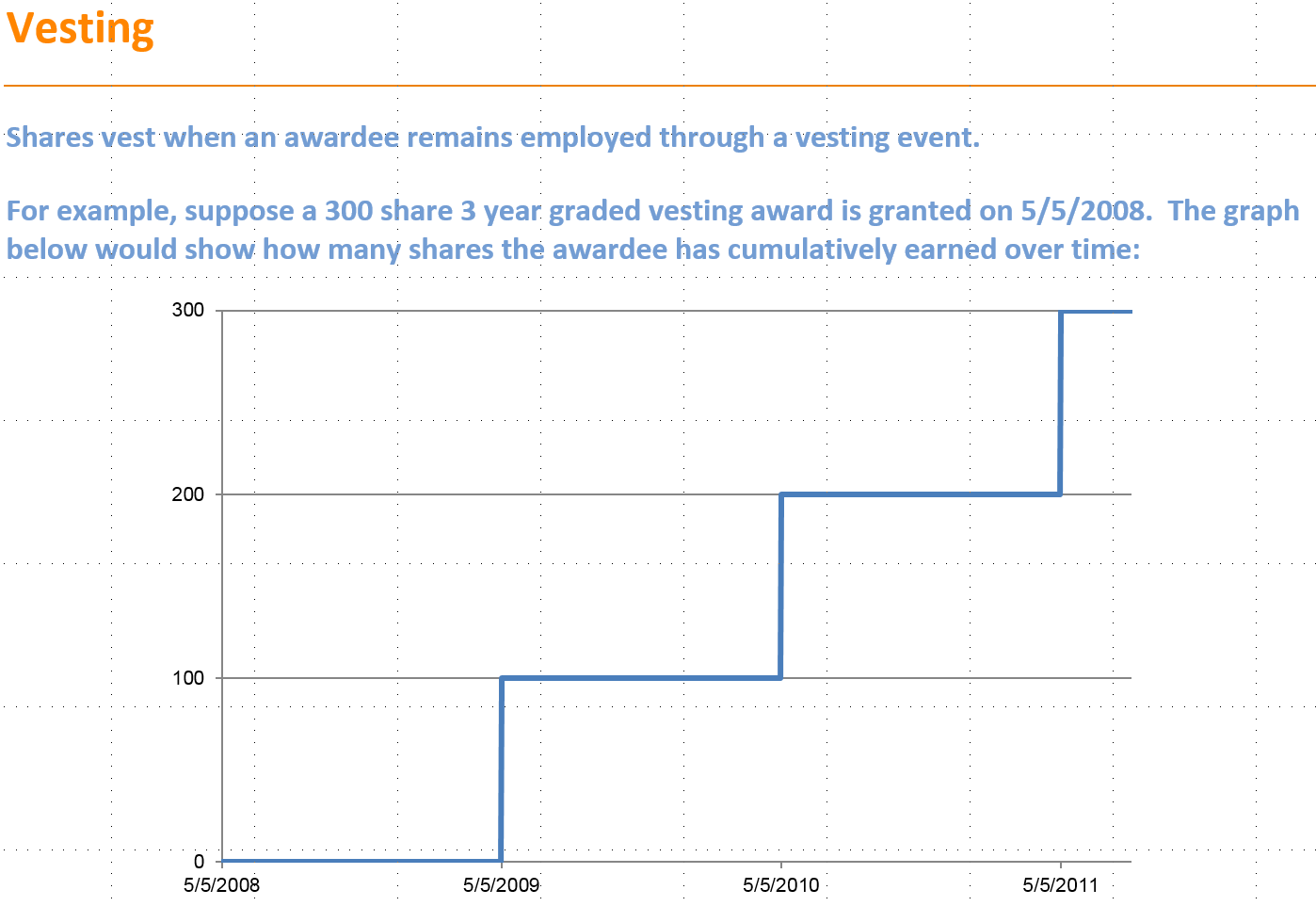
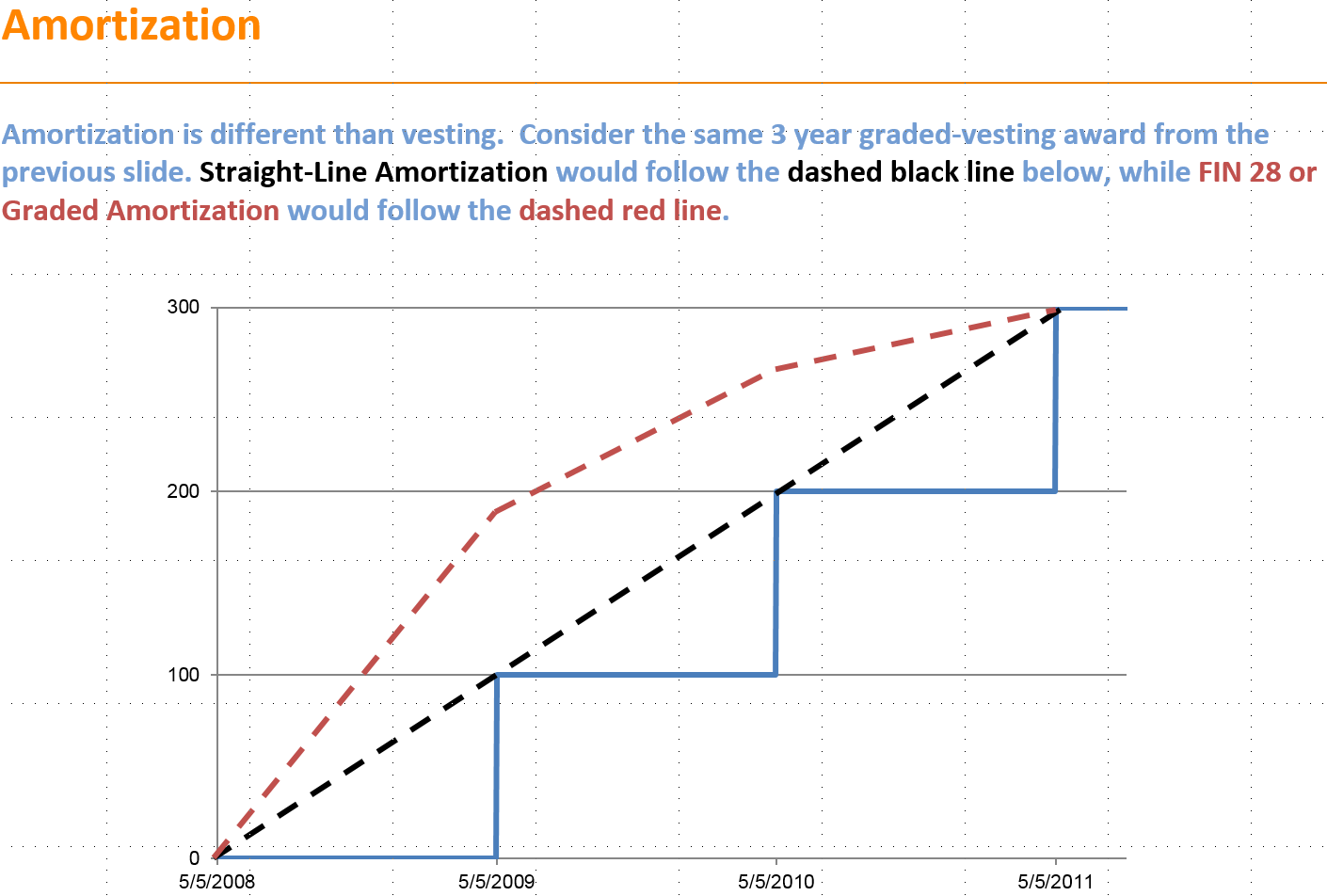
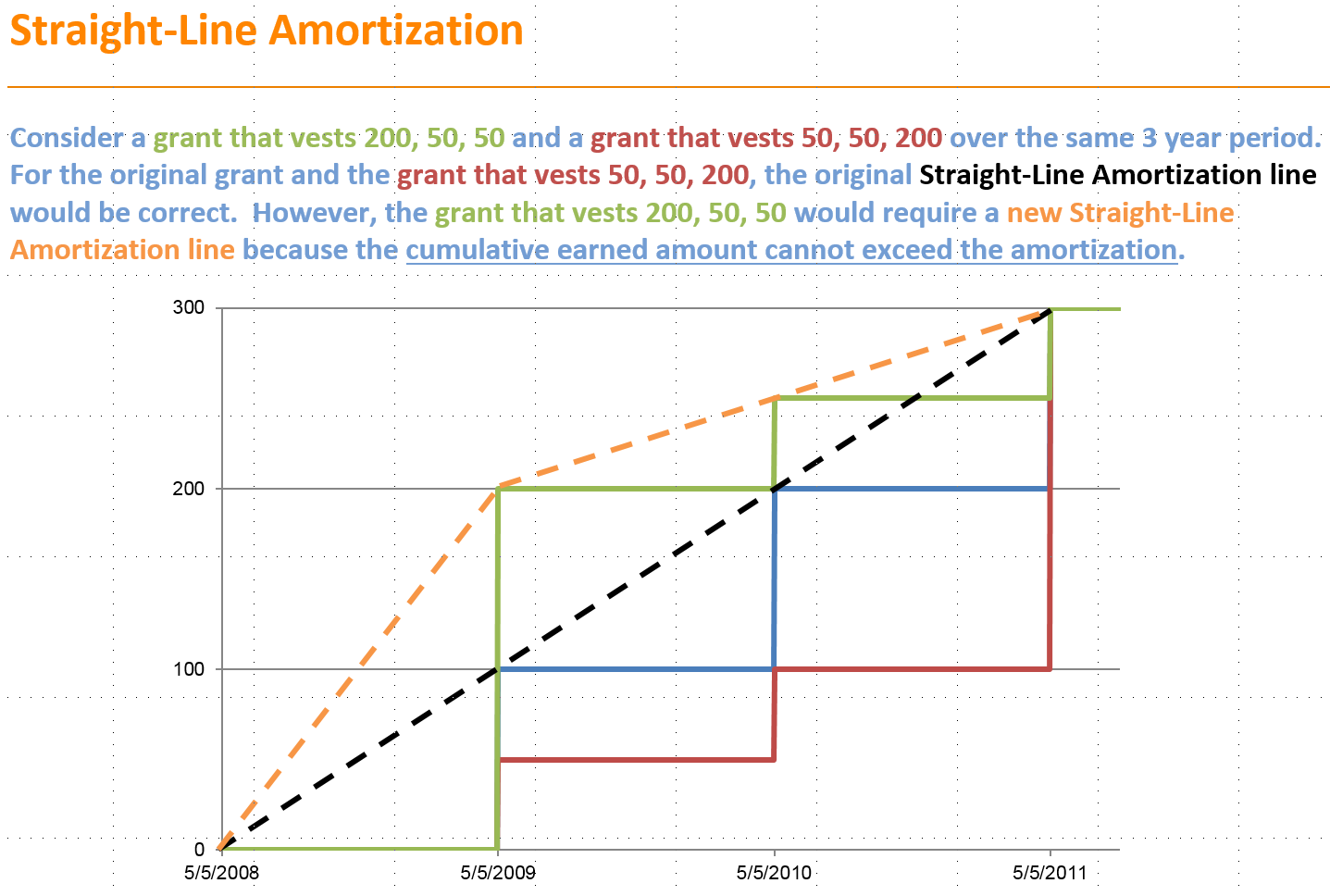
Amortization Specification 1

# Background

The grants file is the master record of a stock-based compensation award. A ‘grant’ is a set of shares contained in an award agreement (like a contract) that have the same set of terms and conditions. Grants are traded with an employee in exchange for their services at the company. A single grant can have multiple parts, called tranches (French word for sub-component). Amortization is about spreading value of the shares in the tranches over time. But, the first step is all about figuring out how. “Straight-line” amortization is what we are trying to do. The rule is that you take the value evenly over time, **\*but\*** the cumulative amortization line can never be below the cumulative earning line. Here are a few pictures that illustrate:



Straight-line is really a misnomer, as the orange amortization line really are two straight lines (we call these “straight-line segments”). A few descriptive words to mention: the red grant would be considered “back-loaded,” as it has more shares near the end (and the entire grant can be amortized with one “straight-line segment”). The green grant would be considered “front-loaded,” as it has more shares near the front of the grant (and needs >1 “straight-line segment”). Also, “requisite service period” or “RSP” are the days over which we spread the expense.

# English Description of Process

The goal is to allocate each tranche to a straight-line segment. The number of straight-line segments is at least 1 (“fully back-loaded”) and is at most the number of tranches (“fully front-loaded”). Here is a description of the process:

* Calculate the cumulative slopes (shares times value over time for just tranche 1, then tranche 1 and 2, then tranche 1, 2 and 3, etc)
* Find the maximum cumulative slope starting from the beginning of the grant
* Allocate the tranches that were used to calculate the maximum cumulative slope to that straight-line segment
* If you have not reached the last tranche, recalculate the cumulative slopes starting from the next tranche
* Repeat until you reach the end of the grant

# Sample Logic (very inefficient – could have collapsed the sections, and I left the error checking in)

do count\_vests=**1** to &num\_vests;

if \_trancheID[count\_vests] = **.** then leave;

end;

num\_vests = count\_vests - **1**;

current\_segment = **1**;

cumulative\_value[**1**] = **0**;

cumulative\_RSP[**1**] = **0**;

do n\_vest=**1** to num\_vests;

if n\_vest = **1** then \_effective\_RSP\_start\_PED[n\_vest] = \_effective\_RSP\_start\_PED[n\_vest] - **1**;

tranche\_RSP[n\_vest] = max(**0**, \_effective\_RSP\_end\_PED[n\_vest] - \_effective\_RSP\_start\_PED[n\_vest]);

cumulative\_value[n\_vest] = cumulative\_value[max(n\_vest-**1**,**1**)] + \_Est\_tranche\_value\_PED[n\_vest];

cumulative\_RSP[n\_vest] = cumulative\_RSP[max(n\_vest-**1**,**1**)] + tranche\_RSP[n\_vest];

if cumulative\_RSP[n\_vest] = **0** then slope[n\_vest] = **0**;

else slope[n\_vest] = cumulative\_value[n\_vest] / cumulative\_RSP[n\_vest];

if slope[n\_vest] ne **.** and slope[n\_vest] < -**0.001** then do;

description = "Slope is negative";

output \_sl\_segment\_assignment\_erros;

end;

if cumulative\_RSP[n\_vest] ne **.** and cumulative\_RSP[n\_vest] < -**0.001** then do;

description = "Cumulative RSP is negative";

output \_sl\_segment\_assignment\_erros;

end;

if cumulative\_value[n\_vest] ne **.** and cumulative\_value[n\_vest] < -**0.001** then do;

description = "Cumulative Value is negative";

output \_sl\_segment\_assignment\_erros;

end;

end;

max\_slope = max(of slope[\*], **0**);

start\_tranche = **1**;

%\* find tranche of last tranche with slope equal to max slope;

do reverse\_vest=num\_vests to start\_tranche by -**1** until(slope[reverse\_vest] = max\_slope);

sl\_segment\_end\_tranche = reverse\_vest;

end;

sl\_segment\_end\_date = \_effective\_RSP\_end\_PED[sl\_segment\_end\_tranche];

do n\_vest = **1** to sl\_segment\_end\_tranche;

segment[n\_vest] = current\_segment;

end;

%\* find the current tranche;

do n\_vest=num\_vests to **1** by -**1**;

if \_effective\_RSP\_end\_PED[n\_vest] >= Period\_end\_date then current\_tranche = n\_vest;

end;

%\* case where grant should be fully expensed;

if current\_tranche = **.** then current\_tranche = num\_vests;

do n\_vest=**2** to num\_vests;

do i=**1** to num\_vests;

cumulative\_value[i] = **0**;

cumulative\_RSP[i] = **0**;

slope[i] = **.**;

end;

if n\_vest <= sl\_segment\_end\_tranche then continue;

else do sub\_vest = n\_vest to num\_vests;

start\_tranche = n\_vest;

cumulative\_value[sub\_vest] = cumulative\_value[max(sub\_vest-**1**,**1**)] + \_Est\_tranche\_value\_PED[sub\_vest];

cumulative\_RSP[sub\_vest] = cumulative\_RSP[max(sub\_vest-**1**,**1**)] + tranche\_RSP[sub\_vest];

if cumulative\_RSP[sub\_vest] = **0** then slope[sub\_vest] = **0**;

else slope[sub\_vest] = cumulative\_value[sub\_vest] / cumulative\_RSP[sub\_vest];

if slope[sub\_vest] ne **.** and slope[sub\_vest] < -**0.001** then do;

description = "Slope is negative";

output \_sl\_segment\_assignment\_erros;

end;

if cumulative\_RSP[sub\_vest] ne **.** and cumulative\_RSP[sub\_vest] < -**0.001** then do;

description = "Cumulative RSP is negative";

output \_sl\_segment\_assignment\_erros;

end;

if cumulative\_value[sub\_vest] ne **.** and cumulative\_value[sub\_vest] < -**0.001** then do;

description = "Cumulative Value is negative";

output \_sl\_segment\_assignment\_erros;

end;

end;

max\_slope = max(of slope[\*], **0**);

prior\_sl\_segment\_end\_tranche = sl\_segment\_end\_tranche;

do reverse\_vest=num\_vests to start\_tranche by -**1** until(slope[reverse\_vest] = max\_slope);

sl\_segment\_end\_tranche = reverse\_vest;

end;

current\_segment = current\_segment + **1**;

do n\_vest2 = (prior\_sl\_segment\_end\_tranche + **1**) to sl\_segment\_end\_tranche;

segment[n\_vest2] = current\_segment;

end;

end;

# SAS vs Python

The way of attacking the problem could be entirely different in Python vs how we have done it in SAS.

In order to figure out the amortization line in SAS, we have to take an earning schedule that looks like this:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GrantID | TrancheID | VestDate | VestShares | ShareValue |
| 1 | 1 | 5/5/2009 | 100 | 10 |
| 1 | 2 | 5/5/2010 | 100 | 10 |
| 1 | 3 | 5/5/2011 | 100 | 10 |
| 2 | 1 | 2/4/2009 | 50 | 15 |
| 2 | 2 | 2/4/2012 | 200 | 15 |
| 3 | 1 | 10/17/2015 | 100 | 12 |

And transpose it to make it look like this:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GrantID | TrancheID1 | TrancheID2 | TrancheID3 | VestDate1 | VestDate2 | VestDate3 | VestShares1 | VestShares2 | VestShares3 | ShareValue1 | ShareValue2 | ShareValue3 |
| 1 | 1 | 2 | 3 | 5/5/2009 | 5/5/2010 | 5/5/2011 | 100 | 100 | 100 | 10 | 10 | 10 |
| 2 | 1 | 2 |  | 2/4/2009 | 2/4/2012 |  | 50 | 200 |  | 15 | 15 |  |
| 3 | 1 |  |  | 10/17/2015 |  |  | 100 |  |  | 12 |  |  |

(since SAS can only ever see 1 observation at a time, and then use arrays and do-loop processing)

Things to note:

* The array width is the maximum number of tranches per grant
* Null values are possible (once we encounter one, we stop the straight-line segment allocation loop)
* We call these parallel arrays, as TrancheID1 corresponds to VestDate1, VestShares1, and ShareValue1
* Information/calculations from one grant do not affect others (each grant’s calculation could happen on a separate thread)
* You always have to start from the beginning of the grant

# Sample Inputs/Outputs

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | | | | Outputs | | | Intermediate Calcs | | |
| GrantID | TrancheID | Grant Date | Vest Date | Vest Shares | Share Value | Total Value | Tranche Days | Straight-Line Segment | 1st Cumulative Slope Calc | 2nd Cumulative Slope Calc | 3rd Cumulative Slope Calc |
| 1 | 1 | 5/5/2008 | 5/5/2009 | 50 | 2 | 100 | 366 | 1 | 0.2732 | n/a | n/a |
| 1 | 2 | 5/5/2008 | 5/5/2010 | 50 | 2 | 100 | 365 | 1 | 0.2736 | n/a | n/a |
| 1 | 3 | 5/5/2008 | 5/5/2011 | 100 | 2 | 200 | 365 | 1 | 0.3650 | n/a | n/a |
| 2 | 1 | 5/5/2008 | 5/5/2009 | 100 | 2 | 200 | 366 | 1 | 0.5464 | n/a | n/a |
| 2 | 2 | 5/5/2008 | 5/5/2010 | 50 | 2 | 100 | 365 | 2 | 0.4104 | 0.2740 | n/a |
| 3 | 1 | 5/5/2008 | 5/5/2009 | 100 | 2 | 200 | 366 | 1 | 0.5464 | n/a | n/a |
| 4 | 1 | 5/5/2008 | 5/5/2009 | 50 | 2 | 100 | 366 | 1 | 0.2732 | n/a | n/a |
| 4 | 2 | 5/5/2008 | 5/5/2010 | 100 | 2 | 200 | 365 | 1 | 0.4104 | n/a | n/a |
| 4 | 3 | 5/5/2008 | 5/5/2011 | 25 | 2 | 50 | 365 | 2 | 0.3193 | 0.1370 | n/a |
| 4 | 4 | 5/5/2008 | 5/5/2012 | 50 | 2 | 100 | 366 | 2 | 0.3078 | 0.2052 | n/a |
| 5 | 1 | 5/5/2008 | 5/5/2009 | 100 | 2 | 200 | 366 | 1 | 0.5464 | n/a | n/a |
| 5 | 2 | 5/5/2008 | 5/5/2010 | 70 | 2 | 140 | 365 | 2 | 0.4651 | 0.3836 | n/a |
| 5 | 3 | 5/5/2008 | 5/5/2011 | 50 | 2 | 100 | 365 | 3 | 0.4015 | 0.3288 | 0.2740 |