ARBUTUS
Drive Your Analytics

ARBUTUS ANALYTICS
Popular Tests

Vendor Management

Managing vendors now requires a multitude of tests to ensure that the associated risks are kept to a minimum. The proliferation of complex supply chains has made vendor management a high-risk area for many organizations. In addition to duplicate payments, vendor data quality, conflicts of interest, and watch-list comparisons are key areas for testing.

Vendor Data Quality	Look for missing or incomplete critical data when input validation is weak or non-existent. Use the various Arbutus functions to test for completeness by deploying filters and the Count command to document the results. IsBlank(<field name>) will tell you if a key character field such as the City field has no content. Format(<field name>) will display the underling numeric/alpha content of a field. For example, to test the vendor telephone numbers for appropriate content, use the expected format of your supplier data to identify non-conforming numbers. In the US and Canada, telephone numbers consist of 10 digits. Your filter would look like this: Format(Phone_No) <> "(999) 999-9999"
Conflicts of Interest	Compare vendor and employee telephone numbers, addresses, bank accounts, and tax IDs to identify identical or similar entries. Use the SortNormalize function to standardize the data, then execute a many-to-many Join between the two files.
Watch List Comparisons	Download the GSA SAM file or the OFAC list and use the same process as the conflict of interest test to identify possible matching names or addresses. Click to learn how to download these two data sets. Download Technical documents
Inactive Vendors	A vendor account that has not been active leaves it open to fraud. To identify vendors who have not been active in the past year, download the payments data and the vendor master file. Then, run an unmatched Join between the two files on the vendor number with the vendor master as the primary file. The output will consist of all vendors who have not been active.

Employee Spending

Spending controls aren't always sophisticated enough to prevent unauthorized transactions in a timely manner. Whether it's P-Cards or Travel & Entertainment expenses, the sooner suspicious expenses can be detected, the sooner they can be resolved.

PCards and Travel & Entertainment	**PCard Split Purchases**	The temptation to split a purchase into two transactions to circumvent PCard approval controls poses a perennial challenge. Additionally, two employees may collude by each claiming one of the purchases. This is easily addressed in Analyzer by (1) isolating the below-the-limit transactions by extracting them to a file, (2) extracting these records to a second, identical file, (3) executing a many-to-many Join between the two files with the employee number and vendor ID as the key fields, and (4) filtering for pairs of transactions that add up to an amount over the employee's limit. Additional filters can be invoked to specify pairs of transactions within x days of each other.
PCards and Travel & Entertainment	**Cardholders with Excessive Declined/Disputed Transactions**	Card processors can provide data files containing declined and disputed transactions by employee cardholders. Analyzer's enhanced Summarize command can identify employees with high rates of declines and disputes.
PCards and Travel & Entertainment	**Terminated Employees & Active Cards**	Suspicious transactions frequently occur in the period just before and after employees are terminated. Controls don't always result in timely invalidation of employee PCards, but prompt identification of such transactions is straightforward with Analyzer. A direct connection to Concur and other databases makes it possible to continuously compare employee transactions and termination dates.
PCards and Travel & Entertainment	**Multiple Cards per Employee**	With an automated download from the enterprise application, issued cards can be scanned with the Duplicates command to identify employees with more than one active card.
PCards and Travel & Entertainment	**Stale Claims**	Analyzing the T&E transaction date with the Age command will identify claims that are more than 30 days older than the report date.

GL JE Risk Scoring

Journal entries, particularly manual JEs that are posted close to end-of-period dates, need to be regularly scanned to identify high-risk items. Manual journal entries are high-risk items because they are not part of an automated process. JEs that fall around the end-of-period dates are also of concern.

<table>
<tr><td>2020 Holidays</td><td>Match(Posting_Date,`20200101`,20200120`,`20200217`,`20200625`,`20200703`,`20200907`,`20201012`,`20201111`,`20201126`,`20201225`)</td></tr>
<tr><td>Weekends</td><td>Match(CDOW(Posting_Date,3),"Sun","Sat")</td></tr>
<tr><td>Keywords in Description</td><td>ListFind("Keywords.txt")</td></tr>
<tr><td>Same Account, Same Amount</td><td>Duplicates ON Account Amount OTHER ALL TO …</td></tr>
<tr><td>Seldom Used Accounts</td><td>Classify ON Account TO Newfile_1 OPEN
Extract Account IF COUNT1 <= 3 TO Newfile_2
OPEN <source file>
OPEN Newfile_2 SECONDARY
Join PKEY Account FIELDS ALL SKEY Account WITH PRESORT SECSORT TO JEs_Seldom_Used_AC OPEN</td></tr>
<tr><td>Large Credits to Revenue 5 Days Prior to Period-End</td><td>(Assumption: Revenue Accounts Begin With "4")

Extract RECORD IF Account = "4" AND Amount_CR <> 0 TO JE_Rev_Accts OPEN

Statistics ON ABS(Amount_CR) STD Number 5

Extract RECORD IF ABS(Amount) > AVERAGE1 + (2* STDDEV1) TO Large_CR_Rev_Accts</td></tr>
<tr><td>Large Credits to Income Statement Non-Revenue Accounts</td><td>(Assumption: Target accounts begin with "5","6", or "7")

Extract RECORD IF Match(Account, "5","6", "7") AND Amount_CR <> 0 TO JE_NonRev_Accts OPEN
Statistics ON ABS(Amount_CR) STD Number 5
Extract RECORD IF ABS(Amount) > AVERAGE1 + (2* STDDEV1) TO Large_CR_NonRev_Accts</td></tr>
<tr><td>Round Amounts</td><td>Filter for all amounts with 0 cents:
MOD(Amount,1) = 0

For a more granular, materiality-based analysis, create a conditional computed field with a default value of Blanks(30):

<table><tr><th>Condition</th><th>Value</th></tr><tr><td>MOD(Amount,1000000) = 0</td><td>"Round Millions"</td></tr><tr><td>MOD(Amount,100000) = 0</td><td>"Round Hundred Thousands"</td></tr><tr><td>MOD(Amount,10000) = 0</td><td>"Round Ten Thousands"</td></tr><tr><td>MOD(Amount,1000) = 0</td><td>"Round Thousands"</td></tr><tr><td>MOD(Amount,100) = 0</td><td>"Round Hundreds"</td></tr><tr><td>MOD(Amount,10) = 0</td><td>"Round Tens"</td></tr><tr><td>MOD(Amount,1) = 0</td><td>"Round Singles"</td></tr></table></td></tr>
<tr><td>Prior-Year Entries Posted 5 Days After Year-End</td><td>Between(Posting_Date,`20190101`,`20190106`) AND Period = "2018"</td></tr>
<tr><td>Amounts Just Below Approval Threshold</td><td>(Assumption: Threshold = $5000)</td></tr>
</table>

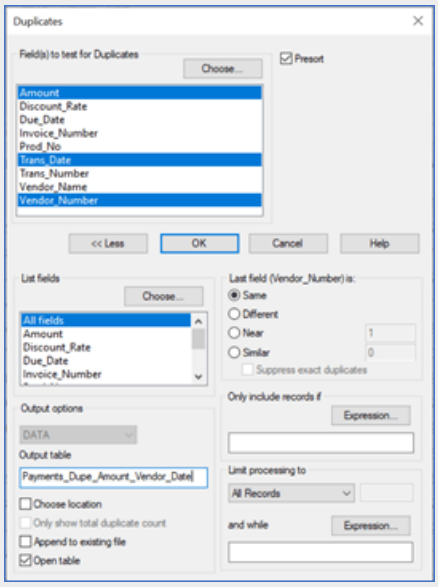
Duplicate Payments

Duplicate payments can be a significant source of financial leakage. Although some systems possess basic built-in duplicate detection, it's not likely that they will detect near duplicates that can come in different configurations. The enhanced Duplicates command in Analyzer has helped users to identify different kinds of fuzzy duplicates.

Same-Same-Same

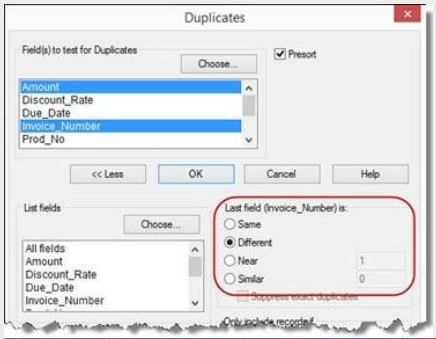
Detecting payments where one or more values are identical, such as same vendor, same amount, same date, is straightforward. In the Duplicates command dialog, select those fields from the "Field(s) to test for Duplicates" list.

You can select additional fields in the "List fields" list to send to the output, such as product number, that may enhance your follow-up analysis. And, finally, the output can be directed to a file in the "Output options" section by selecting "DATA" and naming the output file.

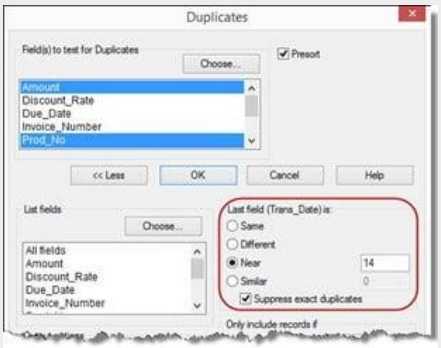
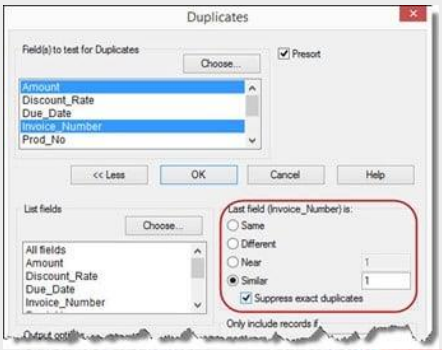


Same-Same-Different

If you're trying to detect same vendor, same amount, different invoice, selecting the "Different" parameter will identify those transactions.



	Vendor_Number	Amount	Invoice_Number	Prod_No	Trans_Date
1	13070	98,423.48	540613	AC102	08/03/2019
2	13070	98,423.48	5406613	AC102	08/03/2019
3	10039	39,759.69	991548	AC097	05/19/2019
4	10039	39,759.69	9961548	AC091	05/19/2019
5	11072	7,119.57	4118706	AC106	01/08/2019
6	11072	7,119.57	4639713	AC071	06/13/2019
7	10673	6,746.47	5651681	AA138	11/23/2019

Same-Same-Near	The "Near" parameter allows greater precision and focus. For example, you could search for same vendor, same month, within $10 for amounts that are very close in value. As well, you could identify same vendor, same product, same amount, date within 14 days to exclude recurring payments. The field being tested for the "Near" quality must be the final field selected in the "Fields to test for Duplicates" list. Note that the results for Near and Similar places the matching records side-by-side. This facilitates more granular testing, such as calculating the actual number of days between the two transaction dates.  |	Vendor Number	Prod No	Invoice Number	Amount	Trans Date	Amount2	Trans Date2 | |---|---|---|---|---|---|---|---| | 1 | 10380 | AC109 | 6936504 | 30.22 | 08/18/2019 | 30.22 | 08/22/2019 | | 2 | 12191 | AC110 | 8194641 | 90.82 | 03/01/2019 | 90.82 | 03/05/2019 | | 3 | 13012 | AC105 | 8227883 | 14.06 | 08/22/2019 | 14.06 | 08/26/2019 |
Same-Same-Similar	It frequently happens that vendors may re-issue the same invoice with a slightly different number, such as re-issuing invoice number "102" as "102a". Or input errors may replace one character so that it is entered as "I02". The "Similar" parameter allows you to test for same vendor, same amount, invoice numbers within 1 character of each other. To compare the invoice numbers, all blanks, leading zeros and punctuation are removed, data is made upper case and similar looking characters (e.g. 1 and I, or 0 and O) are matched. Note how similar the two invoice numbers are for rows 4-6 with substitutions of "I" for "1".  |	Vendor_Number	Amount	Invoice_Number	Invoice_Number2	Trans_Date	Trans_Date2	Prod_No	Prod_No2 | |---|---|---|---|---|---|---|---|---| | 1 | 10039 | 39,759.69 | 991548 | 9961548 | 05/19/2019 | 05/19/2019 | AC097 | AC091 | | 2 | 10119 | 552.14 | 1998068 | 1998086 | 01/23/2019 | 01/23/2019 | AC108 | AC073 | | 3 | 11244 | 61.23 | 96/244 | 96244 | 05/10/2019 | 05/10/2019 | AA168 | AC087 | | 4 | 11255 | 998.10 | 113933 | I-13933 | 11/17/2019 | 12/02/2019 | AC080 | AC080 | | 5 | 11255 | 998.10 | 113933 | I13933 | 11/17/2019 | 11/24/2019 | AC080 | AC080 | | 6 | 11255 | 998.10 | I-13933 | I13933 | 12/02/2019 | 11/24/2019 | AC080 | AC080 | | 7 | 13070 | 98,423.48 | 540613 | 5406613 | 08/03/2019 | 08/03/2019 | AC102 | AC102 |
Suppress Duplicates Parameter	This checkbox will exclude exact matches from the Near and Similar analytics. Exact duplicates are higher-risk, and the presumption is that those would have been already identified. This allows the analyst to focus on a different population without the risk of double-counting.

Counterparty Validation

Compliance and continuity both require frequent testing to detect suspect counterparties. The risks of transacting with counterparties on watch lists is high.
The GSA SAM list contains people and organizations that have committed fraud against the federal government. The OFAC list consists of parties that are suspected of or have committed terrorism. And there are many other watchlists worldwide that should be considered. There are multiple ways in which your counterparties (employees, customers, vendors, and contractors can be compared to such lists.

Normalized Names and Addresses

Use the SortNormalize function to standardize the names and addresses of your counterparties and the population of the watch list. Then, use a many-to-many Join to identify matches. It's possible to integrate a fuzzy search dimension by specifying a filter in the Join for matches that are within one character of each other:

Difference(Vendor_Name, OFAC_Name) <=1

	OFAC Entity Number	OFAC Address Number	OFAC Address	Vendor Number	Vendor Address	OFAC Address SORTNORM	Vendor Address SORTNORM
1	10614	15026	9311 Clancey Avenue	10064	9311 Clancey Avenue	CLANCEY AVE 9311	CLANCEY AVE 9311
2	24427	36901	5599 NW 23rd Ave	10408	5599 NW 23rd Avenue	NW AVE 5599 23RD	NW AVE 5599 23RD
3	24427	36901	5599 NW 23rd Ave	10520	5519 NW 23rd Avenue	NW AVE 5599 23RD	NW AVE 5519 23RD
4	24428	36903	5599 NW 23rd Avenue	10408	5599 NW 23rd Avenue	NW AVE 5599 23RD	NW AVE 5599 23RD
5	24428	36903	5599 NW 23rd Avenue	10520	5519 NW 23rd Avenue	NW AVE 5599 23RD	NW AVE 5519 23RD
6	24428	37118	11301 NW 2 Street	10257	11301 NW 4 Street	ST NW 2 11301	ST NW 4 11301
7	24428	37119	11300 NW 4 Street	10117	11300 NW 4 Street	ST NW 4 11300	ST NW 4 11300
8	24428	37119	11300 NW 4 Street	10257	11301 NW 4 Street	ST NW 4 11300	ST NW 4 11301
9	24428	37120	11350 NW 4 Street	10117	11300 NW 4 Street	ST NW 4 11350	ST NW 4 11300
10	24428	37121	11200 NW 4 Street	10117	11300 NW 4 Street	ST NW 4 11200	ST NW 4 11300
11	24429	36905	5599 NW 23rd Avenue	10408	5599 NW 23rd Avenue	NW AVE 5599 23RD	NW AVE 5599 23RD
12	24429	36905	5599 NW 23rd Avenue	10520	5519 NW 23rd Avenue	NW AVE 5599 23RD	NW AVE 5519 23RD
13	24430	36907	5599 NW 23rd Avenue	10408	5599 NW 23rd Avenue	NW AVE 5599 23RD	NW AVE 5599 23RD
14	24430	36907	5599 NW 23rd Avenue	10520	5519 NW 23rd Avenue	NW AVE 5599 23RD	NW AVE 5519 23RD
15	25578	38700	767 5th Ave, 44th Fl	11256	"767 5th Avenue, 45th Floor"	FL AVE 767 5TH 44TH	FL AVE 767 5TH 45TH

Percent of Word Matches

The normalized names and addresses can be parsed into their individual words with a script. A matching word score can be calculated to display the % of matching words between any two pairs of names or addresses. This level of granularity allows you to quickly order the results with the highest % matches first for review.

	OFAC Address Number	OFAC Address	Vendor Number	Vendor Address	Number of Matched Words	Match %
1	15026	9311 Clancey Avenue	10064	9311 Clancey Avenue	3	100%
2	37119	11300 NW 4 Street	10117	11300 NW 4 Street	3	100%
3	37118	11301 NW 2 Street	10257	11301 NW 4 Street	3	100%
4	36901	5599 NW 23rd Ave	10408	5599 NW 23rd Avenue	4	100%
5	36903	5599 NW 23rd Avenue	10408	5599 NW 23rd Avenue	4	100%
6	36905	5599 NW 23rd Avenue	10408	5599 NW 23rd Avenue	4	100%
7	36907	5599 NW 23rd Avenue	10408	5599 NW 23rd Avenue	4	100%
8	38700	767 5th Ave, 44th Fl	11256	"767 5th Avenue, 45th Floor"	4	80%
9	36901	5599 NW 23rd Ave	10520	5519 NW 23rd Avenue	3	75%
10	36903	5599 NW 23rd Avenue	10520	5519 NW 23rd Avenue	3	75%
11	36905	5599 NW 23rd Avenue	10520	5519 NW 23rd Avenue	3	75%
12	36907	5599 NW 23rd Avenue	10520	5519 NW 23rd Avenue	3	75%
13	37112	480 Park Avenue, Apt. 10B	11951	2 Park Avenue	2	70%
14	37112	480 Park Avenue, Apt. 10B	12302	1 Park Avenue	2	70%
15	5228	525 International Parkway, Suite 509	11657	3 Parkway North	1	60%
16	5228	525 International Parkway, Suite 509	12431	1 Parkway North	1	60%

<< End of File >>

Outliers

Sophisticated statistical tests can rapidly identify outliers in almost any context. Outliers are transactions where the materiality is well beyond historical expectations. Because of their size, errors in processing them can result in misstatements. A very large outlier can also distort what would be considered "normal" for a population.

Population-Level Testing

A common criterion for outliers is that the values are more than two standard deviations above the average for a given population. The Statistics command can quickly generate the mean and the standard deviation:

Field: Total_Cost	Number	Total	Average
Positive	33,220	62,624,618	1,885
Zeros	2,230		
Negative	2,894	-5,462,144	-1,887
Totals	38,344	57,162,474	1,491
Abs Value		68,086,762	
Range		164,474	
Std. Dev		3,155.49	

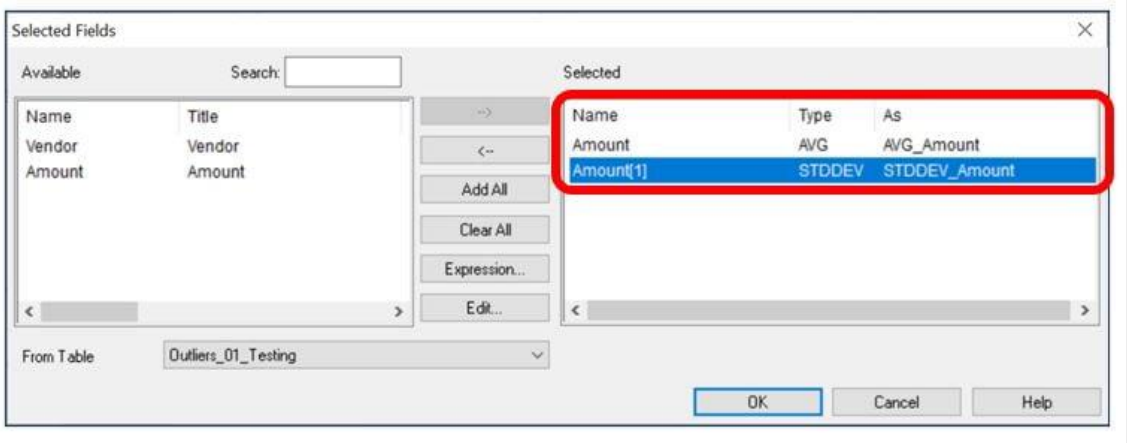
To identify outliers, create a filter:
Total_Cost > 1491 + (2 * 3155.49)

Single-Category-Level Testing

It's also possible to identify outliers at a category level, such as vendors, using the enhanced Summarize command:

1. In the Summarize command, choose the Vendor field as the key field.
2. Open the "Fields to process" dialog
3. Select "Amount" twice
4. Change the Type to AVG and STDDEV

The output file contains the mean and the standard deviation for each vendor:



5. Create a computed field for the 2 SD threshold:
AVG_Amount + (2 * STDDEV_Amount)

	Vendor	AVG_Amount	STDDEV_Amount	Vendor_Threshold
1	10031	792.80	28.6286	850.0572
2	10037	793.32	29.8515	853.0230
3	10039	793.79	29.2476	852.2852
4	10049	793.86	29.4538	852.7676
5	10056	791.32	29.5209	850.3618
6	10061	751.64	58.0991	867.8382
7	10064	693.17	29.3797	751.9294
8	10067	691.82	30.1734	752.1668
9	10069	692.12	29.5366	751.1932
10	10071	750.54	84.5419	919.6238
11	10072	787.59	106.6040	1,000.7980
12	10073	681.52	111.5505	904.6210
13	10074	672.99	175.6771	1,024.3442

6. Open the transaction file.
7. Join to the vendor threshold file and add the threshold field to the output.
8. Filter for Amount > Vendor_Threshold

Amount > Vendor_Threshold

	Vendor	Amount	Vendor_Threshold
13787	10071	945.25	919.6238
13797	10071	943.03	919.6238
13809	10071	944.91	919.6238
13814	10071	936.60	919.6238
19085	10073	961.80	904.6210
19086	10073	947.16	904.6210
19087	10073	1,012.92	904.6210
19088	10073	997.62	904.6210
19093	10073	923.10	904.6210
19094	10073	905.23	904.6210
19095	10073	953.28	904.6210

Multiple-Category-Level Testing

It's also possible to add multiple levels to the category testing. For example, you could test by Vendor-Product combinations to test for pricing consistency.

Follow the same steps as in the previous example, using the Vendor and the Product ID as the key fields.

Technology

The phenomenal growth in scope and complexity of IT requires rigorous testing to ensure that your organization's data and processes are well-protected from the many threats that exist.

Identity Management: Terminated Employees	Use a Join to match terminated employee data from HR to the Active Directory file to identify still-active accounts after employee departures.
SOD	Create a table containing each employee's pairs of duties with a many-to-many Join. Then use a matched Join to identify pairs of employee duties that are prohibited.
Event Log Analysis	Event logs tend to be unstructured. Create a "flat" file using static conditional fields to render the data in a format that can readily be analyzed.
System-Level Settings	Extract system settings at regular points in time, then use the Compare command to identify any changes that may have taken place.
Data Integrity	Data should be rigorously tested prior to analysis to determine whether it is appropriate to use. There are a variety of tests that can be executed to identify potential data issues.
Data Migration	Production data flows regularly to data warehouses, where analysts from different parts of the organization can use it without jeopardizing live data. Continuous monitoring of the migration process can quickly identify migration issues before they pose a serious threat. The Join and Compare commands are essential for this purpose. When new systems are implemented, those commands can also be used to verify that the data has successfully migrated with no integrity issues.
Data Normalization	Key fields can often be in varying formats from one system to another. To make them suitable for comparison, there are a number of functions that can be used in computed fields to normalize the data: • SortNormalize • Normalize • Upper • Lower • Include • Exclude • String • Value • Zoned • AllTrim • Compact • Split • Substring