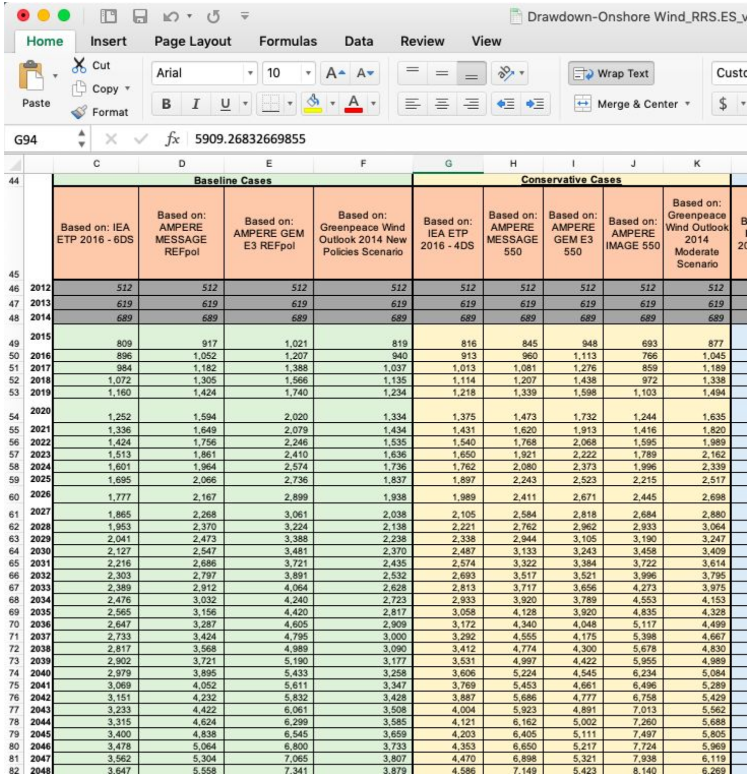
Project Drawdown
Excel → Python

Denton Gentry
dgentry@decarbon.earth

Github: https://github.com/ProjectDrawdown/solutions/

Project Drawdown Climate Solution Models

- Developed in Excel
- Several main categories
    - Reduction and Replacement
      Energy, Transport, Buildings, etc
    - Land
      Forestry, Agriculture, Land Use
    - Unique
      Food Waste, Plant-rich Diet



G94 5909.26832669855

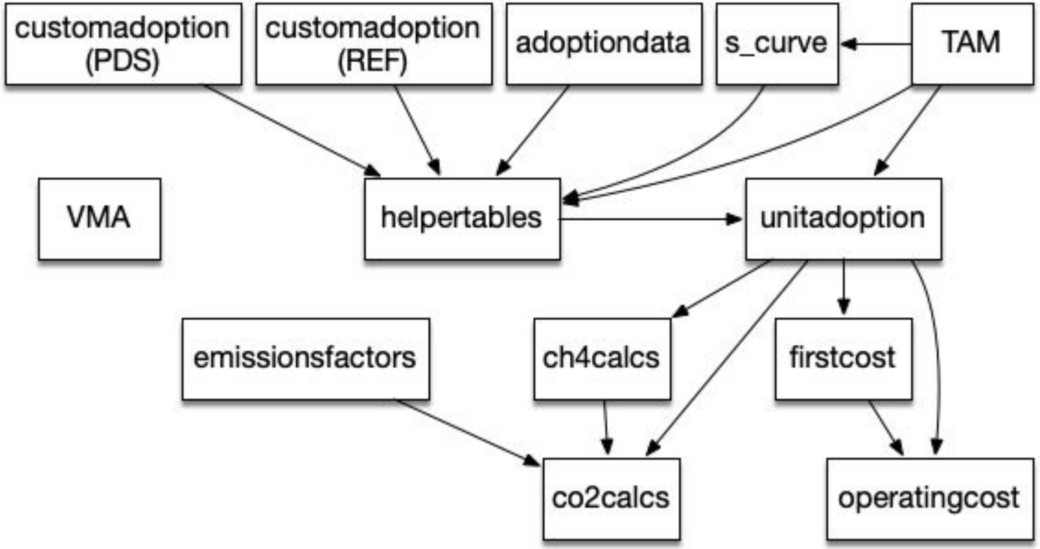
		C	D	E	F	G	H	I	J	K
44		Baseline Cases				Conservative Cases				
45		Based on: IEA ETP 2016 - 6DS	Based on: AMPERE MESSAGE REFpol	Based on: AMPERE GEM E3 REFpol	Based on: Greenpeace Wind Outlook 2014 New Policies Scenario	Based on: IEA ETP 2016 - 4DS	Based on: AMPERE MESSAGE 550	Based on: AMPERE GEM E3 550	Based on: AMPERE IMAGE 550	Based on: Greenpeace Wind Outlook 2014 Moderate Scenario
46	2012	*512*	*512*	*512*	*512*	*512*	*512*	*512*	*512*	*512*
47	2013	*619*	*619*	*619*	*619*	*619*	*619*	*619*	*619*	*619*
48	2014	*689*	*689*	*689*	*689*	*689*	*689*	*689*	*689*	*689*
49	2015	809	917	1,021	819	816	845	948	693	877
50	2016	896	1,052	1,207	940	913	960	1,113	766	1,045
51	2017	984	1,182	1,388	1,037	1,013	1,081	1,276	859	1,189
52	2018	1,072	1,305	1,566	1,135	1,114	1,207	1,438	972	1,338
53	2019	1,160	1,424	1,740	1,234	1,218	1,339	1,598	1,103	1,494
54	2020	1,252	1,594	2,020	1,334	1,375	1,473	1,732	1,244	1,635
55	2021	1,336	1,649	2,079	1,434	1,431	1,620	1,913	1,416	1,820
56	2022	1,424	1,756	2,246	1,535	1,540	1,768	2,068	1,595	1,989
57	2023	1,513	1,861	2,410	1,636	1,650	1,921	2,222	1,789	2,162
58	2024	1,601	1,964	2,574	1,736	1,762	2,080	2,373	1,996	2,339
59	2025	1,695	2,066	2,736	1,837	1,897	2,243	2,523	2,215	2,517
60	2026	1,777	2,167	2,899	1,938	1,989	2,411	2,671	2,445	2,698
61	2027	1,865	2,268	3,061	2,038	2,105	2,584	2,818	2,684	2,880
62	2028	1,953	2,370	3,224	2,138	2,221	2,762	2,962	2,933	3,064
63	2029	2,041	2,473	3,388	2,238	2,338	2,944	3,105	3,190	3,247
64	2030	2,127	2,547	3,481	2,370	2,487	3,133	3,243	3,458	3,409
65	2031	2,216	2,686	3,721	2,435	2,574	3,322	3,384	3,722	3,614
66	2032	2,303	2,797	3,891	2,532	2,693	3,517	3,521	3,996	3,795
67	2033	2,389	2,912	4,064	2,628	2,813	3,717	3,656	4,273	3,975
68	2034	2,476	3,032	4,240	2,723	2,933	3,920	3,789	4,553	4,153
69	2035	2,565	3,156	4,420	2,817	3,058	4,128	3,920	4,835	4,328
70	2036	2,647	3,287	4,605	2,909	3,172	4,340	4,048	5,117	4,499
71	2037	2,733	3,424	4,795	3,000	3,292	4,555	4,175	5,398	4,667
72	2038	2,817	3,568	4,989	3,090	3,412	4,774	4,300	5,678	4,830
73	2039	2,902	3,721	5,190	3,177	3,531	4,997	4,422	5,955	4,989
74	2040	2,979	3,895	5,433	3,258	3,606	5,224	4,545	6,234	5,084
75	2041	3,069	4,052	5,611	3,347	3,769	5,453	4,661	6,496	5,289
76	2042	3,151	4,232	5,832	3,428	3,887	5,686	4,777	6,758	5,429
77	2043	3,233	4,422	6,061	3,508	4,004	5,923	4,891	7,013	5,562
78	2044	3,315	4,624	6,299	3,585	4,121	6,162	5,002	7,260	5,688
79	2045	3,400	4,838	6,545	3,659	4,203	6,405	5,111	7,497	5,805
80	2046	3,478	5,064	6,800	3,733	4,353	6,650	5,217	7,724	5,969
81	2047	3,562	5,304	7,065	3,807	4,470	6,898	5,321	7,938	6,119
82	2048	3,647	5,558	7,341	3,879	4,586	7,149	5,423	8,140	6,269

Excel models

Well structured in Excel:

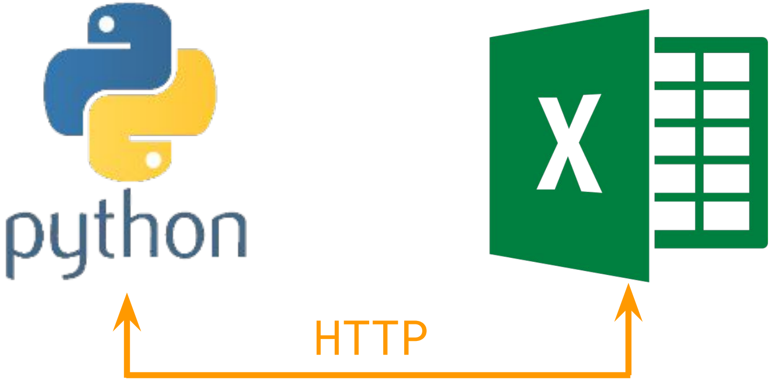
- partitioned calculations
- tabs == modules

[illegible]	[illegible]	[illegible]
$87,716,610,681.19	$581,821,627,703.17	$494,105,017,021.97
$90,840,167,559.98	$602,540,086,080.97	$511,699,918,520.99
$93,887,282,977.22	$622,751,510,554.47	$528,864,227,577.26
$96,849,200,773.85	$642,397,821,785.34	$545,548,621,011.49
$97,559,747,865.68	$647,110,859,171.51	$549,551,111,305.83



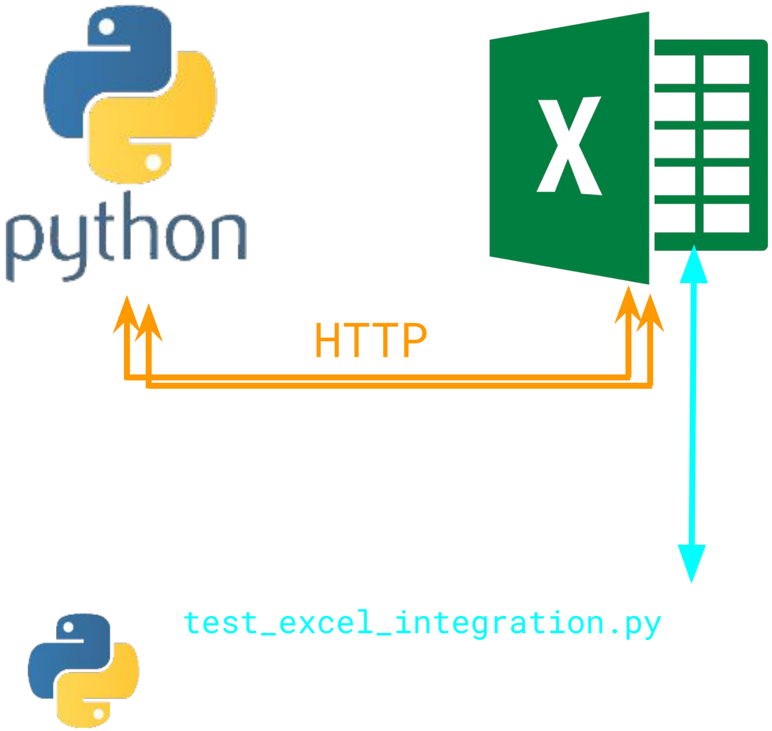
Step 1: Incremental Implementation

- Python runs in local webserver
- Visual Basic HTTP fetch from Excel



Step 2: Integration test

- Start Excel twice
  - original file
  - HTTP fetch file
- Compare results

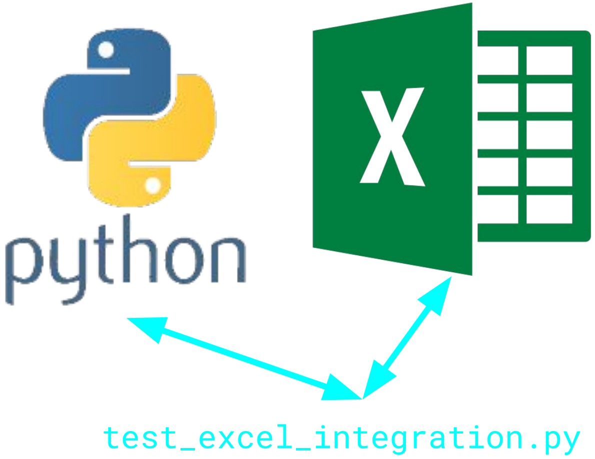


Step 2: Integration test excerpt

```
def verify_tam_data(obj, verify):
    """Verify Total Addressable Market sheet."""
    verify['TAM Data'] = [
        ('W46:Y94', obj.tm.forecast_min_max_sd(
            region='World').reset_index(drop=True), None),
        ('AA46:AC94', obj.tm.forecast_low_med_high(
            region='World').reset_index(drop=True), None),
        ('BX50:BZ96', obj.tm.forecast_trend(region='World',
            trend='Linear').reset_index(drop=True), None),
        ('CE50:CH96', obj.tm.forecast_trend(region='World',
            trend='Degree2').reset_index(drop=True), None),
```

Step 3: Discontinue HTTP

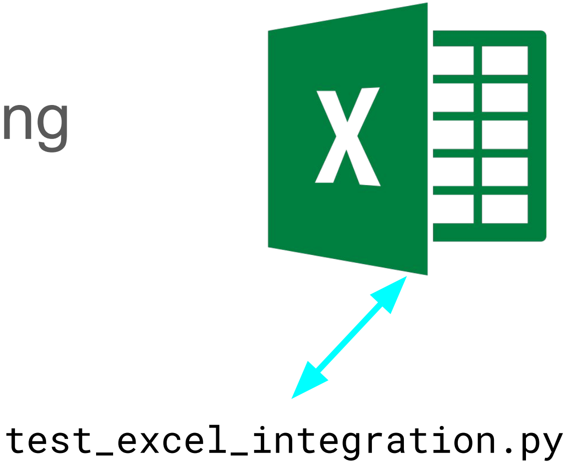
- standalone Python models
- test compares results



test_excel_integration.py

Integration test v1

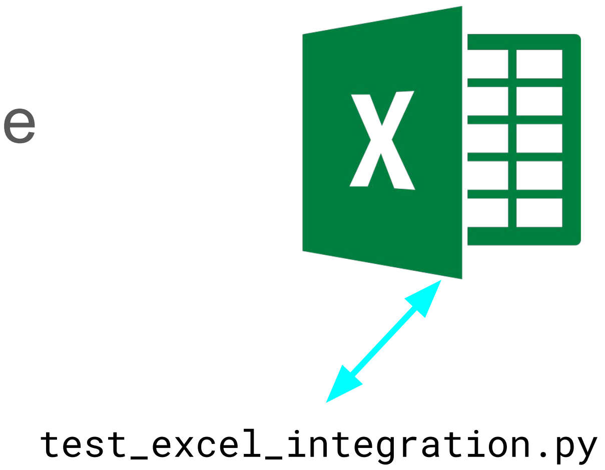
- https://www.xlwings.org/ for everything



test_excel_integration.py

Integration test v2

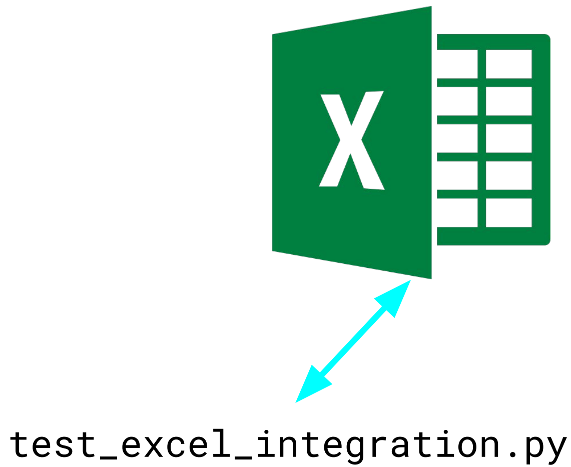
- xlwings to run VBA, write out the file
- xlrd to read data from the file



test_excel_integration.py

Integration test v3

- xlwings offline to write out ZIP
- test uses only the ZIP



`test_excel_integration.py`

What comes after Excel?

- At some point we will retire Excel

- What will we do then?

Github: https://github.com/ProjectDrawdown/solutions/