



AFP
2020

VIRTUAL
EXPERIENCE
10/19-10/29

Advanced Dynamic Financial Modeling

Carl Seidman

CPA/CFF, CIRA, CFE, CGMA, AM

Seidman Financial

How does this resonate with you?

Why Many Financial Models Fail

① Today's Model Requirements

- Specific objectives and usage
- Limited maintenance
- Contained data
- Limited time periods
- Generally requires:
 - Less-frequent updates
 - Less flexibility
 - Fewer dynamics

② The Model 'Evolution'

- Broad scope and use
- Ongoing maintenance
- Growing data sets
- Evolving time periods
- Generally requires:
 - More-frequent updates
 - More flexibility
 - Greater dynamics

Excel is an extremely flexible tool however...

~~Intentional
Scale~~

~~Incorporates
New
Data Easily~~

~~Seamless
Agile
Updates~~

~~Manual
Involvement~~

Excel is an extremely flexible tool however...

Intentional
Scale

~~Incorporates
New
Data Easily~~

~~Seamless
Agile
Updates~~

~~Manual
Involvement~~

/IFP 2020



Dynamic Modeling

Building robust yet flexible models that can

- ✓ Scales intentionally

Excel is an extremely flexible tool however...

Intentional
Scale

~~Incorporates
New
Data Easily~~

Seamless
Agile
Updates

~~Manual
Involvement~~

/IFP 2020



Dynamic Modeling

Building robust yet flexible models that:

- ✓ Scales intentionally
- ✓ Seamlessly agile updates

Excel is an extremely flexible tool however...

Intentional
Scale

Incorporates
New
Data Easily

Seamless
Agile
Updates

~~Manual
Involvement~~

AFP 2020



Dynamic Modeling

Building robust yet flexible models that:

- ✓ Scales intentionally
- ✓ Seamlessly agile updates
- ✓ Incorporates new data easily

Excel is an extremely flexible tool however...

**Intentional
Scale**

**Incorporates
New
Data Easily**

**Seamless
Agile
Updates**

**Manual
Involvement**

/AFP 2020



Dynamic Modeling

Building robust yet flexible models that:

- ✓ **Scales intentionally**
- ✓ **Seamlessly agile updates**
- ✓ **Incorporates new data easily**
- ✓ **Minimizes manual involvement**

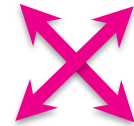
The objective of financial modeling isn't building a model

The objective is to provide end-users great intelligence

So they can make better decisions

How might the model need to change?

What new data is likely to enter the model?



How might the data need to be presented?

What degree of dynamics/flexibility is necessary?

(or is too much)?

How Dynamic Functionality Differs

Excel



**SaaS
Platform**

Excel Dynamic Functionality

- **Dynamic Tables**
- **Dynamic References**
- **Dynamic Ranges**
- **Dynamic Arrays**

- 
-
- **OFFSET(**
 - **INDIRECT(**
 - **INDEX(+ MATCH(**
 - **SEQUENCE(**
 - **UNIQUE(**
 - **FILTER(**
 - **SORT(**
 - **SORTBY(**
 - **RANDARRAY(**
 - **DYNAMIC (#) ARRAYS**



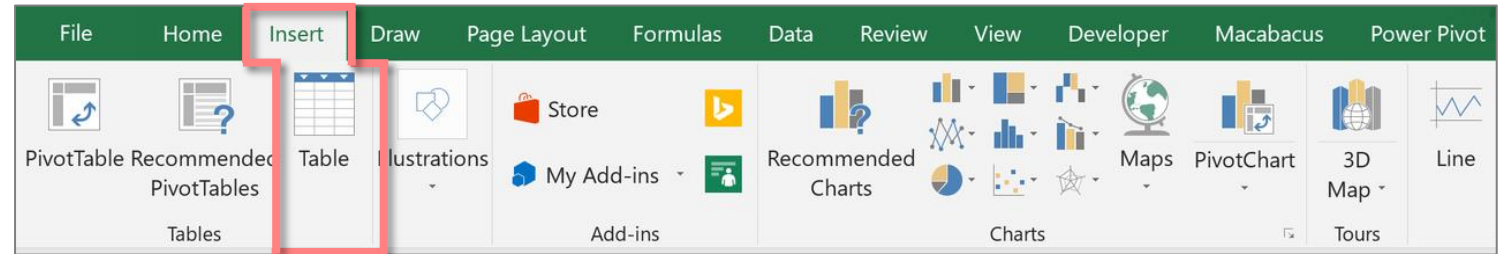
DYNAMIC TABLES



**Allows for the dynamic referencing
of expanding and contracting data sets
without requiring a formula update**

Dynamic Tables

- ✓ Scales intentionally
- ✓ Seamlessly agile updates
- ✓ Incorporates new data easily
- ✓ Minimizes manual involvement



G16 fx =SUM([Total Sales])

	A	B	C	D	E	F	G
1	Salesman	Region	State	Product	Sales Units	Price Per Unit	Total Sales
2	Rogers	East	New York	Ford Interceptor	575	\$299.99	\$172,494.25
3	Smith	East	New Jersey	Chevy Tahoe	408	\$1,099.99	\$448,795.92
4	Johnson	Northeast	Maine	Ford Edge	579	\$799.99	\$463,194.21
5	Williams	Northeast	Rhode Island	Ford Crown Victoria	27	\$499.99	\$13,499.73
6	Thomas	Northeast	Massachusetts	Humvee	682	\$1,699.99	\$1,159,393.18
7	Williams	West	California	Chevy Tahoe	484	\$1,099.99	\$532,395.16
8	Williams	West	Oregon	Ford Crown Victoria	992	\$499.99	\$495,990.08
9	Johnson	West	Washington	Ford Interceptor	791	\$299.99	\$237,292.09
10	Thomas	West	Nevada	Ford Edge	679	\$799.99	\$543,193.21
11	Johnson	West	Utah	Chevy Tahoe	614	\$1,099.99	\$675,393.86
12	Williams	Midwest	Michigan	Ford Edge	727	\$799.99	\$581,592.73
13	Thomas	South	Texas	Ford Interceptor	921	\$299.99	\$276,290.79
14	Thomas	Southeast	Florida	Ford Edge	861	\$799.99	\$688,791.39
15	Thomas	Southeast	Georgia	Chevy Tahoe	550	\$1,099.99	\$604,994.50
16							\$6,893,311.10

None
Average
Count
Count Numbers
Max
Min
Sum
StdDev

Dynamic Tables



- ✓ Tables can be named and referenced →
- ✓ Entire data sets and columns can be referenced by a single formula
- ✓ Formulas reference dynamic ranges

Salesman	Region	State	Product	Sales Units	Price Per Unit	Total Sales
Smith	East	New York	Ford Edge	38	\$799.99	\$30,399.62
Williams	West	California	Ford Crown Victoria	523	\$499.99	\$261,494.77
Smith	West	Utah	Chevy Tahoe	637	\$1,099.99	\$700,693.63
Smith	West	Nevada	Ford Edge	155	\$799.99	\$123,998.45
Rogers	West	Nevada	Humvee	795	\$1,699.99	\$1,351,492.05
Rogers	East	New Jersey	Ford Edge	313	\$799.99	\$250,396.87
Smith	East	New York	Ford Crown Victoria	50	\$499.99	\$24,999.50
Rogers	West	Nevada	Ford Interceptor	371	\$299.99	\$111,296.29
Williams	East	New York	Chevy Tahoe	698	\$1,099.99	\$767,793.02
Smith	South	Texas	Humvee	170	\$1,699.99	\$288,998.30
Smith	West	Nevada	Ford Crown Victoria	361	\$499.99	\$180,496.39
Smith	West	California	Ford Edge	710	\$799.99	\$567,992.90
Smith	South	New Mexico	Ford Crown Victoria	737	\$499.99	\$368,492.63
Williams	South	Oklahoma	Chevy Tahoe	253	\$1,099.99	\$278,297.47
Williams	South	Texas	Humvee	191	\$1,699.99	\$324,698.09
Thomas	West	California	Ford Crown Victoria	321	\$499.99	\$160,496.79
Smith	West	Nevada	Ford Interceptor	839	\$299.99	\$251,691.61
Williams	South	Texas	Ford Crown Victoria	212	\$499.99	\$105,997.88
Williams	South	Texas	Ford Edge	152	\$799.99	\$121,598.48



OFFSET FUNCTION



Why You Should Use It:

Allows for the dynamic referencing of expanding and contracting data sets without requiring a formula updates

OFFSET FUNCTION

OFFSET(reference, rows, cols, [height], [width])











Reference: The reference from which you want to base the offset.



***Microsoft is incorrect!
It can ALSO be negative***

Source: Microsoft

OFFSET FUNCTION

	A	B	C	D	E	F	G	H	I	J	K	
1	Vehicle Make	Vehicle ID	Region	Region Name	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19	Jul-19	
2	Chevy Tahoe	100 001		Northwest	24545	32642	10755	21228	17812	48349	11509	
3	Ford Crown Victoria	101 006		Pacific	17878	15331	12006	30310	39294	18790	25301	
4	Ford Edge	102 006		Pacific	44482	10626	11081	48966	11231	44565	34671	
5	Ford Interceptor	103 004		Southwest	12814	41459	22946	21307	22108	37476	10129	
6	Chevy Tahoe	100 003		Midwest	38508	48444	37828	41543	14423	15271	20857	
7	Ford Crown Victoria	101 006		Pacific	29040	34048	43240	32870	41051	46881	35524	
8	Ford Edge	102 005		Mountain	41502	49307	14787	16835	22399	26342	32145	
9	Ford Interceptor	103 005		Mountain	29186	48846	41684	22734	23982	19729	35538	
10	Chevy Tahoe	100 001		Northwest	48127	32579	30853	10860	28932	42381	49115	



OFFSET FUNCTION

Make Dynamic!

Width = 7

	A	B	C	D	E	F	G	H	I	J	K
1	Vehicle Make	Vehicle ID	Region	Region Name	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19	Jul-19
2	Chevy Tahoe	100 001		Northwest	24545	32642	10755	21228	17812	48349	11509
3	Ford Crown Victoria	101 006		Pacific	17878	15331	12006	30310	39294	18790	25301
4	Ford Edge	102 006		Pacific	44482	10626	11081	48966	11231	44565	34671
5	Ford Interceptor	103 004		Southwest	12814	41459	22946	21307	22108	37476	10129
6	Chevy Tahoe	100 003		Midwest	38508	48444	37828	41543	14423	15271	20857
7	Ford Crown Victoria	101 006		Pacific	29040	34048	43240	32870	41051	46881	35524
8	Ford Edge	102 005		Mountain	41502	49307	14787	16835	22399	26342	32145
9	Ford Interceptor	103 005		Mountain	29186	48846	41684	22734	23982	19729	35538
10	Chevy Tahoe	100 001		Northwest	48127	32579	30853	10860	28932	42381	49115

Height = 9

Make Dynamic!

OFFSET FUNCTION

Width = 7

	A	B	C	D	E	F	G	H	I	J	K
1	Vehicle Make	Vehicle ID	Region	Region Name	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19	Jul-19
2	Chevy Tahoe	100 001		Northwest	24545	32642	10755	21228	17812	48349	11509
3	Ford Crown Victoria	101 006		Pacific	17878	15331	12006	30310	39294	18790	25301
4	Ford Edge	102 006		Pacific	44482	10626	11081	48966	11231	44565	34671
5	Ford Interceptor	103 004		Southwest	12814	41459	22946	21307	22108	37476	10129
6	Chevy Tahoe	100 003		Midwest	38508	48444	37828	41543	14423	15271	20857
7	Ford Crown Victoria	101 006		Pacific	29040	34048	43240	32870	41051	46881	35524
8	Ford Edge	102 005		Mountain	41502	49307	14787	16835	22399	26342	32145
9	Ford Interceptor	103 005		Mountain	29186	48846	41684	22734	23982	19729	35538
10	Chevy Tahoe	100 001		Northwest	48127	32579	30853	10860	28932	42381	49115

Height = 9

=OFFSET(reference, rows, cols, [height], [width])
=OFFSET(E2 , 0 , 0 , 9 , 7)

OFFSET FUNCTION

	A	B	C	D	E	F	G	H	I	J	K
1	Vehicle Make	Vehicle ID	Region	Region Name	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19	Jul-19
2	Chevy Tahoe	100	001	Northwest	24545	32642	10755	21228	17812	48349	11509
3	Ford Crown Victoria	101	006	Pacific	17878	15331	12006	30310	39294	18790	25301
4	Ford Edge	102	006	Pacific	44482	10626	11081	48966	11231	44565	34671
5	Ford Interceptor	103	004	Southwest	12814	41459	22946	21307	22108	37476	10129
6	Chevy Tahoe	100	003	Midwest	38508	48444	37828	41543	14423	15271	20857
7	Ford Crown Victoria	101	006	Pacific	29040	34048	43240	32870	41051	46881	35524
8	Ford Edge	102	005	Mountain	41502	49307	14787	16835	22399	26342	32145
9	Ford Interceptor	103	005	Mountain	29186	48846	41684	22734	23982	19729	35538
10	Chevy Tahoe	100	001	Northwest	48127	32579	30853	10860	28932	42381	49115

= OFFSET (E2, 0 , 0 , 9 , 7)

*Height should expand/contract
up and down
as data added/removed*

*Use:
COUNT(
MIN(
MAX(
ISNUM(
ISBLANK(
MATCH(*

*to determine the size and shape
of the data set*

OFFSET FUNCTION

	A	B	C	D	E	F	G	H	I	J	K
1	Vehicle Make	Vehicle ID	Region	Region Name	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19	Jul-19
2	Chevy Tahoe	100	001	Northwest	24545	32642	10755	21228	17812	48349	11509
3	Ford Crown Victoria	101	006	Pacific	17878	15331	12006	30310	39294	18790	25301
4	Ford Edge	102	006	Pacific	44482	10626	11081	48966	11231	44565	34671
5	Ford Interceptor	103	004	Southwest	12814	41459	22946	21307	22108	37476	10129
6	Chevy Tahoe	100	003	Midwest	38508	48444	37828	41543	14423	15271	20857
7	Ford Crown Victoria	101	006	Pacific	29040	34048	43240	32870	41051	46881	35524
8	Ford Edge	102	005	Mountain	41502	49307	14787	16835	22399	26342	32145
9	Ford Interceptor	103	005	Mountain	29186	48846	41684	22734	23982	19729	35538
10	Chevy Tahoe	100	001	Northwest	48127	32579	30853	10860	28932	42381	49115

= OFFSET (E2, 0 , 0 , 9 , 7)

*Width should expand/contract
left and right
as data added/removed*

OFFSET FUNCTION

	A	B	C	D	E	F	G	H	I	J	K
1	Vehicle Make	Vehicle ID	Region	Region Name	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19	Jul-19
2	Chevy Tahoe	100	001	Northwest	24545	32642	10755	21228	17812	48349	11509
3	Ford Crown Victoria	101	006	Pacific	17878	15331	12006	30310	39294	18790	25301
4	Ford Edge	102	006	Pacific	44482	10626	11081	48966	11231	44565	34671
5	Ford Interceptor	103	004	Southwest	12814	41459	22946	21307	22108	37476	10129
6	Chevy Tahoe	100	003	Midwest	38508	48444	37828	41543	14423	15271	20857
7	Ford Crown Victoria	101	006	Pacific	29040	34048	43240	32870	41051	46881	35524
8	Ford Edge	102	005	Mountain	41502	49307	14787	16835	22399	26342	32145
9	Ford Interceptor	103	005	Mountain	29186	48846	41684	22734	23982	19729	35538
10	Chevy Tahoe	100	001	Northwest	48127	32579	30853	10860	28932	42381	49115

= OFFSET (E2, 0 , 0 , 9 , 7)

***This formula should be named
(Ex: DYNAMIC_VEHICLE_DATA)***

And can be dynamically referenced EVERYWHERE in the model

When To Use Dynamic Tables vs. OFFSET

Dynamic Tables

- When data is symmetrical and complete with no skipped lines or gaps
- When entire columns will be referenced
- When data is expected to be appended at the bottom
- When there is no data series

OFFSET Function

- When data is asymmetrical and may have skipped lines and gaps
- When entire columns and/or rows will be referenced
- When data is expected to be added to the right or left
- When there is a data series

DYNAMIC INDEX RANGES

Why You Should Use Them:

Allows for the dynamic referencing of expanding and contracting data ranges prompted by the user's input

INDEX

INDEX finds a corresponding value in a table or a range by coordinate

INDEX(array, row_index_num, col_index_num)

MATCH(lookup_value, lookup_array, [type])

MATCH

MATCH finds a corresponding location in a table or a range

Dynamic INDEX MATCH Ranges

	B	C	D	E	F	G	H	I	J	K	L	M
1												
2	Financial Forecast (in 000s)											Total Range
3	<i>Active Scenario: Scenario A</i>								Jan-20 -	Jan-20		
4	(in millions)	Stub Period	Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20	Jun-20	Jun-20		
5												
6												
7	Beg. Unpaid Principal Balance		47,623	47,623	53,814	60,810	68,715	77,648		47,623		
8	+ Originations (Not Sold)		28,574	28,574	32,288	36,486	41,229	46,589		213,740		
9	+ Transfer / Purchases of Loans		3,000	3,000	3,000	3,000	3,000	3,000		18,000		
10	- Sale of Loans		(1,611)	(19,269)	(16,704)	(19,274)	-	-		(56,858)		
11	- Net-Charge-Offs		(1,905)	(1,905)	(2,153)	(2,432)	(2,749)	(3,106)		(14,249)		
12	- Misc. Servicing Fees		(800)	(804)	(808)	(812)	(816)	(820)		(4,860)		
13	- Principal Paid Down		(27,258)	(3,405)	(8,628)	(9,062)	(31,731)	(35,568)		(115,653)		
14	End. Unpaid Principal Balance		47,623	53,814	60,810	68,715	77,648	87,742		87,742		

Dynamic INDEX MATCH Ranges

	B	C	D	E	F	G	H	I	J	K	L	M
1												
2	Financial Forecast (in 000s)		Dynamic Header									Total Range
3	<i>Active Scenario: Scenario A</i>									Jan-20 -	Jan-20	
4	(in millions)	Stub Period	Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20		Jun-20	Jun-20	
5												
6												
7		Beg. Unpaid Principal Balance	47,623	47,623	53,814	60,810	68,715	77,648		47,623		
8		+ Originations (Not Sold)	28,574	28,574	32,288	36,486	41,229	46,589		213,740		
9		+ Transfer / Purchases of Loans	3,000	3,000	3,000	3,000	3,000	3,000		18,000		
10		- Sale of Loans	(1,611)	(19,269)	(16,704)	(19,274)	-	-		(56,858)		
11		- Net-Charge-Offs	(1,905)	(1,905)	(2,153)	(2,432)	(2,749)	(3,106)		(14,249)		
12		- Misc. Servicing Fees	(800)	(804)	(808)	(812)	(816)	(820)		(4,860)		
13		- Principal Paid Down	(27,258)	(3,405)	(8,628)	(9,062)	(31,731)	(35,568)		(115,653)		
14		End. Unpaid Principal Balance	47,623	53,814	60,810	68,715	77,648	87,742		87,742		

Dynamic SUM

Dynamic INDEX MATCH Ranges

	B	C	D	E	F	G	H	I	J	K	L	M
1												
2	Financial Forecast (in 000s)		="Active Scenario: "&Assumptions!\$C\$3									Total Range
3	Active Scenario: Scenario A									Jan-20 -	Jan-20	
4	(in millions)	Stub Period	Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20		Jun-20	Jun-20	
5												
6												
7	Beg. Unpaid Principal Balance		47,623	47,623	53,814	60,810	68,715	77,648		47,623		
8	+ Originations (Not Sold)		28,574	28,574	32,288	36,486	41,229	46,589		213,740		
9	+ Transfer / Purchases of Loans		3,000	3,000	3,000	3,000	3,000	3,000		18,000		
10	- Sale of Loans		(1,611)	(19,269)	(16,704)	(19,274)	-	-		(56,858)		
11	- Net-Charge-Offs		(1,905)	(1,905)	(2,153)	(2,432)	(2,749)	(3,106)		(14,249)		
12	- Misc. Servicing Fees		(800)	(804)	(808)	(812)	(816)	(820)		(4,860)		
13	- Principal Paid Down		(27,258)	(3,405)	(8,628)	(9,062)	(31,731)	(35,568)		(115,653)		
14	End. Unpaid Principal Balance		47,623	53,814	60,810	68,715	77,648	87,742		87,742		

=SUM(INDEX(D12:I12,1,MATCH(M3,D4:I4,0)):INDEX(D12:I12,1,MATCH(M4,D4:I4,0)))



INDIRECT FUNCTION



Why You Should Use It:

Allows for the reference of data, cells and worksheets without requiring a physical link

Physical Link

Returns the contents of a cell based upon a physical formula

INDIRECT

Returns the contents of a cell based upon a text string reference

INDIRECT(ref_text)

INDIRECT FUNCTION

	A	B	C	D
1		Costs 2018		
2		Month	Year	Value
3		Jan	2018	125,345
4		Feb	2018	132,577
5		Mar	2018	133,735
6		Apr	2018	126,878
7		May	2018	188,134
8		Jun	2018	170,189
9		Jul	2018	154,859
10		Aug	2018	190,718
11		Sep	2018	189,525
12		Oct	2018	172,041
13		Nov	2018	120,835
14		Dec	2018	192,862

	D	E	F
1			Physical link
2		= 'Costs 2018'!D3	125,345
3		=INDIRECT("Costs 2018!"&"D3")	125,345
4		=INDIRECT(""&D6&"!"&"D3")	125,345
5			
6		Costs 2018	Dynamic INDIRECT
7		Costs 2018	
8		Costs 2019	
9		Costs 2020	

Text String

Costs 2018	Costs 2019	Costs 2020
------------	------------	------------



DYNAMIC ARRAYS



Why You Should Use Them:

Allows for various expansion of arrays and spills of dynamic data and ranges



Dynamic Arrays

Introduced in Summer 2020

- ✓ Returns arrays (or ranges) of results rather than a single result
- ✓ Accomplishes tasks that once required extremely complex or nuanced formulas

Examples

SEQUENCE: Generates a sequence of numbers in a range of rows and/or columns

UNIQUE: extracts unique items from a range of cells/table

FILTER: Filters a range of cells/table based on input criteria

SORT: Sorts a list by a specified other column in that list

SORTBY: Sorts a list by a second list

RANDARRAY: Generates a range of random numbers in an array

Dynamic Arrays

Can we build formulas that allow this model to expand and update automatically with minimal manual involvement?

6 months							
Month number	1	2	3	4	5	6	
Date	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19	
Year	2019	2019	2019	2019	2019	2019	
Month	1	2	3	4	5	6	
Day	31	28	31	30	31	30	

SEQUENCE FUNCTION

SEQUENCE (rows , [columns] , [start] , [step])

Rows: the number of rows to return

Columns: the number of columns to return

Start: the first number in the sequence

Step: the amount to increment each subsequent value in the sequence

SEQUENCE FUNCTION

B2 *fx* =SEQUENCE(1,\$A\$1,1,1)

	A	B	C	D	E	F	G
1	6 months						
2	Month number	1	2	3	4	5	6
3	Date	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19
4	Year	2019	2019	2019	2019	2019	2019
5	Month	1	2	3	4	5	6
6	Day	31	28	31	30	31	30

SEQUENCE (rows , [columns] , [start] , [step])

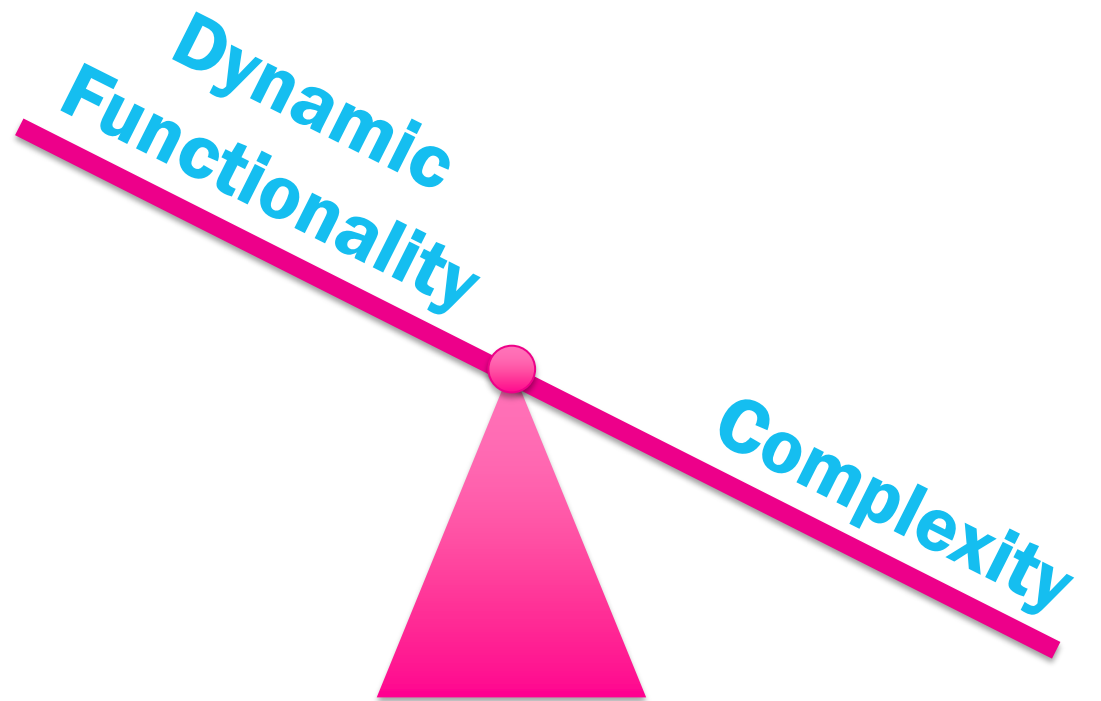
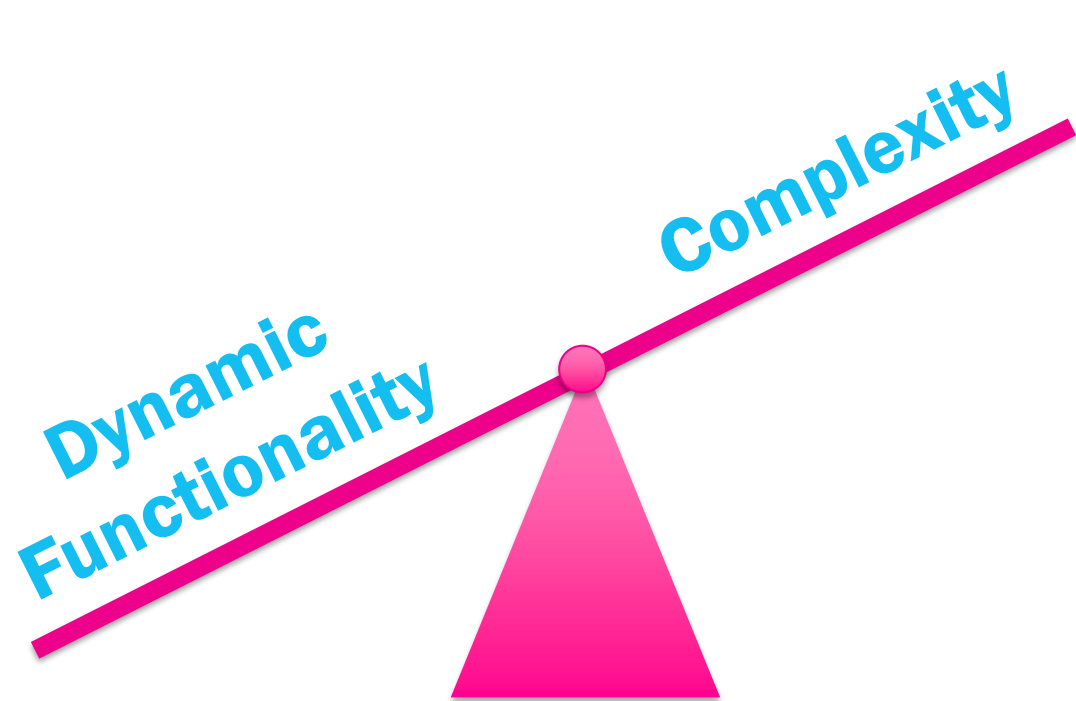
Dynamic (#) Arrays

Formulas with a hash (#) mark allow for expansion of an array driven by a precedent array

	A	B	C	D	E	F	G
1	6 months						
2	Month number	1	2	3	4	5	6
3	Date	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19
4	Year	2019	2019	2019	2019	2019	2019
5	Month	1	2	3	4	5	6
6	Day	31	28	31	30	31	30

One Final Thought

**Doesn't dynamic functionality
increase complexity for end-users?**



Remember: The objective of financial modeling isn't building a model

It's about providing end-users good information

So they can make better decisions

**Don't avoid great functionality
because it's novel for end-users**

**Don't make models so complex
that they lose the end-users**

DO develop your model-builders and end-users to:

**Intentionally
Scale**

**Incorporate
New
Data Easily**

**Update
Seamlessly**

**Minimize
Manual
Involvement**



**AFP
2020**

**VIRTUAL
EXPERIENCE**
10/19-10/29

Contact and Support

Carl Seidman

carl@seidmanfinancial.com