Demo 07.02a Development (Problem)

Reading: Friedland 07 (Development Technique)
Model: Minor random variation around an otherwise stable development pattern.
Problem Type: Paid Claims Development Example

Find Calculate ultimate claims for all accident years using data as of year-end 2023

Given

cumulative paid claims

AY	**12**	**24**	**36**	**48**
2020	48.1	141.2	200.7	240.0
2021	47.4	140.5	201.0	
2022	48.2	139.6		
2023	48.0			

Assume no development past 48 months. In other words, the triangle is fully dveloped by 48 months.

The paid claims on the latest diagonal are in brown font for instructional purposes within the solution.

Demo 07.02a Development (Solution)

Step A =====> **link ratios for paid claim triangle**

AY	12-24	24-36	36-48	**48-ult**
2013	2.936	1.421	1.196	
2014	2.963	1.430		
2015	2.897			
2016				
selected	2.932	1.426	1.196	**1.000**

Tail Factor: The triangle is fully developed as of 48 months. That means the 48-ult tail factor is equal to 1.0

Step B =====> <====== Select the **average** of the values in each column

Step C =====> **calculate age-to-ultimate LDFs**

	12-ult	24-ult	36-ult	48-ult
age -> ult	4.999	1.705	1.196	**1.000**

<====== (**selected**) x (prior [age -> ult])

(calculate from right-to-left)

Step D =====> **calculate ultimate losses based on latest paid losses**

	'23@12	**'22@24**	**'21@36**	**'20@48**
diagonal	48	140	201	240
ultimate	240	238	240	240

final answers ===>

<====== (**diagonal**) x (age -> ult)

Sometimes it's nice to present the ultimates in a column to the right of the original triangle:

cumulative paid claims

AY	**12**	**24**	**36**	**48**
2020	48.1	141.2	200.7	240.0
2021	47.4	140.5	201.0	
2022	48.2	139.6		
2023	48.0			

estimated ultimates	real ultimates	% error
240	240	0%
240	240	0%
238	240	-1%
240	240	0%

Interesting side note:
This example was created using my simulation software ***SimPolicy*** *. One of the input parameters to the simulation is the value of the ultimate loss. For this simulation, each AY was given the same* ***ultimate loss of 240*** *. That means we can see how accurate our estimates are. More to the point, we can often see how* ***inaccurate*** *our estimates are regardless of how we select our LDFs (Loss Development Factors) in Step B.*

Moral:
Don't agonize for too long over selecting LDFs. In a real-life situation there will be a lot of noise or random variation that cannot be accounted for in any reserving method. Do the best you can with the information you've got but make allowances for the fact that your estimates will never be exactly right, especially for AYs at early stages of development.

This example:
Our estimates were essentially accurate. The simulation introduced a small amount of random varition, or noise, around an otherwise stable development pattern.

Demo 07.02b Development (Problem)

Reading: Friedland 07 (Development Technique)
Model: Changing development pattern A
Problem Type: Paid Claims Development Example

Find Calculate ultimate claims for all accident years using data as of year-end 2023

Given

cumulative paid claims

AY	12	24	36	48
2020	48.0	140.4	200.7	240.0
2021	43.6	136.0	198.6	
2022	40.0	132.1		
2023	36.9			

Assume no development past 48 months. In other words, the triangle is fully dveloped by 48 months.

The paid claims on the latest diagonal are in brown font for instructional purposes within the solution.

Demo 07.02b Development (Solution)

Step A =====> **link ratios for paid claim triangle**

	AY	12-24	24-36	36-48	48-ult	
	2013	2.924	1.430	1.196		
	2014	3.116	1.461			
	2015	3.302				
	2016					
Step B =====>	**selected**	3.209	1.445	1.196	**1.000**	<====== Select the **average** of the values in each column

Tail Factor: The triangle is fully developed as of 48 months. That means the 48-ult tail factor is equal to 1.0

Step C =====> **calculate age-to-ultimate LDFs**

		12-ult	24-ult	36-ult	48-ult	
	age -> ult	5.546	1.728	1.196	**1.000**	<====== (**selected**) x (prior [age -> ult])

(calculate from right-to-left)

Step D =====> **calculate ultimate losses based on latest paid losses**

		'23@12	**'22@24**	**'21@36**	**'20@48**	
	diagonal	37	132	199	240	
final answers ===>	ultimate	205	228	237	240	<====== (**diagonal**) x (age -> ult)

Sometimes it's nice to present the ultimates in a column to the right of the original triangle:

cumulative paid claims

AY	**12**	**24**	**36**	**48**
2020	48.0	140.4	200.7	240.0
2021	43.6	136.0	198.6	
2022	40.0	132.1		
2023	36.9			

estimated ultimates	real ultimates	% error
240	240	0%
237	240	-1%
228	240	-5%
205	240	-15%

Interesting side note:

This example was created using my simulation software ***SimPolicy*** *. One of the input parameters to the simulation is the value of the ultimate loss. For this simulation, each AY was given the same* ***ultimate loss of 240*** *. That means we can see how accurate our estimates are. More to the point, we can often see how* ***inaccurate*** *our estimates are regardless of how we select our LDFs (Loss Development Factors) in Step B.*

Moral:

Don't agonize for too long over selecting LDFs. In a real-life situation there will be a lot of noise or random variation that cannot be accounted for in any reserving method. Do the best you can with the information you've got but make allowances for the fact that your estimates will never be exactly right, especially for AYs at early stages of development.

This example:

Our estimates are ***no longer accurate*** *. This is because the development changes for each successive AY. Selecting the average of prior age-to-age factors in Step B is not good judgment. We will examine ways of correcting for this inconsistency in the dvelopment pattern but one easy way to improve the estimates is to recognize the upward trend in age-to-age factors as you move down the column. A better choice for the 12-24 selection in Step B would be the last value in the column instead of the average.*

(Or if you truly believe there is an underlying trend then select an LDF that follows the trend, something like 3.50. But that's risky because trends are rarely so obvious. It's likely that at least some of the apparent trend is due to random variation or noise.)

Anyway, selecting the average of the latest 2 values in the 12-24, which is x.xxx (versus 3.114) gives this slightly: improved estimate for AY 2023:

AY	estimated ultimates	real ultimates	% error	
2020	240	240	0%	
2021	237	240	-1%	
2022	228	240	-5%	
2023	205	240	-15%	<====== *recognizing the upward trend in the 12-24 column produced a slight improvement*

Demo 07.02c Development (Problem)

Reading: Friedland 07 (Development Technique)

Model: Moderate random variation around an otherwise stable development pattern.

Problem Type: Paid Claims Development Example

Find Calculate ultimate claims for all accident years using data as of year-end 2023

Given

cumulative paid claims

AY	12	24	36	48
2020	42.4	145.7	202.3	240.0
2021	56.1	144.0	205.3	
2022	52.1	137.2		
2023	42.2			

Assume no development past 48 months. In other words, the triangle is fully dveloped by 48 months.

The paid claims on the latest diagonal are in brown font for instructional purposes within the solution.

Demo 07.02c Development (Solution)

Step A =====> **link ratios for paid claim triangle**

AY	12-24	24-36	36-48	**48-ult**
2013	3.436	1.388	1.186	
2014	2.568	1.426		
2015	2.636			
2016				
selected	2.880	1.407	1.186	**1.000**

Step B =====> (selected row) <====== Select the **average** of the values in each column

Tail Factor: The triangle is fully developed as of 48 months. That means the 48-ult tail factor is equal to 1.0

Step C =====> **calculate age-to-ultimate LDFs**

	12-ult	24-ult	36-ult	48-ult
age -> ult	4.808	1.670	1.186	**1.000**

<====== (**selected**) x (prior [age -> ult])

(calculate from right-to-left)

Step D =====> **calculate ultimate losses based on latest paid losses**

	'23@12	**'22@24**	**'21@36**	**'20@48**
diagonal	42	137	205	240
ultimate	203	229	244	240

final answers ===> (ultimate row)

<====== (**diagonal**) x (age -> ult)

Sometimes it's nice to present the ultimates in a column to the right of the original triangle:

cumulative paid claims

AY	**12**	**24**	**36**	**48**
2020	42.4	145.7	202.3	240.0
2021	56.1	144.0	205.3	
2022	52.1	137.2		
2023	42.2			

estimated ultimates	real ultimates	% error
240	240	0%
244	240	2%
229	240	-5%
203	240	-15%

Interesting side note:

This example was created using my simulation software ***SimPolicy*** *. One of the input parameters to the simulation is the value of the ultimate loss. For this simulation, each AY was given the same* ***ultimate loss of 240*** *. That means we can see how accurate our estimates are. More to the point, we can often see how* ***inaccurate*** *our estimates are regardless of how we select our LDFs (Loss Development Factors) in Step B.*

Moral:

Don't agonize for too long over selecting LDFs. In a real-life situation there will be a lot of noise or random variation that cannot be accounted for in any reserving method. Do the best you can with the information you've got but make allowances for the fact that your estimates will never be exactly right, especially for AYs at early stages of development.

This example:

This is tricky. The simulation used a consistent development pattern for each AY but introduced a high degree of random variation, or noise. That makes it hard to tell whether or not the development pattern is stable. This often happens in real situations but unfortunately there you don't have the benefit of knowing the true underlying development pattern or what the real ultimate losses are.

It looks like selecting the average of the latest 2 (instead of 3) values in the 12-24 column might be better, but in fact, it makes the final estimates worse. Here's the result of choosing 2.602 *(instead of 2.880)*

AY	estimated ultimates	real ultimates	% error
2020	240	240	0%
2021	244	240	2%
2022	229	240	-5%
2023	183	240	-24%

<====== *AY 2023 estimate got* ***worse*** *even though the modified LDF selection looked more reasonable!*

Note that the prior AYs didn't change. That's because the 12-24 selection only affects the latest AY.

Demo 07.02d Development (Problem)

Reading:	Friedland 07 (Development Technique)
Model:	Changing development pattern B
Problem Type:	Paid Claims Development Example

Find Calculate ultimate claims for all accident years using data as of year-end 2023

Given

cumulative paid claims

AY	**12**	**24**	**36**	**48**
2020	46.8	144.3	196.9	240.0
2021	53.3	145.4	203.1	
2022	60.0	151.2		
2023	68.6			

Assume no development past 48 months. In other words, the triangle is fully dveloped by 48 months.

The paid claims on the latest diagonal are in brown font for instructional purposes within the solution.

Demo 07.02d Development (Solution)

Step A =====> **link ratios for paid claim triangle**

Step B =====> (selected row) <====== Select the **average** of the values in each column

AY	12-24	24-36	36-48	**48-ult**
2013	3.085	1.364	1.219	
2014	2.726	1.397		
2015	2.520			
2016				
selected	2.777	1.381	1.219	**1.000**

Tail Factor: The triangle is fully developed as of 48 months. That means the 48-ult tail factor is equal to 1.0

Step C =====> **calculate age-to-ultimate LDFs**

	12-ult	24-ult	36-ult	48-ult
age -> ult	4.673	1.683	1.219	**1.000**

<====== (**selected**) x (prior [age -> ult])
(calculate from right-to-left)

Step D =====> **calculate ultimate losses based on latest paid losses**

	'23@12	**'22@24**	**'21@36**	**'20@48**
diagonal	69	151	203	240
ultimate	320	254	248	240

final answers ===> (ultimate row)

<====== (**diagonal**) x (age -> ult)

Sometimes it's nice to present the ultimates in a column to the right of the original triangle:

cumulative paid claims

AY	**12**	**24**	**36**	**48**
2020	46.8	144.3	196.9	240.0
2021	53.3	145.4	203.1	
2022	60.0	151.2		
2023	68.6			

estimated ultimates	real ultimates	% error
240	240	0%
248	240	3%
254	240	6%
320	240	34%

Interesting side note:
This example was created using my simulation software ***SimPolicy***. *One of the input parameters to the simulation is the value of the ultimate loss. For this simulation, each AY was given the same* ***ultimate loss of 240***. *That means we can see how accurate our estimates are. More to the point, we can often see how* ***inaccurate*** *our estimates are regardless of how we select our LDFs (Loss Development Factors) in Step B.*

Moral:
Don't agonize for too long over selecting LDFs. In a real-life situation there will be a lot of noise or random variation that cannot be accounted for in any reserving method. Do the best you can with the information you've got but make allowances for the fact that your estimates will never be exactly right, especially for AYs at early stages of development.

This example:
This is similar to an example from above where there appeared to be a trend in the LDFs as you scan down the column. Here the trend is ***downward*** *rather than upward. As discussed previously, a better LDF selection might be the latest value in the first column instead of the average. If you select 2.623 (instead of 2.777) you get:*

AY	estimated ultimates	real ultimates	% error
2020	240	240	0%
2021	248	240	3%
2022	254	240	6%
2023	303	240	26%

<====== *recognizing the downward trend in the 12-24 column improved our estimate*

Note that the prior AYs didn't change. That's because the 12-24 selection only affects the latest AY.

Demo 07.02e Development (Problem)

Reading: Friedland 07 (Development Technique)
Model: Stable Development Pattern with Increase in Paid Loss across AYs
Problem Type: Paid Claims Development Example

Find Calculate ultimate claims for all accident years using data as of year-end 2023

Given

cumulative paid claims

AY	**12**	**24**	**36**	**48**
2020	48	140	201	240
2021	52	152	217	
2022	56	164		
2023	60			

Assume no development past 48 months. In other words, the triangle is fully dveloped by 48 months.

The paid claims on the latest diagonal are in brown font for instructional purposes within the solution.

Demo 07.02e Development (Solution)

Step A =====> **link ratios for paid claim triangle**

AY	12-24	24-36	36-48	**48-ult**
2013	2.924	1.430	1.196	
2014	2.924	1.430		
2015	2.924			
2016				
selected	2.924	1.430	1.196	**1.000**

Step B =====>

Tail Factor: The triangle is fully developed as of 48 months. That means the 48-ult tail factor is equal to 1.0

Step C =====> **calculate age-to-ultimate LDFs**

	12-ult	24-ult	36-ult	48-ult
age -> ult	5.000	1.710	1.196	**1.000**

<====== (**selected**) x (prior [age -> ult])

(calculate from right-to-left)

Step D =====> **calculate ultimate losses based on latest paid losses**

	'23@12	**'22@24**	**'21@36**	**'20@48**
diagonal	60	164	217	240
ultimate	300	280	260	240

final answers ===>

<====== (**diagonal**) x (age -> ult)

Sometimes it's nice to present the ultimates in a column to the right of the original triangle:

cumulative paid claims

AY	**12**	**24**	**36**	**48**
2020	48	140	201	240
2021	52	152	217	
2022	56	164		
2023	60			

estimated ultimates	real ultimates	% error
240	240	0%
260	260	0%
280	280	0%
300	300	0%

Interesting side note: ***(Simulated ultimates are different in this example!)***
This example was created using my simulation software ***SimPolicy*** *. One of the input parameters to the simulation is the value of the ultimate loss. For this simulation, the* ***ultimate losses are as in the table above*** *. Each AY has a different value.*

This example:
Although the paid losses increase from AY to AY, the development pattern is exactly the same for each AY. That means the key assumption of stability/consistency is satisfied and the development method will produce the correct ultimates.