Refinitiv Developer Days 2019

Spreadsheets, Python and Financial Data

Felix Zumstein

Agenda

1. Introduction
2. Spreadsheets in the age of blockchain and AI
3. Demo: Eikon with Excel and Python

xlwings

1. Introduction

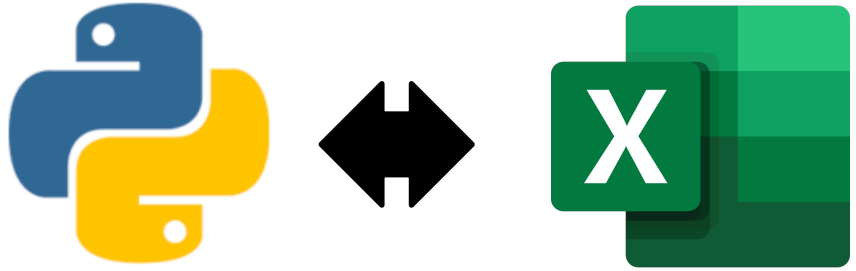
A small team with a big mission

Innovative Solutions for Microsoft Excel

xlwings

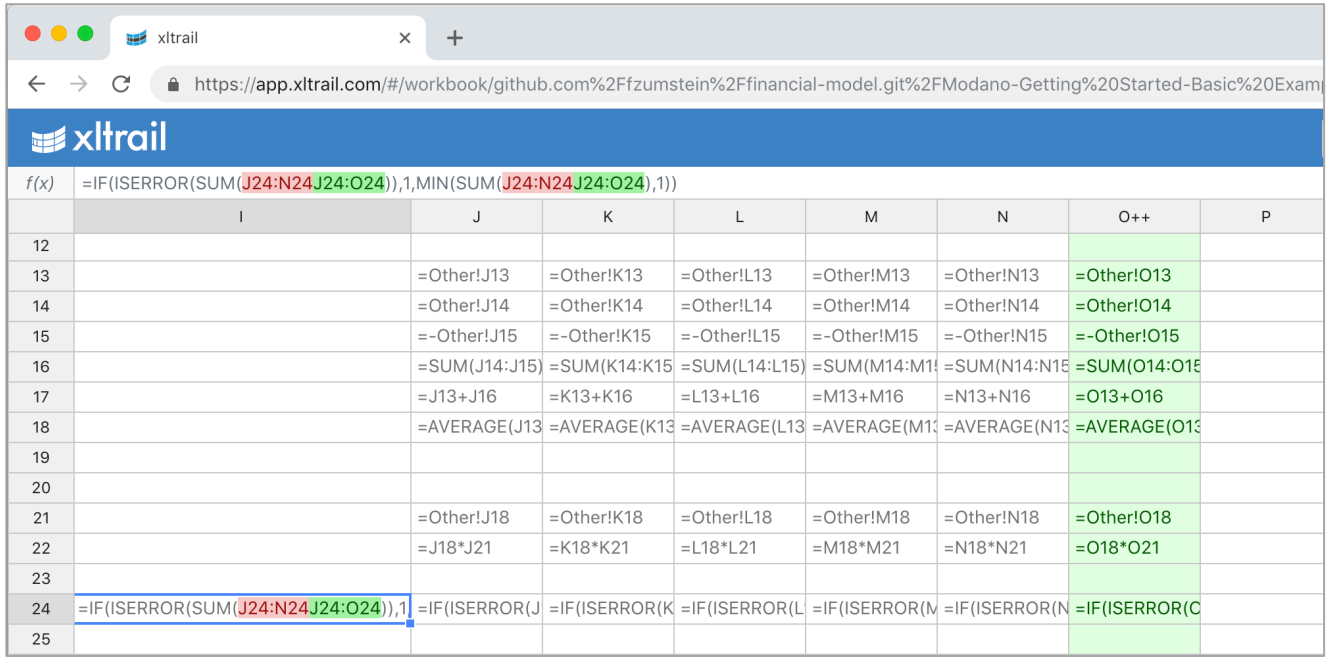
xlwings

- Python for Excel
- Scripts, UDFs,
  Macros, REST API
- Windows & Mac
- Free & open-source
- Professional support plans



xlwings

xltrail




```
107 107        If GetConfigFromFile(GetConfigFilePath, "DEBUG UDFS", setting) Then
108     -          pressed = setting
        108 +      If setting = "True" Then
        109 +          pressed = True
        110 +      Else
        111 +          pressed = False
        112 +      End If
109 113        Else
```

- Version control for Excel
- Tracks formulas and VBA code

xlwings

2. Spreadsheets in the age of blockchain and AI

Marks and Spencer 2016 Q1 report

Marks & Spencer Group PLC + Add to myFT

M&S takes back shop-soiled figures

Embarrassment as 'rise' in sales of 1.3% switched to fall of 0.4%

"It is not good," Helen Weir, M&S's chief financial officer, told the Financial Times after the error was discovered. She said she was "shocked" when she found out that double-counting in a spreadsheet had led M&S to say that sales had risen 1.3 per cent in the three months to July, when they had actually fallen 0.4 per cent.

xlwings

Source: https://www.ft.com/content/1fc57f26-4467-11e6-9b66-0712b3873ae1 (7/7/2016)

The London Whale

Search Bloomberg Sign In

Quicktake

The London Whale

By Patricia Hurtado

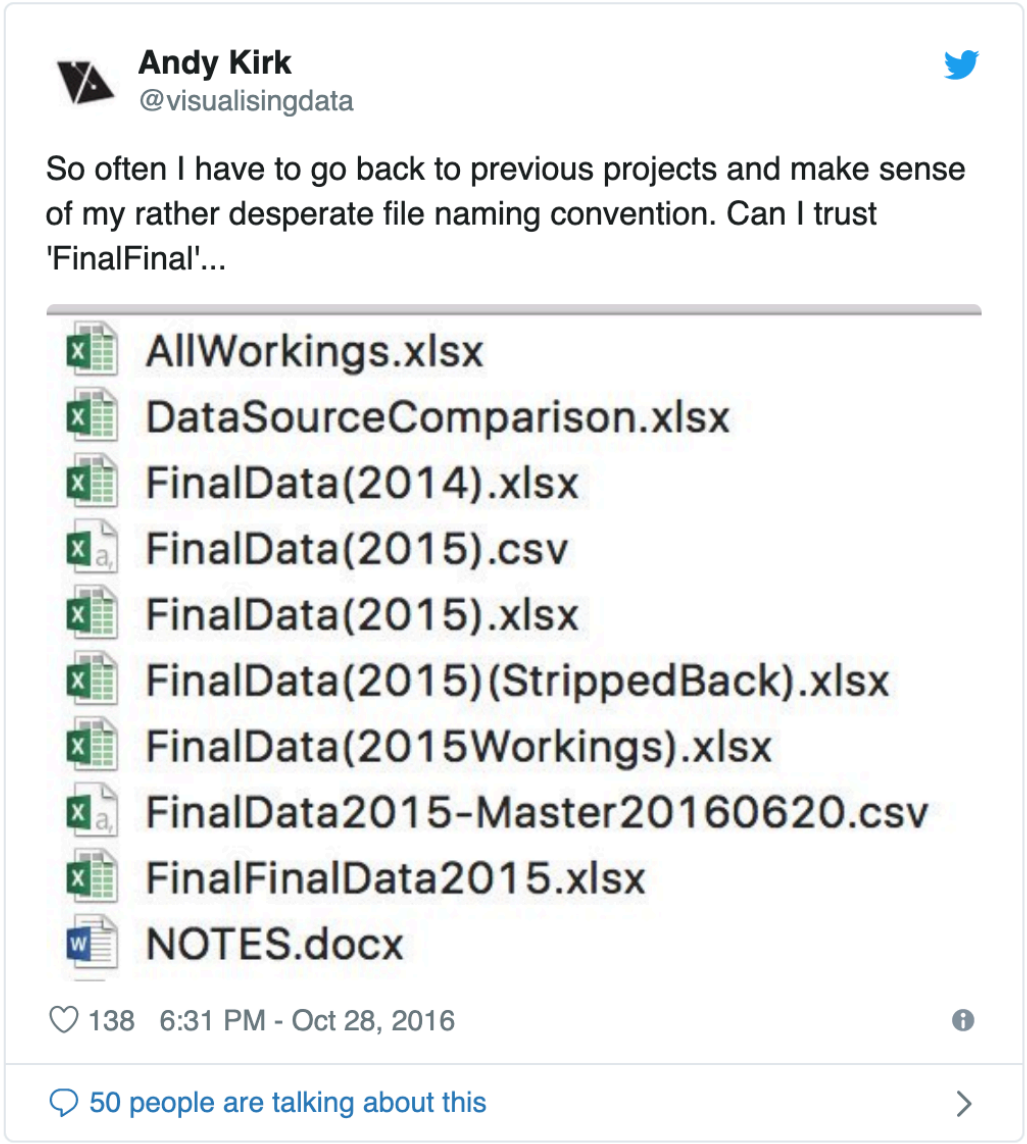
Updated on 23 February 2016, 23:04 CET

broader systemic failure: Risk limits, for instance, were breached more than 300 times before the bank switched to a more lenient risk-evaluation formula – one that underestimated risk by half because of a spreadsheet error. The report by the Fed's inspector general also supported the view

xlwings

Source: https://www.bloomberg.com/quicktake/the-london-whale (2/23/2016)

Look familiar? Part I



xlwings

Look familiar? Part II

```
=OFFSET('Appendix 4 -
Odds'!$B$3,ROW(INDEX('Appendix 4 -
Odds'!$B$3:$AH$35,MATCH(I12,'Appendix 4 -
Odds'!$B$3:$B$35,0),1))-
3,COLUMN(INDEX('Appendix 4 -
Odds'!$B$3:$AH$35,,MATCH(I13,'Appendix 4 -
Odds'!$B$3:$AH$3,0),1))-2)+INDEX('Round Robin
Predictions'!$C$4:$O$36,MATCH(I12, 'Round
Robin Predictions'!$D$4:$D$36,0),3)-
INDEX('Round Robin
Predictions'!$C$4:$O$36,MATCH(I13, 'Round
Robin Predictions'!$D$4:$D$36,0),3)
```

xlwings

Source: https://techcommunity.microsoft.com/t5/Excel-Blog/How-We-Built-our-Soccer-Tournament-Predictor-in-Excel/ba-p/199802

Our interpretation

1) Excel = code, so treat it as such!
2) Migrating away from Excel is not always possible.

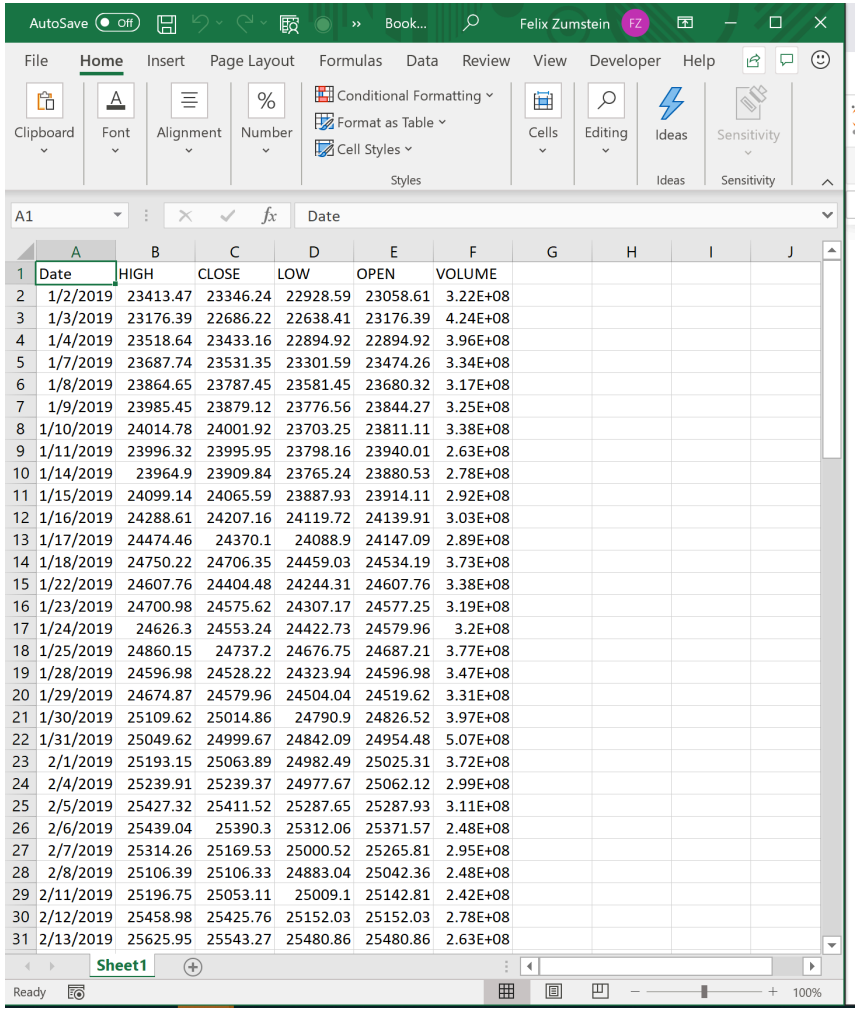
xlwings

3. Demo

xlwings

1. Interactive use with Jupyter notebook
2. UDF: Correlation matrix
3. Reporting
4. Macro: Monte Carlo simulation
5. xlwings workflow: Automated tests
6. REST API

1. Interactive: Jupyter notebook



	A	B	C	D	E	F
1	Date	HIGH	CLOSE	LOW	OPEN	VOLUME
2	1/2/2019	23413.47	23346.24	22928.59	23058.61	3.22E+08
3	1/3/2019	23176.39	22686.22	22638.41	23176.39	4.24E+08
4	1/4/2019	23518.64	23433.16	22894.92	22894.92	3.96E+08
5	1/7/2019	23687.74	23531.35	23301.59	23474.26	3.34E+08
6	1/8/2019	23864.65	23787.45	23581.45	23680.32	3.17E+08
7	1/9/2019	23985.45	23879.12	23776.56	23844.27	3.25E+08
8	1/10/2019	24014.78	24001.92	23703.25	23811.11	3.38E+08
9	1/11/2019	23996.32	23995.95	23798.16	23940.01	2.63E+08
10	1/14/2019	23964.9	23909.84	23765.24	23880.53	2.78E+08
11	1/15/2019	24099.14	24065.59	23887.93	23914.11	2.92E+08
12	1/16/2019	24288.61	24207.16	24119.72	24139.91	3.03E+08
13	1/17/2019	24474.46	24370.1	24088.9	24147.09	2.89E+08
14	1/18/2019	24750.22	24706.35	24459.03	24534.19	3.73E+08
15	1/22/2019	24607.76	24404.48	24244.31	24607.76	3.38E+08
16	1/23/2019	24700.98	24575.62	24307.17	24577.25	3.19E+08
17	1/24/2019	24626.3	24553.24	24422.73	24579.96	3.2E+08
18	1/25/2019	24860.15	24737.2	24676.75	24687.21	3.77E+08
19	1/28/2019	24596.98	24528.22	24323.94	24596.98	3.47E+08
20	1/29/2019	24674.87	24579.96	24504.04	24519.62	3.31E+08
21	1/30/2019	25109.62	25014.86	24790.9	24826.52	3.97E+08
22	1/31/2019	25049.62	24999.67	24842.09	24954.48	5.07E+08
23	2/1/2019	25193.15	25063.89	24982.49	25025.31	3.72E+08
24	2/4/2019	25239.91	25239.37	24977.67	25062.12	2.99E+08
25	2/5/2019	25427.32	25411.52	25287.65	25287.93	3.11E+08
26	2/6/2019	25439.04	25390.3	25312.06	25371.57	2.48E+08
27	2/7/2019	25314.26	25169.53	25000.52	25265.81	2.95E+08
28	2/8/2019	25106.39	25106.33	24883.04	25042.36	2.48E+08
29	2/11/2019	25196.75	25053.11	25009.1	25142.81	2.42E+08
30	2/12/2019	25458.98	25425.76	25152.03	25152.03	2.78E+08
31	2/13/2019	25625.95	25543.27	25480.86	25480.86	2.63E+08

104 rows × 5 columns

```
In [4]: # View the data in Excel
        xw.view(data)

In [5]: # Create a new Excel workbook
        wb = xw.Book()

In [6]: # Write a DataFrame
        wb.sheets[0]['A1'].value = data

In [7]: # Read a DataFrame
        wb.sheets[0]['A1'].options(pd.DataFrame, expand='table').value
```

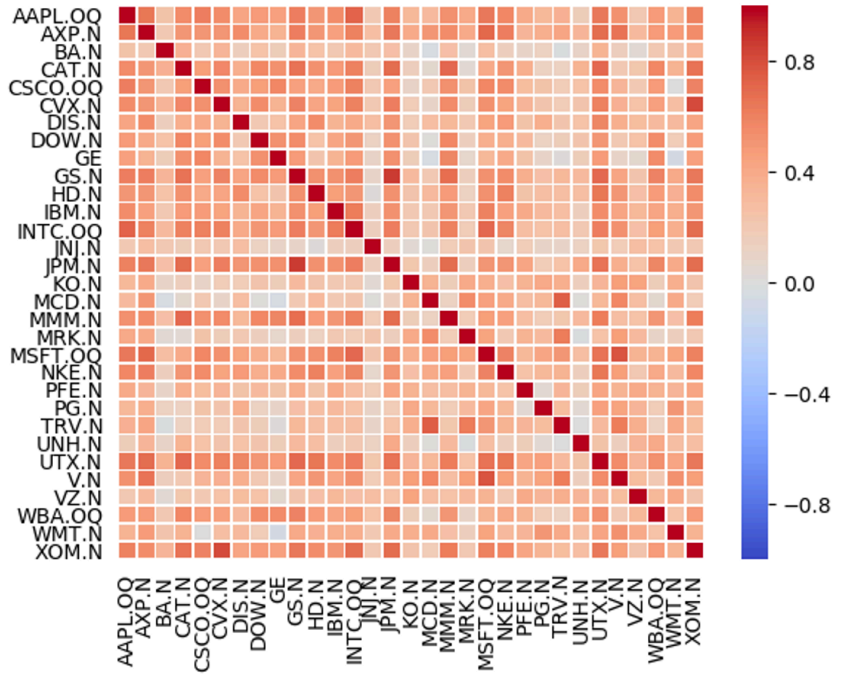
Out[7]:

Date	HIGH	CLOSE	LOW	OPEN	VOLUME
2019-01-02	23413.47	23346.24	22928.59	23058.61	321566851.0
2019-01-03	23176.39	22686.22	22638.41	23176.39	424238479.0
2019-01-04	23518.64	23433.16	22894.92	22894.92	396015724.0
2019-01-07	23687.74	23531.35	23301.59	23474.26	334203936.0
2019-01-08	23864.65	23787.45	23581.45	23680.32	317423120.0
2019-01-09	23985.45	23879.12	23776.56	23844.27	324574321.0
2019-01-10	24014.78	24001.92	23703.25	23811.11	338145444.0
2019-01-11	23996.32	23995.95	23798.16	23940.01	262647497.0
2019-01-14	23964.90	23909.84	23765.24	23880.53	277558125.0
2019-01-15	24099.14	24065.59	23887.93	23914.11	291574850.0
2019-01-16	24288.61	24207.16	24119.72	24139.91	302833511.0
2019-01-17	24474.46	24370.10	24088.90	24147.09	288587672.0

xlwings

2. UDF: Correlation matrix

	A	B	C	D	E	F	G	H	I	J	K
1	**RICS**	<Corr Plot>									
2	AAPL.OQ		AAPL.OQ	AXP.N	BA.N	CAT.N	CSCO.OQ	CVX.N	DIS.N	DOW.N	GE
3	AXP.N	AAPL.OQ	1	0.652667093	0.243216988	0.561688615	0.617156978	0.547986694	0.422578993	0.487864034	0.45670486
4	BA.N	AXP.N	0.652667093							02421555	0.353202205
5	CAT.N	BA.N	0.243216988	0.2018						0649424	0.171248844
6	CSCO.OQ	CAT.N	0.561688615	0.49572						0726244	0.525654885
7	CVX.N	CSCO.OQ	0.617156978	0.51076						6694523	0.571751085
8	DIS.N	CVX.N	0.547986694	0.51158						4494245	0.345554917
9	DOW.N	DIS.N	0.422578993	0.54282						8878003	0.256275637
10	GE	DOW.N	0.487864034	0.40242						1	0.536946879
11	GS.N	GE	0.45670486	0.35320						6946879	1
12	HD.N	GS.N	0.61695036	0.61632						0542903	0.486673877
13	IBM.N	HD.N	0.49476905	0.50552						1747323	0.227726613
14	INTC.OQ	IBM.N	0.547439634	0.4535						0.427203	0.344340819
15	JNJ.N	INTC.OQ	0.721700866	0.60874						6396173	0.47764493
16	JPM.N	JNJ.N	0.189466684	0.26722						9418792	0.110954813
17	KO.N	JPM.N	0.598451024	0.64101						9271139	0.532052069
18	MCD.N	KO.N	0.318872782	0.393						7282084	0.171841328
19	MMM.N	MCD.N	0.299612011	0.49566						0040192	-0.044986946
20	MRK.N	MMM.N	0.536521256	0.54736						2692169	0.578813343
21	MSFT.OQ	MRK.N	0.40686132	0.39415						5738885	0.078385847
22	NKE.N	MSFT.OQ	0.63430709	0.70656						1062411	0.322583275
23	PFE.N	NKE.N	0.573821073	0.61299						2633456	0.397904469
24	PG.N	PFE.N	0.404830778	0.33028						3348023	0.246940687
25	TRV.N	PG.N	0.319892529	0.37712						1229972	0.091921829
26	UNH.N	TRV.N	0.360358319	0.42255708	-0.011669303	0.131916837	0.181992877	0.181694952	0.22721182	0.170751375	0.020594517
27	UTX.N	UNH.N	0.165893846	0.331299795	0.116937254	0.352342724	0.251377173	0.23892755	0.255978789	0.234044975	0.169789571
28	V.N	UTX.N	0.639784189	0.693593956	0.346519517	0.697671076	0.560681282	0.60628573	0.583768635	0.501496627	0.482606218



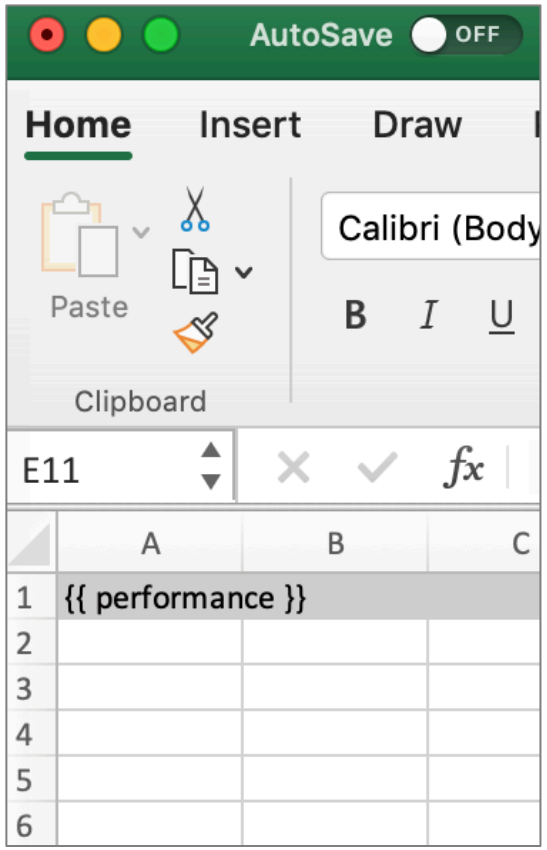
Returns

Ready

xlwings

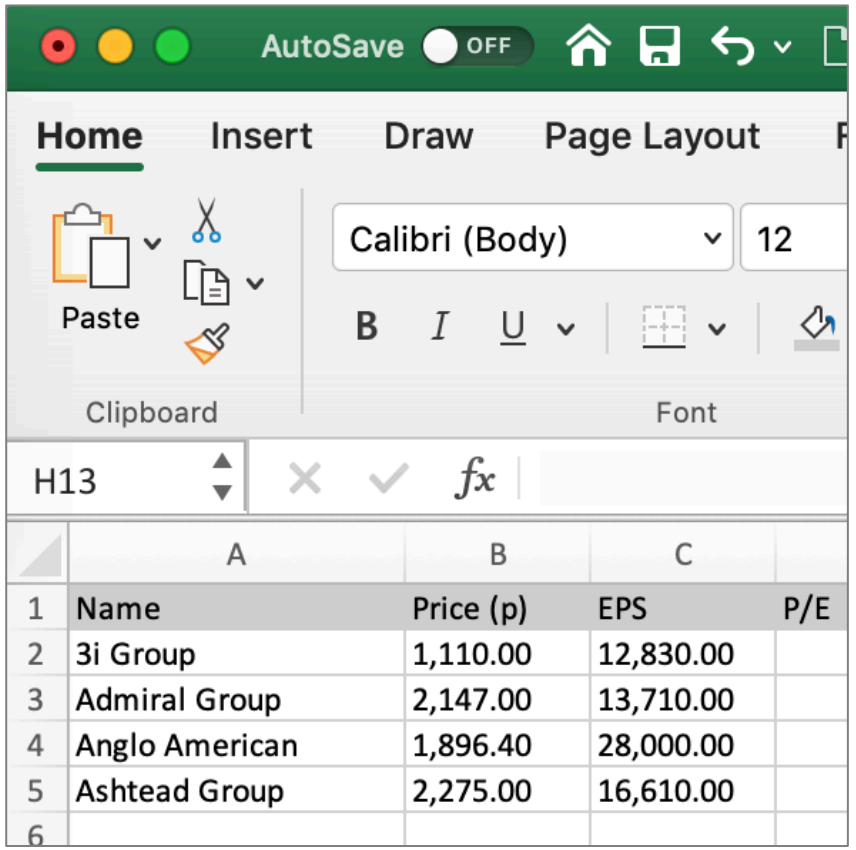
3. Reporting: xlwings-reports

Template




```
create_report(
"template.xlsx",
performance=df
)
```

Report



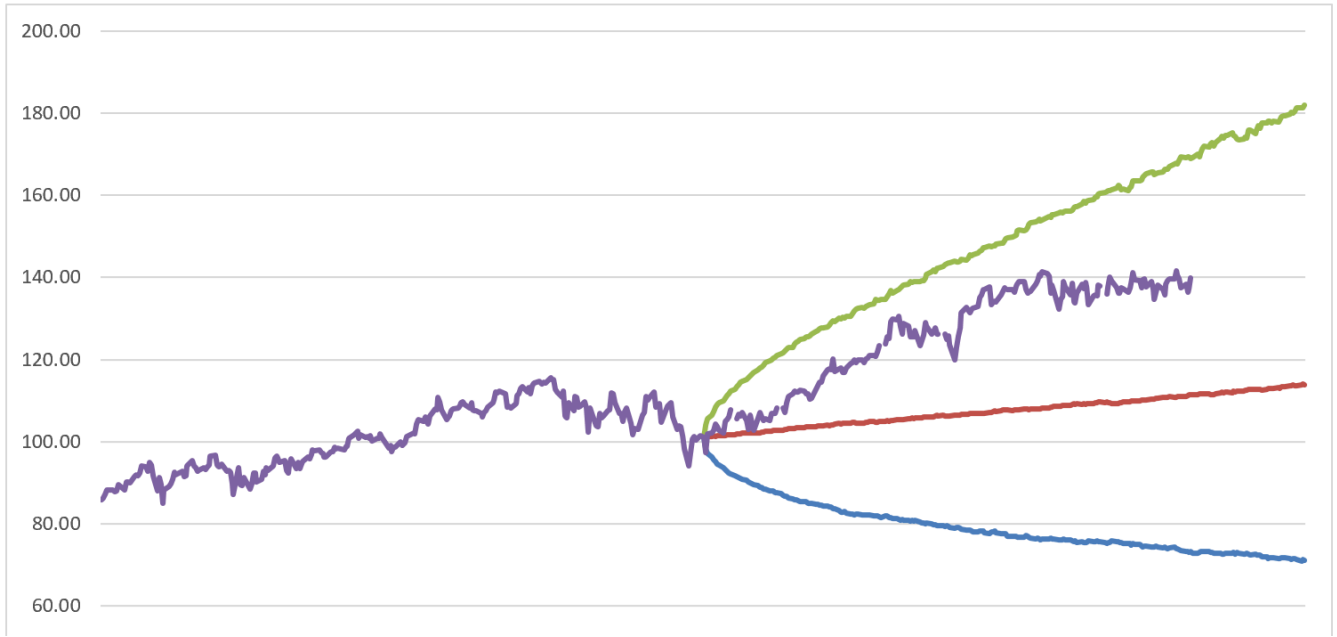
xlwings

4. Macro: Monte Carlo simulation

Monte Carlo simulation based on mean/stdev of first 252 trading days

Number of Simulations:	5,000
Start Date:	1/1/2018
RIC:	MSFT.OQ

Run



xlwings

5.a The xlwings workflow

2 – Collaborate

- Make collaboration easy through proper version control (no emails!)
- Peer review changes
- Spot errors before they bite

3 - Test

- Don't trust humans! Run automated tests

4 - Release

- Know which workbook is the latest one
- Release and deploy your workbook automatically

1 - Automate

- No copy/paste errors! Work smarter through automation
- Reduce errors by getting rid of complex formulas

5.b Automated testing

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**
2	Intial Balance	30	20	30	60	100	140	160	150	150	170	200	230
3	Inflows	20	40	60	70	70	50	20	30	50	60	60	40
4	Outflows	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30
5	**Ending Balance**	**20**	**30**	**60**	**100**	**140**	**160**	**150**	**150**	**170**	**200**	**230**	**240**

```python
def test_cash_flow_formula_integrity(book):
    sheet = book.sheets[0]
    sheet['B2'].value = 100
    sheet['B3:M3'].value = 10
    sheet['B4:M4'].value = -5
    assert sheet['M5'].value == 160
```

xltrail

f(x)

	A	B	C
1		Jan	Feb
2	Intial Balance	30	=B520
3	Inflows	20	40
4	Outflows	-30	-30
5	Ending Balance	=SUM(B2:B4)	=SUM(C2:C4)
6			

xlwings

6. REST API

Run the xlwings REST API server:

```
$ xlwings restapi run
```

Issue a GET request, e.g:

```
http://127.0.0.1:5000/book/timeseries.xlsx/sheets/0/range
```

xlwings

Thank You

Let’s connect:
https://www.linkedin.com/in/felix-zumstein