Table of Contents

Introduction

Functions 101: Growth Ratios
1. Cycle 3
2. Growth 4
3. Growth# 4
4. GrowthYTD 5
5. GrowthNCUA or GrowthYTDAnn 5
6. GrowthQtr 5
7. GrowthQtrAnn 5

Functions 201: Calculations
8. Annual 6
9. FullYear 7
10. Change 7
11. QtrChange 7
12. Booked 7
13. QtrAve 8
14. QtrAveAnn 8

Functions 301: Benchmarking
15. Rank 9
16. ReverseRank 9
17. Percentile 10
18. ReversePercentile 10
19. AtPercent 10
20. Peer 10
Introduction

What Are Functions?
Functions are commands you can use on custom displays in Peer-to-Peer to modify account codes. They are "shortcuts" for calculations that finance executives commonly use in evaluating financial performance. For example, when you type “Growth” plus an account code bracketed by parentheses, e.g. Growth(a010), Peer will calculate annual asset growth so you don't have to know the equation yourself.

Functions simplify more complex formulas by doing the work for you!

Helpful Hints

• **Have your 5300 handy.**
  You can print out the latest 5300 Call Report under the Help menu in the upper right corner of your Peer-to-Peer account.

• **Put an “a” in front of all 5300 account codes.**
  Many account codes are numeric values and peer will read any number as just that, a number. So, when building custom displays, just put an “a” in front of any account code from the NCUA 5300 Call Report.

• **Bank codes start with a “R”.**
  Peer–to–Peer uses the 8-character FFIEC call report codes, which begin with the letter "r". These are drawn from the FFIEC 041 form (Statement of condition for banks with Domestic Offices Only), and the 031 form (Statement of Condition for Banks with Domestic and Foreign Offices). Many accounts used on the 031 form are equivalent to accounts on the 041, and the last four characters of their codes are the same. Peer–to–Peer uses the 041 (domestic) version of the code in these cases, even if the bank filed the 031 call report. For example, the 031 form uses the code "RCFDM708" for real estate construction loans, while the 041 form uses "RCONM708" for the same purpose. Peer uses "RCONM708".

• **Casing doesn’t matter.**
  Functions are NOT case sensitive. We are just using mixed case letters in this document to make them easier to read and remember.

• **Terminology.**
  When a function is used with an account, we refer to this as a formula. So Growth(members) is the formula for member growth; Annual(a270) is the formula for a credit union’s annual marketing expenses. “Growth” and “annual” are the functions in these two examples.

• **You can use account codes in equations.**
  When using functions, you can add, subtract, multiply and divide account codes just like any number within the parenthesis. So, to see total auto loan growth (page 2 of the Call Report), you can type **Growth(a385 + a370)**, where a385 is new and a370 is used vehicle loans.
Functions 101: Growth Calculations

Growth is often viewed as a key indicator of success, so Peer makes it easy to calculate your growth rates in five different ways. There is also a cycle function that is essential to understanding growth calculations. This section covers the following seven functions:

1. **Cycle** – this function allows you to display historical values for a specific period.
2. **Growth** – used to calculate the year-over-year, or actual 12-month, growth rate.
3. **Growth3** – calculates the compound annual growth rate over a variety of time periods.
4. **GrowthYTD** – calculates the growth rate from the prior year-end.
5. **GrowthNCUA / GrowthYTDAnn** – the projected annual growth date based on year-to-date performance.
6. **GrowthQtr** – calculates your growth over the previous quarter.
7. **GrowthQtrAnn** – forecasts a 12 month growth rate based on the current quarter’s growth.

1. **Cycle**

We include the **Cycle** function in this section because all **Growth** function use data from prior periods in the calculation. Therefore, in order to understand how we calculate each different growth function, you need to be familiar with the cycle function.

The **Cycle** function is indicated with a colon “:” and a number, and is used to pull historical data for account codes from any time cycle. It is different from all the other functions in that it is used AFTER the account code.

Rather than making any calculations, the “:” function simply pulls data from the time period you specify. Typing “:” will denote that you want to find data from previous business cycles; the number after the colon indicates how far back in time you want to go, with each 0.25 equaling one quarter.

It is important to note that the “:” function is dependent on the cycle that you have currently selected at the top of the screen. In the screenshot below, Peer-to-Peer will calculate backward from 2Q 2015.

![Cycle Function Screenshot]

An example using total loans, or 025b on the 5300 Call Report:
- Your loan balance from the previous quarter = a025b:0.25
- Your loan balance six months ago = a025b:0.5
- Your loan balance nine months ago = a025b:0.75
- Your loan balance for the same quarter last year = a025b:1

Remember, in our software products you must put an “a” in front of account codes from the 5300.

You may also apply the cycle operator to an entire formula as long as you enclose the formula in parentheses. For example,

(loans+investments+cash):3 is equivalent to: loans:3+investments:3+cash:3
There is one additional special cycle character you may encounter in growth calculations – :dec. This function always selects the value from the previous year’s end. If you want to look at data from last year’s end, use a396:dec.

If you don’t want to calculate how far back to go, you could use :2Q16 instead (for example) to pull data from the second quarter of 2016. It can be used with formulas, as well as individual account codes and defined facts. For example, total assets from the first quarter of 2015 = Assets:1Q15

One drawback of this notation is that the displays would need to be modified in future quarters to produce equivalent results relative to those quarters. This is more useful as a quick shortcut for temporary displays.

2. Growth
The Growth function is used to calculate year-over-year, or actual 12-month growth. Since it is calculating the rate of growth, it is displayed as a percentage and typically used on line charts or in tables. It’s the most commonly used function in Peer-to-Peer!

This, and all other functions with the exception Cycle, are placed in front of an account code or codes in parentheses. So, to calculate the actual 12-month growth of loans you type Growth(a025b) or Growth(loans)¹.

The growth function calculates the rate of change by taking the difference between the results today and one year ago and dividing by the value one year ago. It saves you from having to type out the full formula using cycle functions as such:

\[
\text{Growth}(a025b) \text{ or } \text{Growth}(loans) = \frac{(a025b - a025b:1)}{a025b:1} = \frac{(loans - loans:1)}{loans:1}
\]

You can also use Growth on ratios. We already have a shortcut for return on assets which is just “ROA”. By typing Growth(ROA), you are asking for the percentage growth rate of your ROA compared to this time last year.

**FAQ – Why is your growth different from NCUA’s FPR report?**
NCUA calculates growth differently on their Financial Performance Reports (FPR). They are projecting an annual growth rate using your year-to-date growth rate. There is a function that uses the same calculation, GrowthNCUA, described on page 5.

3. Compound Annual Growth Rates (CAGR) – Growth#
The standard Growth function can also be manipulated to calculate compound growth over 2, 3, 4, 5 or 10 year periods just by adding the number after Growth. So, to calculate your membership’s 3-year CAGR, you type Growth3(members) or Growth3(a083b).

A compound annual growth rate, or CAGR, is an average of how quickly a statistic has grown each year for a given period of time. Investopedia describes it as such: “It is essentially an imaginary number that describes the rate at which an investment would have grown if it had grown at a steady rate, which virtually never happens in reality.”

¹ “Loans” is a defined fact, or shortcut, included in Peer for commonly used accounts. See our Defined Facts Manual for more information.
Peer calculates compound growth by taking the ratio of the results today over the prior period results, to the nth root, with n being the number of years into the past you have specified. That number is then subtracted by 1 to give a percentage.

- 2-year compound annual growth: \( \text{Growth2}(a083) = (a083/a083:2)^{(1/2)} - 1 \)
- 3-year compound annual growth: \( \text{Growth3}(a083) = (a083/a083:3)^{(1/3)} - 1 \)
- 4-year compound annual growth: \( \text{Growth4}(a083) = (a083/a083:4)^{(1/4)} - 1 \)
- 5-year compound annual growth: \( \text{Growth5}(a083) = (a083/a083:5)^{(1/5)} - 1 \)
- 10-year compound annual growth: \( \text{Growth10}(a083) = (a083/a083:10)^{(1/10)} - 1 \)

4. GrowthYTD

GrowthYTD, also read as "year-to-date growth", measures your rate of growth since the beginning of the year. It calculates the rate of change by taking the difference between the results of the current cycle and the prior year-end results, then dividing by the prior year-end results.

\[
\text{GrowthYTD}(a025b) = \frac{(a025b - a025b:dec)}{a025b:dec}
\]

Note the use of the cycle function :dec in this formula rather than numerical designations. This is how Peer-to-Peer knows to find the prior year-end rather than look back a set number of cycles.

5. GrowthNCUA or GrowthYTDAnn

Another growth function is GrowthNCUA which matches what NCUA uses on its Financial Performance Reports (FPRs). This function projects your full calendar year growth based on the rate of growth so far this year. If it's easier to remember, you can use GrowthYTDAnn (which stands for YTD growth annualized) instead of GrowthNCUA.

GrowthNCUA projects an annual growth rate using your year-to-date growth rate. So, NCUA's FPR growth rates are calculated as such:

- At 1Q: \( \text{GrowthNCUA}(a025b) = ((a025b - a025b:dec)/(a025b:dec)) \times 4/1 \)
- At 2Q: \( \text{GrowthNCUA}(a025b) = ((a025b - a025b:dec)/(a025b:dec)) \times 4/2 \)
- At 3Q: \( \text{GrowthNCUA}(a025b) = ((a025b - a025b:dec)/(a025b:dec)) \times 4/3 \)
- At 4Q: \( \text{GrowthNCUA}(a025b) = ((a025b - a025b:dec)/(a025b:dec)) \times 4/4 \)

6. GrowthQtr

Our final two growth functions look at a credit union's performance over just the last 3-month period. GrowthQtr calculates the percent change over the previous quarter.

\[
\text{GrowthQtr}(a025b) = a025b-a025b:.25
\]

7. GrowthQtrAnn

Our final growth function is GrowthQtrAnn which project a full year's growth based on your most recent quarter's growth rate. GrowthQtrAnn calculates the percent change over the previous quarter and then annualizes that growth at a compounded rate.

\[
\text{GrowthQtrAnn}(a025b) = \frac{a025b-a025b:.25}{a025b:.25} ^ 4
\]
Functions 201: Calculations

This chapter includes functions that are designed to be used with either YTD or continually accruing items, but not both. It is important to use the correct one when building your displays. Since some account codes are reported year-to-date, we’ve created functions that automatically know how to calculate changes correctly.

8. Annual – used only with YTD items, this function estimates a full year’s performance based on current year-to-date performance.

9. FullYear – also used only with YTD items, this function calculates the actual 12-month value for the prior 12 months for any YTD account.

10. Change – calculates the absolute difference between this year and last for any account. Most commonly used with continually accruing accounts.

11. QtrChange – used with continually accruing items, it calculates the difference between quarters.

12. Booked – used with year-to-date accounts, it calculates the dollar or number booked in each quarter.

13. QtrAve – used with balance sheet items to calculate yield, it averages the current cycle with the prior year-end.

14. QtrAveAnn – used with year-to-date items, this function averages this year’s annualized value with the prior full year’s actual value.

8. Annual

The Annual function is used to project full year amounts for items that are reported as year-to-date (YTD) on the call report. So Annual(a031b) projects your full year loan originations, assuming they continue at the same rate as they have so far this year.

Annualizing a YTD line item enables you to make meaningful comparisons across multiple time periods, regardless of which period you are using for your comparison. One common use is in expense and productivity ratios, such as Operating Expense to Average Assets or Average Loan Originations per Employee.

This function is only used on account codes that are reported year-to-date on the Call Report. Here are some examples:

<table>
<thead>
<tr>
<th>Description</th>
<th>Example Code</th>
<th>Where on Call Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year-to-Date Loan Originations</td>
<td>a031b</td>
<td>Page 2</td>
</tr>
<tr>
<td>All Income &amp; Expense Items (Ex. Interest Income)</td>
<td>a115</td>
<td>Page 5</td>
</tr>
<tr>
<td>Loan Charge-offs &amp; Recoveries (Ex. Total C/Os)</td>
<td>a550</td>
<td>Page 9</td>
</tr>
<tr>
<td>Schedule A Specialized Lending (Ex. 1st Mortgages Granted)</td>
<td>a988</td>
<td>Starting on Page 13</td>
</tr>
</tbody>
</table>

Peer annualizes a number by multiplying the value by 4 and then dividing by the current quarter. You do not need to make these calculations – Peer is running all this for you in the background! The account used in this example, a031b, is the account code for year-to-date loan originations.

- If your current cycle is 1Q: \( \text{annual(a031b)} = \frac{(a031b * 4)}{1} \)
- If your current cycle is 2Q: \( \text{annual(a031b)} = \frac{(a031b * 4)}{2} \)
- If your current cycle is 3Q: \( \text{annual(a031b)} = \frac{(a031b * 4)}{3} \)
- If your current cycle is 4Q: \( \text{annual(a031b)} = \frac{(a031b * 4)}{4} \)
9. FullYear

The FullYear function is also used specifically with year-to-date (YTD) line items to give you the actual accumulated value over the previous 12 months. Since YTD line items "reset" at the beginning of each year, it is difficult to see the absolute change in YTD items across different calendar years. The FullYear function allows you to easily see the one year accrual in any YTD item of your choosing.

Peer calculates a YTD item’s full year accumulation by adding the YTD amount to the difference between the previous year-end’s value and the value from one year ago. This is easier to see in a formula!

\[
\text{FullYear}(a031b) = a031b + (a031b:dec - a031b:1)
\]

where a031b is this year’s accrual; a031b:dec is total loan originations as of December 31 the prior year; and a031:1 is the value of a031b at the same cycle last year.

10. Change

Unlike Growth which calculates a percentage rate, the Change function calculates the absolute difference between this year and last for any continually accruing account code. It is the numerator of the Growth calculation.

For example, to calculate how much your loan portfolio has grown in absolute dollars, you would use:

\[
\text{Change}(a025b) = (a025b - a025b:1)
\]

Change displays as a dollar or number, whichever format the original account code is in. Since it’s a volume of change, column charts may be more appropriate.

11. QtrChange

The QtrChange function is used to calculate the absolute difference between this quarter and last quarter for balance sheet items or other accounts that are continually accruing. It will display as a dollar or number, whichever format the original account code is in. Since it’s a volume of change, column charts may be more appropriate.

\[
\text{QtrChange}(a025b) = (a025b - a025b:.25)
\]

12. Booked

Booked is similar to QtrChange but is used on year-to-date items. It is a quick and easy way to calculate how much you have booked in any year-to-date account in any given quarter.

Since credit unions report some items as a year-to-date number, this function parses which cycle you are in to calculate the change from the prior quarter. Unlike continually accruing items, YTD accounts reset each January 1, so the Booked function knows to treat first quarter differently. In other quarters, it subtracts the prior cycle value from the current one.

\[
\text{IF Cycle} = \text{1Q} \quad \text{Booked}(a025b) = (a025b) \quad \text{ELSE} \quad \text{Booked}(a025b) = (a025b - a025b:.25)
\]
13. **QtrAve**
The **QtrAve** function is used only **continually accruing items** and calculates the average balance with the prior year-end. It is primarily used when calculating yields.

\[
\text{QtrAve}(a_{025b}) = \frac{(a_{025b} + a_{025b}:\text{dec})}{2}
\]

Let’s look at how we calculate the yield on the loan portfolio. Simply dividing YTD loan interest income \( (a_{110}) \) by the current quarter’s loan balance (loans) will understate your yield (assuming you are growing the portfolio) or overstate your yield (if you are shrinking it). This is because YTD loan interest income is measured as an accumulated amount over the year, while loan balances are a static value at that specific quarter. In order to get a more accurate measurement of yields, we would instead use \( a_{110}/\text{QtrAve(loans)} \)

14. **QtrAveAnn**
There is another function similar to QtrAve that works with **year-to-date account codes**: **QtrAveAnn**. This function calculates this year’s annualized data and then averages this value with last year’s actual value.

\[
\text{QtrAveAnn}(a_{270}) = \frac{(\text{annual}(a_{270}) + a_{270}:\text{dec})}{2}
\]
Functions 301: Benchmarking

The six functions in this section allow you to **benchmark**, or **rank**, your performance against a set of peer credit unions. They are designed to not only tell you if you are above or below a peer group average, but specify where in the peer group your performance falls in order to set improvement goals. They can be used to build scorecards or on charts to show how your performance ranking has changed over time.

15. **Rank** – calculates where your performance stands in comparison to other credit unions as a numerical rank, where the highest values are assigned the lowest ranks.

16. **ReverseRank** – same as above, but used when higher value is given a higher rank.

17. **Percent** – similar to Rank, this calculates a percentile ranking, where a higher value is assigned a higher rank.

18. **ReversePercent** – Similar to Percent, but higher values receive lower percentiles.

19. **AtPercent** – used to display a metric’s explicit value for a specified percentile.

20. **Peer** – returns the peer group average for any metric.

Note: Before using any of these function, make sure the credit union you want to analyze is set as the Primary. The comparison set should include either individual credit unions or a peer group containing a group of credit unions that you want to benchmark again. If you select multiple peer groups, these function look at each credit union within those groups individually to determine where your performance stands within the whole.

### 15. Rank

The **Rank** function can be used with any account or ratio and returns a numerical rank for your performance within the peer group comparison set. **Rank** is used on values where higher is generally considered “better”.

For example, **Rank(ROA)** will return a numerical value for where the primary credit union’s ROA falls within the comparison set. It returns a number from 1 (the best) to the maximum number of credit unions in your comparison set. Our software ranks discreet values, not necessarily equal to the number of credit unions in your peer group.

You can use this, and the other function in this section, in conjunction with other functions. For example, **Rank(Growth(members))** will return your primary institution’s ranking in Member Growth compared to the rest of the institutions in the comparison set.

### 16. ReverseRank

**ReverseRank** works the same way as Rank but should be used with metrics where **lower is better**. Examples include ratios like Operating Expense/Average Assets, Efficiency or Delinquency.

As an example, **ReverseRank(a041b/loans)** will return a numerical rank for how the primary institution’s delinquency ratio measures against its peer comparisons. The lower the value, the better the ranking, with 1 being the best.
17. Percentile
Similar to Rank, the Percentile function shows how a credit union compares to all the institutions in the comparison set. However, rather than return a numerical rank, this function returns the percentile your credit union falls into for that metric. Percentile assumes that higher is better.

Our software ranks all the values for a performance metric, e.g. ROA, from the credit unions in the peer comparisons you have selected. The percentile ranking is then determined via a simple equation to ensure that the incremental differences between each rank is uniform throughout.

18. ReversePercentile
The ReversePercentile function works the same way as Percentile but should be used when you are looking at metrics were lower is better. Examples include ratios like Operating Expense/Average Assets, Efficiency or Delinquency.

19. AtPercent
The AtPercent function requires the user to specify both a metric (an account code, a ratio, or a formula), as well as a performance level. It returns a metric’s explicit value for the specified percentile.

For example, to show how fast credit unions in your comparison set are growing their membership in the top quartile of performance, you would specify

\[
\text{AtPercent}(75, \text{growth(members)})
\]

Where 75 indicates the top quartile and growth(members) is the formula for member growth.

20. Peer
The Peer function will return the average for your entire comparison set for whatever metric you specify. For example, typing Peer(ROA) on a table or a line chart, will return the average ROA for all peers selected, compared to ROA which would return the value for the primary institution or group.

Peer group averages are computed by totaling the values for each individual account and dividing each sum by the number of credit unions in the peer group. All ratios and other formulas are then computed using these average account values. This confers greater weights to credit unions with higher values in each account.