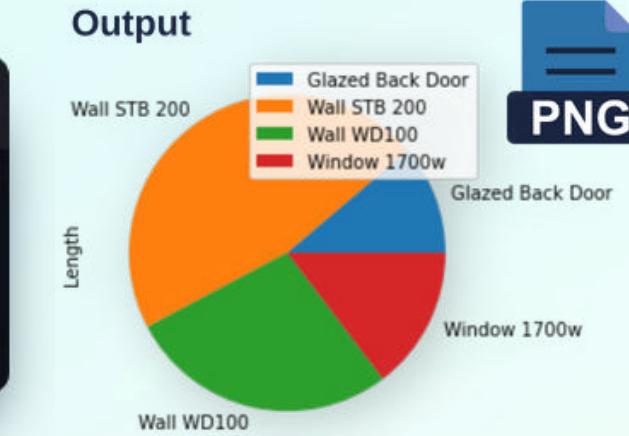


PIE chart



Input

```
- □ × Pie chart.py  
1 # Create a basic pie chart  
2  
3 df.groupby(['Type']).sum().plot.pie(y='Length')
```

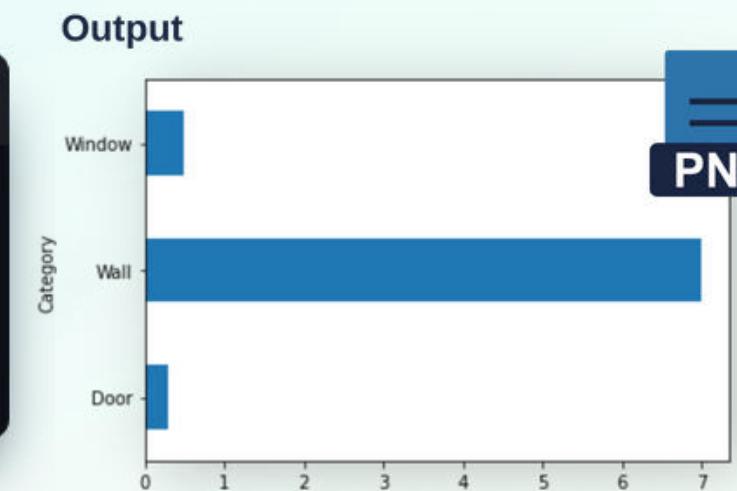


BAR chart



Input

```
- □ × Bar plot.py  
1 # The bar plot can be created as follows  
2  
3 dfp = df.groupby('Category')['Volume'].sum()  
4 dfp.plot(kind='barh')
```



Regular Expression



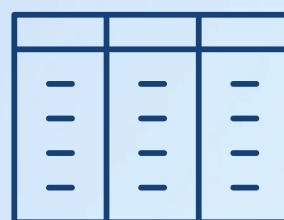
Input

```
- □ × RegEx.py  
1 #Regular expression in Revit and IFC  
2  
3 df[df['Category'].str.match('Wal*')]
```

Output

	Id	Category	Type	Length	Volume	grid icon
0	12577	Wall	Wall WD100	3200	1.0	
1	15889	Wall	Wall STB 200	5400	6.0	

QTO TakeOff



Input

```
- □ X QTO by RegEx.py  
1 #QTO - Finding volumetric quantities for the group  
2  
3 dfq = df[df['Category'].str.match('Wal*')]  
4 dfq = dfq.groupby('Category')['Volume', 'Length'].sum()
```

snappy.io

Output

Category	Volume	Length
Wall	7.0	8600

EXCEL Data Export



Input

```
- □ X Export to Excel.py  
1 # Creating a grouping and saving as Excel  
2  
3 dfe = df.groupby(['Category'])['Length'].agg(['sum', 'count'])  
4 dfe.to_excel("output.xlsx", sheet_name='Category_estimate')
```

snappy.io

Output

	A	B	C	D
2	Door	1300	1	
3	Wall	8600	2	
4	Window	1700	1	
5				

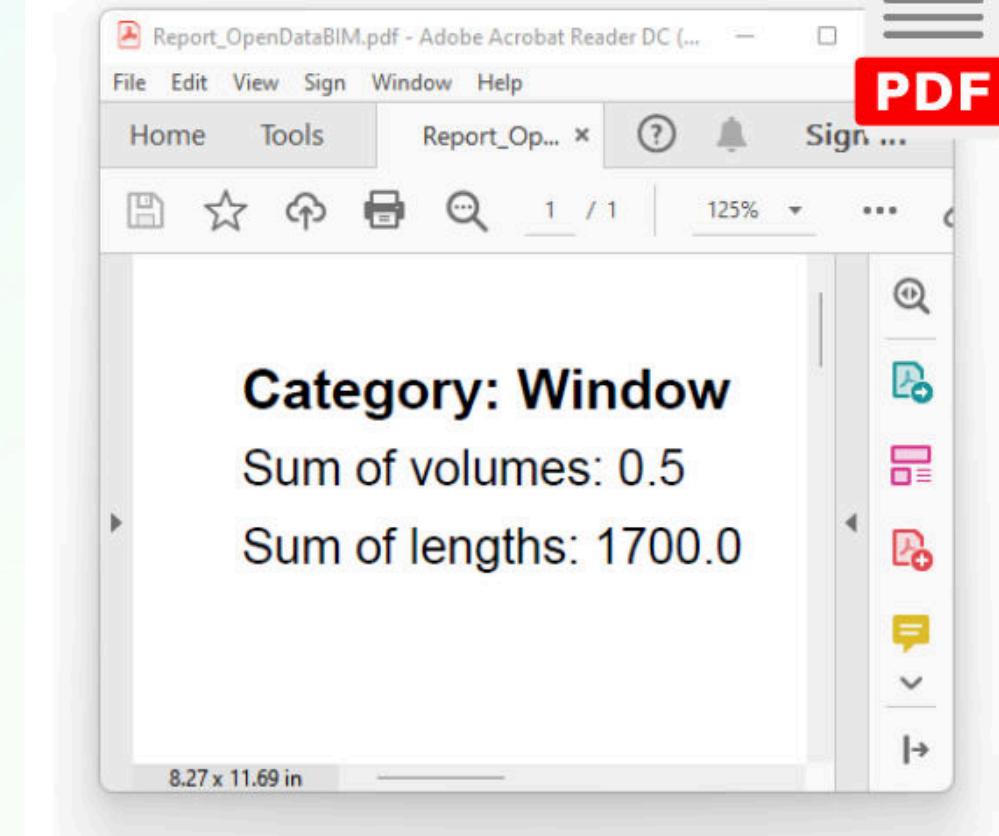
PDF Document



Input

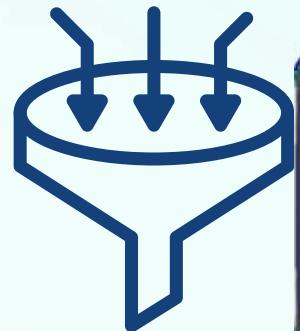
```
- □ X Creating a PDF document.py  
1 from fpdf import FPDF  
2  
3 # Determining the volumetric characteristics of the group  
4 s_cat = 'Window'  
5 dfq= df[df['Category'].str.match(s_cat)]  
6 dfq = dfq.groupby('Category')['Volume', 'Length'].sum()  
7 cat_len = str(dfq.iloc[0]['Length'])  
8 cat_vol = str(dfq.iloc[0]['Volume'])  
9  
10 # Creating a PDF document based on the parameters found  
11 pdf = FPDF()  
12 pdf.add_page()  
13 pdf.set_font('Arial', 'B', 16)  
14 pdf.cell(190, 8, 'Category: ' + s_cat, 2, 1, 'L')  
15 pdf.set_font('Arial', '', 14)  
16 pdf.cell(190, 8, 'Sum of volumes: ' + cat_vol, 2, 1, 'L')  
17 pdf.cell(190, 8, 'Sum of lengths: ' + cat_len, 2, 1, 'L')  
18  
19 # Saving a document in PDF format  
20 pdf.output('c:\Report_DataDrivenConstruction.pdf', 'F')
```

Output





FILTER



Filtering data in Revit and IFC projects.py

```
1 # Whether each element contains the values
2
3 df[df['Category'].isin(['Wall', 'Window'])]
```

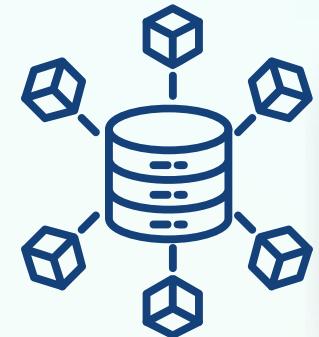
Input

	Id	Category	Type	Length	Volume
0	12577	Wall	Wall WD100	3200	1.0
1	15889	Wall	Wall STB 200	5400	6.0
3	74456	Window	Window 1700w	1700	0.5



Filter the data in the project to keep the wall category items in the project

GROUP



GroupBy Revit IFC.py

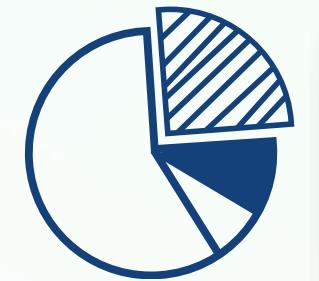
```
1 # Grouping a Revit or IFC project by parameters
2
3 df.groupby('Category')['Volume', 'Length'].sum()
```

Output

Category	Volume	Length
Door	0.3	1300
Wall	7.0	8600
Window	0.5	1700

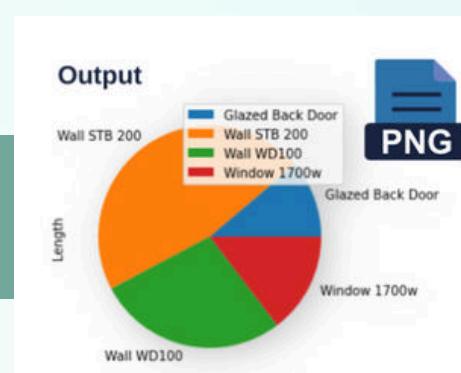
Group the project by the "Type Name" parameter and show the volume of each group

PDF



Creating a PDF document.py

```
1 from fpdf import FPDF
2
3 # Determining the volumetric characteristics of the group
4 s_cat = 'Window'
5 dfq= df[df['Category'].str.match(s_cat)]
6 dfq = dfq.groupby('Category')['Volume', 'Length'].sum()
7 cat_len = str(dfq.iloc[0]['Length'])
8 cat_vol = str(dfq.iloc[0]['Volume'])
9
10 # Creating a PDF document based on the parameters found
11 pdf = FPDF()
12 pdf.add_page()
13 pdf.set_font('Arial', 'B', 16)
14 pdf.cell(190, 8, 'Category: ' + s_cat, 2, 1, 'L')
15 pdf.set_font('Arial', '', 14)
16 pdf.cell(190, 8, 'Sum of volumes: ' + cat_vol, 2, 1, 'L')
17 pdf.cell(190, 8, 'Sum of lengths: ' + cat_len, 2, 1, 'L')
18
19 # Saving a document in PDF format
20 pdf.output('c:\Report_DataDrivenConstruction.pdf', 'F')
```



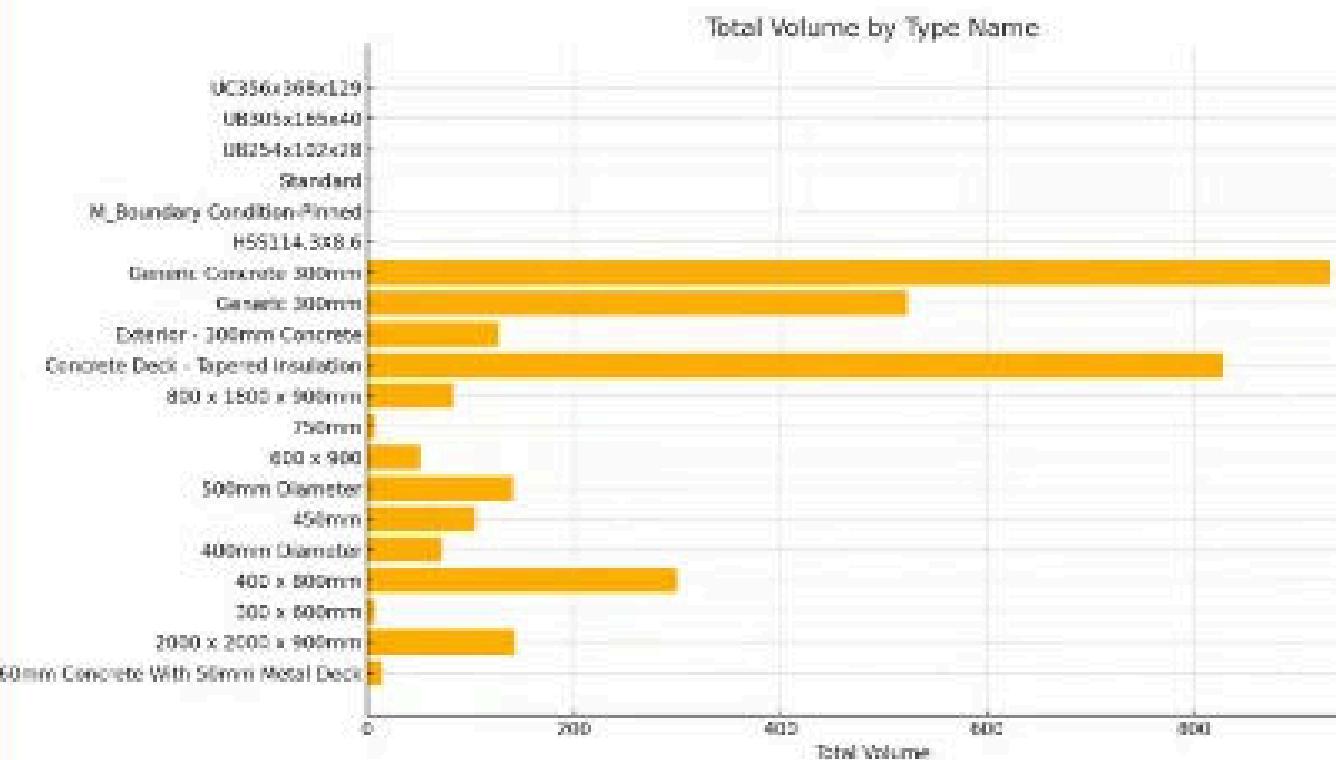
Choose the first 20 types by volume and show the result as a Pie chart



Create a PDF report with a table and a graph

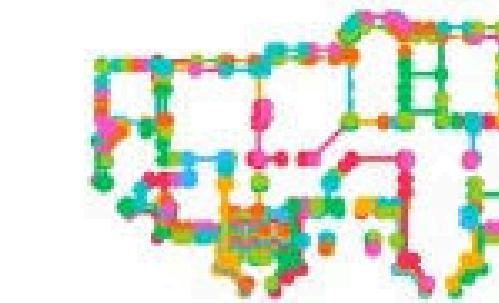
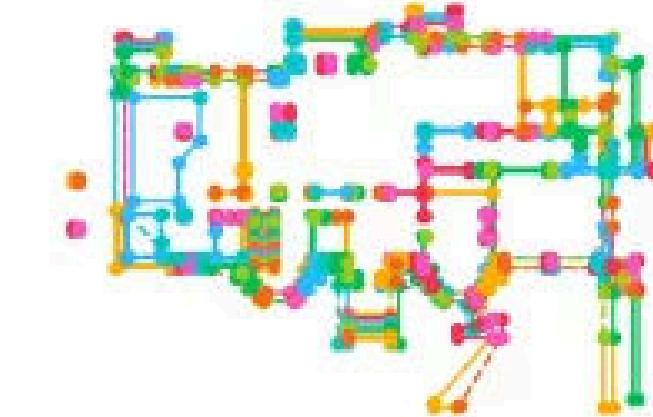
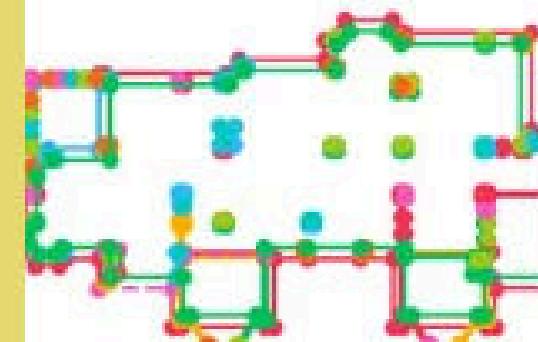
Quick QTO with graph from Revit

2022_rst_advancedsampleproject.rvt



Plot Polylines from DWG

family_house_florida.dwg



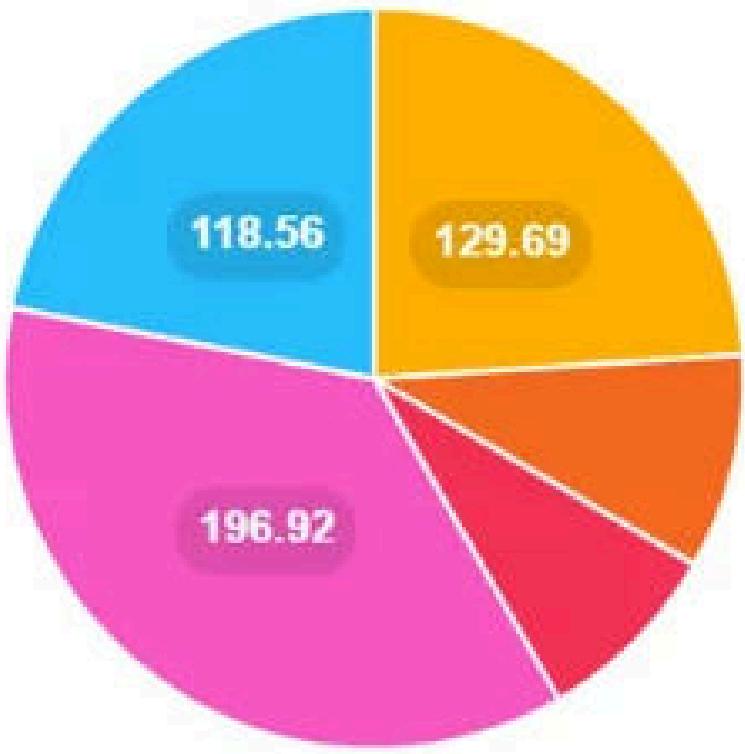
Group the data in Dataframe by "Type Name" while summarizing the "Volume" parameter and show the number of items in the group. And show it all as a horizontal bar chart without zero values



Find ids from column "Layer" with value "wall". Get this IDs and find in "ParentID". Than take for each group with "ParentID" column "Point" - x,y,z from each line. Plot separate polylines for each group based on "ParentID" and connects first and last points. Plot all lines with matplotlib without legend

Quick QTO with graph from Revit

Ifc2x3_Duplex_Architecture.ifc



Grouped Wall Data With Area from Revit

2023_rac_basic_sample_project.rvt



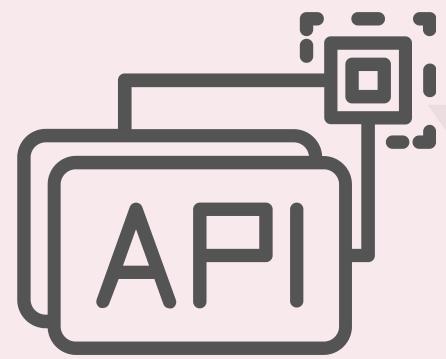
Type Name	Total Area	Count
CL_W1	393.12 sq m	10
Cavity wall_sliders	9.37 sq m	1
Foundation - 300mm Concrete	30.90 sq m	1
Interior - 165 Partition (1-hr)	17.25 sq m	3
Interior - Partition	186.54 sq m	14
Retaining - 300mm Concrete	195.79 sq m	10
SH_Curtain wall	159.42 sq m	9
SIP 202mm Wall - conc clad	114.76 sq m	4
Wall - Timber Clad	162.91 sq m	8



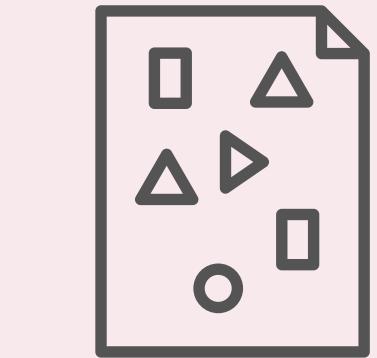
Take only the items that have Level 1 and Level 2 values in the "Parent" parameter and take the items that have IfcSlab values in the "Category" parameter, then group these items by the "ObjectType" parameter and sum the values in the "PSet_Revit_Dimensions Area" parameter and show them as a pie chart



Take only the items that have "OST_Walls" in the "Category" parameter, group them by "Type Name", sum the value of the "Area" column and add the quantity and show them in a table by removing zero values.



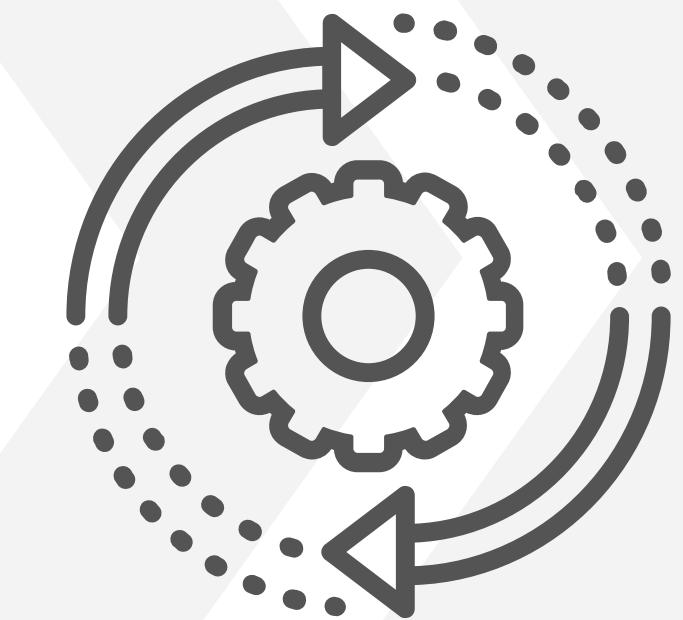
**closed
data**



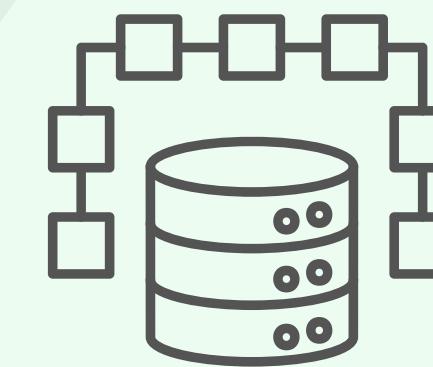
**api
cloud**

**unstructured
data**

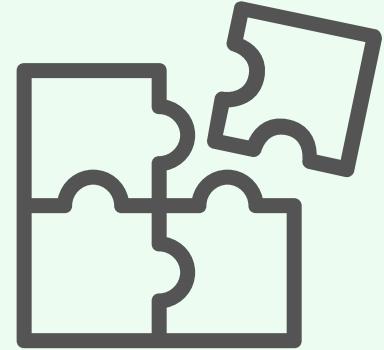
transformation



**open
data**



**structured
data**



**common
format**



**open
source**

chatGPT

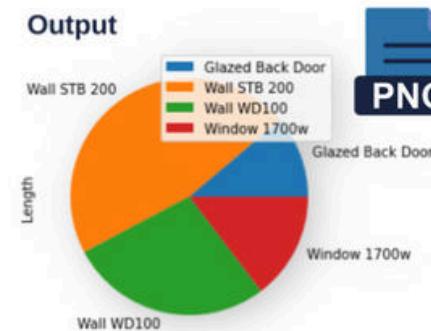
LLM, Alpaca

Output

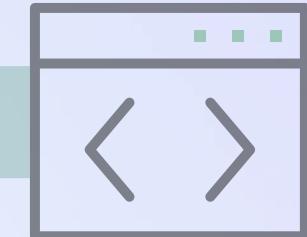
	ID	Category	Type	Length	Volume
0	12577	Wall	Wall WD100	3200	1.0
1	15889	Wall	Wall STB 200	5400	6.0
3	74456	Window	Window 1700w	1700	0.5

Output

Category	Volume	Length
Door	0.3	1300
Wall	7.0	8600
Window	0.5	1700



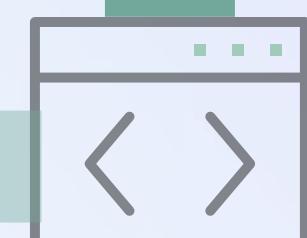
Show the differences between the new version of the project and the latest version



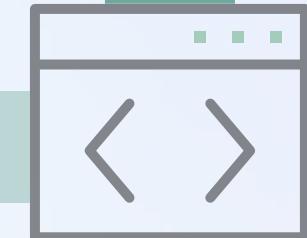
Filter the data in the project to keep the wall category items in the project



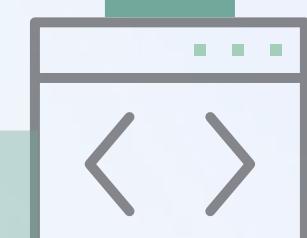
Group the project by the "Type Name" parameter and show the volume of each group



Choose the first 20 types by volume and show the result as a Pie chart



Create a PDF report with a table and a graph



PANDAS

Milliseconds

1.5" x 1.5"	0.00
Lamelle 11.5	74.82
MW 11.5	141.28
MW 17.5	67.43
STB 20.0	173.78
STB 25.0 WD 12.0	7.33
STB 30.0	88.57
STB 30.0 Rot	16.82
Standard	0.00
WC Trennwand 5.0	1.61

1 Line of code

IDE

```
QTO.py  
df[df['Category'].isin(['OST_Walls',  
'OST_Columns'])].groupby('Type')['Volume'].sum()
```

Effort



Input



Time



Output

1 Sentence

LLM Chat

Sum the 'Volume' column, grouped by
'Type', but only for rows where
'Category' is either 'OST_Walls' or
'OST_Columns'

Seconds

1.5" x 1.5"	0.00
Lamelle 11.5	74.82
MW 11.5	141.28
MW 17.5	67.43
STB 20.0	173.78
STB 25.0 WD 12.0	7.33
STB 30.0	88.57
STB 30.0 Rot	16.82
Standard	0.00
WC Trennwand 5.0	1.61

CHATGPT



