

WRDS Research Macros©

WRDS Research Macros© are standardized SAS routines frequently used by researchers in Finance and Accounting to perform common empirical procedures that are likely to be a part of any research project. They include tools and utilities applied in routine data cleaning and manipulation to prepare the datasets for statistical analysis (e.g., winsorizing outliers, populating sample from one frequency to another, merging CRSP stock and event data, quarterizing Compustat year-to-date values, calculating dummies and industry classifications, etc.). WRDS Research Macros © also comprise of analytical routines that are used as a part of the actual statistical analysis (e.g. calculating idiosyncratic volatility, returns compounding, doing event studies, Fama and MacBeth regressions, etc) and many other repetitively performed empirical tasks. WRDS Research Macros © product aims to simplify and improve programming efficiency of WRDS users and better address their computing needs. More specifically, research macros, each with its own specific purpose, will help eliminate cluttering, and enhance productivity of business researchers who use SAS for empirical analysis. The Research Macros are transparent and flexible as WRDS users always have an opportunity to view the code and customize it to fit their specific needs. The ultimate objective of this product is to convert major sample programs and research applications into a set of more useful "macros" that can be invoked and used directly by our users within their own SAS programs.

To automatically point to the macros in this library within your SAS programs, users need to add the following line to your autoexec.sas file in your WRDS home directory:

options sasautos=('/wrds/wrdsmacros/', SASAUTOS) MAUTOSOURCE;

If you are using any of WRDS Research Macros in your research, we appreciate if you could, please, provide a citation to WRDS Research Macros. Our suggested citation of WRDS Research Macros is as follows: WRDS Research Macros, 2010, Wharton Research Data Services, The Wharton School, University of Pennsylvania, <https://wrds.wharton.upenn.edu>.

Below is the list of the currently available macros. If you would like to contribute to WRDS with your macro, please feel free to send it to us (wrds-support@wharton.upenn.edu) and we will happy to add your program to WRDS Research Macros with your name as an attribution of your efforts.

List of WRDS Research Macros©

- Utilities
 1. [NWORDS](#): Counts number of words within a text string. Useful input for other macros.
 2. [CSV](#): Exports a SAS file, within Unix, into a Excel spreadsheet or a csv file.
- Analytical Macros
 1. [WINSORIZE](#): Winsorizes or Trims your sample. Uses SAS Proc Univariate procedure.
 2. [EVTSTUDY](#): Calculates Cumulative Abnormal Returns within event window.
 3. [POPULATE](#): Converts data to monthly frequency. Annual to Monthly or Quarterly to Monthly.
 4. [NEUT](#): Neutralizes (orthogonalizes) variables by a list of numerical or categorical common factors.
 5. [FFI](#): Creates Fama & French 5, 10, 12, 17, 30, 38, 48 or 49 Industry Classification Variables.
 6. [FM](#): Calculates Fama-MacBeth coefficients with Newey-West adjusted standard errors.
 7. [MAKE DUMMIES](#): If a variable takes 3 values ("ABC","DEF","HIJ"), 3 dummy variables are created.
 8. [TRADE DATE WINDOWS](#): Creates a trading-date window of user-specified size.
- Database Specific Macros
 1. [CRSPMERGE](#): Merges CRSP Stock Time-Series with CRSP Event Data.
 2. [BETA](#): Compute Market-Model Beta for all CRSP securities on a rolling daily or monthly basis.
 3. [IDVOL](#): Calculates idiosyncratic volatility using time-series regressions for various risk models.
 4. [QUARTERIZE](#): Quarterizes some or all Compustat Quarterly YTM Cash Flow Items.
 5. [COMPOUND](#): Calculates the continuously compound returns.
 6. [VW AVGPRICE](#): Using TAQ data generates average trade prices for user-specified time intervals.
 7. [INDCLASS](#): Creates up to 4 industry classifications including Fama-French classification.
 8. [INDUSTRY RATIOS](#): Computes a broad range of financial ratios aggregated at the industry level.
- Linking Macros
 1. [ICLINK](#): Creates a link table between IBES TICKER and CRSP PERMNO.
 2. [CCM](#): Uses CRSP-Compustat Merged Table to Add Permno to Compustat Data.
 3. [OCLINK](#): Creates a link table between OptionMetrics SECID and CRSP PERMNO.
 4. [TCLINK](#): Create a link table between TAQ SYMBOL and CRSP PERMNO.
- WRDS SEC Platform Macros
 1. [LINEPARSE](#): Parses and extracts a paragraph. Extracts number of lines around a given line number.
 2. [TEXTPARSE](#): Parses any hits that match a given string. Extracts a given number of preceding characters.
 3. [LINEPARAPARSE](#): Parses and extracts a paragraph. Extracts number of lines around a given line number.
 4. [PARAPARSE](#): Parses and extracts a paragraph with pre-specified number of lines around a string.

UTILITIES:

1. NWORDS

Summary	Counts number of words within a text string.
---------	--

Invocation Line	%NWORDS(INVAR);
Example	%let numwords=%NWORDS("TO BE OR NOT TO BE"); %put &numwords;
SAS Code	View nwords.sas code
Usage Notes	Very useful for Loops.

2. CSV

Summary	The CSV macro exports a SAS dataset, within Unix, into a formatted Excel spreadsheet or a comma delimited csv file. The user specifies the input dataset, whether in a permanent or temporary library, and has the option to specify the name of the output file, or the output file type (excel or csv).
Invocation Line	%CSV(INSET=,OUTSET=,EXCEL=);
Example	%CSV(inset=crsp1);
SAS Code	View csv.sas code
Usage Notes	The default extract is a csv file (EXCEL=0). If a formatted Excel spreadsheet is desired, use EXCEL=1 option.

[Back to the List of WRDS Research Macros](#)

ANALYTICAL MACROS:

3. WINSORIZE

Summary	Winsorizes or Trims your sample. This macro uses SAS Proc Univariate procedure to calculate breaking points by for each date provided in the input dataset. Then, breakpoints are used for trimming or winsorizing depending on the user's input. Users need to provide a SAS dataset and specify the date or the sort variable within which breakpoints are determined and the variables of interest are winsorized.
Invocation Line	%WINSORIZE (INSET=,OUTSET=,SORTVAR=,VAR=,PERC1=,TRIM=);
Example	%WINSORIZE (INSET=crsp1,OUTSET=crsp2,SORTVAR=date,VAR=ret,PERC1=1,TRIM=0);
SAS Code	View winsorize.sas code
Usage Notes	TRIM=0 performs winsorization, while TRIM=1 trims outliers and converts them into missing values

4. EVTSTUDY

Summary	The Event Study Macro calculates Cumulative Abnormal Returns and various t-statistics (Patell's t, Sign t, etc) for an input dataset (Inset) containing (permno, evtdate) combinations. Outputs mean cumulative abnormal return within the event window (set by the variables start and end) and different statistics into the output dataset (Outset). Abnormal returns are computed using either Market, Fama-French or Carhart 4-factor models during the estimation period. The program draws on some of the ideas outlined in the event study macro in Chapter 6 of Boehmer, Broussard and Kallunki (2002)
Invocation Line	%EVTSTUDY (INSET=, OUTSET=, OUTSTATS=, ID=, DATA=, EVTDATE=, ESTPER=, START=, END=, GAP=, GROUP=, MODEL=);
Example	%EVTSTUDY (INSET=DATA1, OUTSET=DATA2, OUTSTATS=STATS, ID=PERMNO, DATA=CRSP, EVTDATE=DATE, ESTPER=110, START=-1, END=+1, GAP=5, GROUP=SUBGROUP, MODEL=FF); Performs an event study for the input dataset (INSET) with PERMNO as identifier (ID), date variable DATE (EVTDATE) with estimation window length of 110 trading days (ESTPER). The event window is set to be (-1,+1) using START and END parameters. The pre-event window's length is set to 5 trading days (GAP). The group variable is identified by SUBGROUP field in INSET table. The benchmark risk model is specified as Fama-French 3-factor model. The cross-sectional output is DATA2 (OUTSET parameter), test statistics are in STATS (OUTSTATS parameter)
SAS Code	View evtstudy.sas code
Usage Notes	A note on the calculated stats: <ol style="list-style-type: none"> 1. Patell (1976) statistic assumes cross-sectional independence in abnormal returns and no event-induced change in variance of the event window returns 2. Cross-sectional test still assumes the cross-sectional independence, but accounts for event-induced variance changes 3. Boehmer's statistic accounts for both event-induced variance changes as well as potential cross-sectional dependence of CARs 4. The Sign statistic relies on the assumption of no cross-correlation of abnormal returns 5. To find out the results of the hypothesis test for comparing groups, find the row of output labeled 'Model' and look at the column labeled 'F-value' for the Fisher statistic and Pr>F for the associated p-value. HOVTEST tests for whether variances of two groups are the same

5. POPULATE

Summary	Populates a dataset into monthly frequency. Annual to Monthly or Quarterly to Monthly. It also generates a monthly date variable
Invocation Line	%POPULATE (INSET=, OUTSET=, DATEVAR=, IDVAR=, MONTHVAR=, FORWARD_MAX=);
Example	%POPULATE (INSET=FUNDA1,OUTSET=FUNDA2,DATEVAR=DATADATE,IDVAR=GVKEY FYR);
SAS Code	View populate.sas code
Usage Notes	Starts from any frequency (annual, quarterly, or sporadic frequency) and populates the data into monthly snapshots, by generating a monthly date, MONTHVAR (default is MDATE), variable. FORWARD_MAX is the periodicity of the original dataset, in months, e.g. Quarterly (=3), Semi-Annual(=6) or Annual(=12), and it also represents the maximum carry-forward intervals. So, for cases when the gap between dates is more than forward_max, POPULATE macro will not fill all gaps. Also, if the gap between dates is less than forward_max, POPULATE will not generate overlapping dates, and instead will stop populating forward a date, as soon as a new date starts, as long as it is within the forward_max distance in months.

6. NEUT

Summary	Neutralizes (orthogonalizes) a set of variables by a list of numerical or categorical common factors
Invocation Line	%NEUT (INSET=,OUTSET=,IDVAR=,DATEVAR=,NEUTVAR=,SUFFIX=,NEUTFAC=,NEUTCFCAC=,WEIGHT=);
Example	%NEUT(INSET=d1,OUTSET=d2,IDVAR=permno,DATEVAR=date,NEUTVAR= ret retx,SUFFIX=adj,NEUTFAC=rmrf smb hml umd, NEUTCFCAC=FFI48, weight=size_1_log);
SAS Code	View neut.sas code
Usage Notes	Neutralization is done every period by running cross-sectional regressions of the dependent variables (NEUTVAR) on the common factors (NEUTFAC and NEUTCFCAC). The neutralized variables are simply the residual from the cross-sectional regressions. A PROC GLM is used to accommodate the categorical independent variables. If no suffix is specified, the new neutralized variables will overwrite the old raw variables.

7. FFI

Summary	Creates Fama & French 5, 10, 12, 17, 30, 38, 48 or 49 Industry Classification Variables with Descriptions
Invocation Line	%FFI48(SIC_CODE);
Example	%FFI48(SIC);
SAS Code	View ffi48.sas code
Usage Notes	F&F Industry Macros are invoked within a data step, and as long as an SIC code is available. The Macros produce 2 variables: - FFI48: F&F Industry Number using X industry classification scheme, where X can be either one of: 5, 10, 12, 17, 30, 38, 48 or 49 - FFI48_DESC: F&F Industry Short Descriptions.

8. FM

Summary	Performs Fama-MacBeth Regressions. Calculates FM coefficients with Newey-West adjusted standard errors
Invocation Line	FM (inset=,outset=,datevar=,depvar=, indvars=,lag=, help=);
Example	FM(inset=data1, outset=data2, datevar=date, depvar=avrret, indvars=size bm mom, lag=1) runs cross-sectional FM regressions avrret=const size bm mom "by date" within data1 and outputs average FM coefficients with standard errors adjusted for serial correlation (lag=1, first-order), t-stats, p-values as well as degrees of freedom for each regressor (i.e., size, bm and mom) into data2 dataset
SAS Code	View fm.sas code
Usage Notes	Output dataset contains the parameter name (one of the independent variables), estimate value (FM coefficient),corrected and uncorrected standard error (stderr and stderr_uncorr), t-value (tvalue and tvalue_uncorr),pvalue (probt and probt_uncorr) and the degree of freedom for each independent variable in FM regression

9. MAKE_DUMMIES

Summary	Makes a copy of a dataset, with a set of dummy variables added, where each new dummy variable corresponds to a distinct value of a user-specified variable. For instance if variable TYPE takes four values ("FIXED INC","GROWTH","BALANCED", and "TAXFREE"), four dummy variables are created: TYPE_DUM1, TYPE_DUM2, TYPE_DUM3, and TYPE_DUM4. It also will make variable the "dummy pointer" variable TYPE_DNUM, which specifies which dummy is set for each record produced.
Invocation Line	%make_dummies(indsn=,var=,outdsn=,help=,cleanup=);

Example	%make_dummies(indsn=company_annuals,var=company_name,outdsn=read_for_regression)
SAS Code	View make_dummies.sas code
Usage Notes	Using %make_dummies(help=yes) will print a complete explanation of the macro on your SAS log. In addition to the dummy variables, setting "cleanup=no" will preserve a "lookup table" dataset in the temporary sas library WORK.

10. TRADE_DATE_WINDOWS

Summary	Makes a dataset with For any dataset containing a date variable, make a copy with a trading-date window of user-specified size. The resulting dataset will have three new variables: BEGDATE, ENDDATE, and WINDOW_SIZE. These will all be calculated using a trading-day calendar only – non-trading dates are ignored.
Invocation Line	%trade_date_windows(freq=,size=,minsize=,outdsn=,help=);
Example	%trade_date_windows(freq=d,size=100,minsize=80,outdsn=windows_100days)
SAS Code	View trade_date_windows.sas code
Usage Notes	Using %trade_date_windows(help=yes) will print a complete explanation of the macro on your SAS log. Note that there is no user-designated input dataset. Instead the result of this program can be used as a "lookup table" with a user dataset to produce date windows. Currently this macro produces no BEGDATE prior to 12/31/2005.

DATABASE SPECIFIC MACROS:

11. CRSPMERGE

Summary	Merges CRSP Stock Time-Series with CRSP Event Data: Do you want to keep CRSP common stocks only for your research? or keep securities Listed in NYSE only? Given the structure of CRSP data, these tasks are not trivial. The macro generates monthly datasets by default, but users can use the s=d option to extract daily CRSP datasets instead.
Invocation Line	%CRSPMERGE(S=, START=, END=, SFVARS=, SEVARS=, FILTERS=, OUTSET=);
Example	%let msevars=ticker ncusip shrcd exchcd; %let msfvars=prc ret retx shrout cfacpr cfacshr; %let filter = exchcd in (1,2,3); %let date1=01JAN1982 %let date2=31DEC2009; %crspmerge(start=&date1,end=&date2,sfvars=&msfvars,sevars=&msevars,filters=&filter);
SAS Code	View crspmerge.sas code
Usage Notes	Make sure CRSP library is correctly assigned the corresponding daily and/or monthly CRSP dataset folder. This CRSPMERGE macro calls NWORDS macro that is located in WRDS server. See CRSP dataset list for the contents of the CRSP daily and monthly events (DSE or MSE) and stocks (DSF or MSF) datasets. Filters can be used directed to subset CRSP data using HISTORICAL characteristics, such as SIC codes, stock codes, or exchange codes. e.g.: - to keep NYSE and AMEX securities, use: filters = exchcd in (1,2) ; - to keep common stocks, use: filters = shrcd in (10,11); - to keep companies in Crude Petroleum and Natural Gas industry, use:filters=siccd=1311

12. BETA

Summary	Compute Market-Model Beta for all CRSP securities on a rolling daily or monthly basis.
Invocation Line	%BETA(S=, START=, END=, WINDOW=, MINWIN=);
Example	%BETA(S=d,START=01JAN1990,END=30JUN2001,WINDOW=250,MINWIN=60); or %BETA(S=m,START=01JAN1990,END=30JUN2001,WINDOW=60,MINWIN=12);
SAS Code	View beta.sas code
Usage Notes	For every month or day, the macro computes the beta of the stock using the trailing returns within the pre-specified window. The efficiency of the code arises from the superior performance of PROC EXPAND in computing rolling statistics. The code can be easily modified to efficiently compute the four loadings for each of Fama and French factors.

13. IDVOL

Summary	Calculates idiosyncratic volatility using time-series monthly/daily regressions for various risk models
Invocation Line	%IDVOL (inset=, outset=, datevar=, retvar=, freq=, window=, step=, min=, model=);

Example	%IDVOL (inset=RETS, outset=RESuITS, datevar=DATE, retvar=RET, freq=D, window=22, step=1, min=15, model=FF) Calculates idiosyncratic volatility (standard deviation of residuals) at the individual stock level using estimation windows of 22 trading days (WINDOW) rolled forward by 1 day (STEP) with at least 15 trading days of non-missing returns (MIN) based on Fama-French (FF)model. The input dataset RETS should contain PERMNO identifier, date and return info. Output dataset RESuITS will contain estimates of idiosyncratic risk (standard deviation of residuals (IDRISK_STD), number of non-missing returns in the estimation window (NUSED) as well as beginning (inclusive) and ending (exclusive) date of the estimation window window (START_DATE and END_DATE, respectively)
SAS Code	View idvol.sas code
Usage Notes	

14. QUARTERIZE

Summary	Some Compustat variables, mainly cash flow statement items, are provided in a YTD (year-to-date) format. This macro quarterizes some or all Compustat Quarterly YTM Cash Flow Items.
Invocation Line	%QUARTERIZE(INSET=, OUTSET=, IDVAR=, FYEAR=, FQTR=, VARS=);
Example	%QUARTERIZE;
SAS Code	View quarterize.sas code
Usage Notes	The input dataset should have a fiscal year and quarter variables (FYEAR and FQTR), before invoking the macro. GVKEY FYR Combination is necessary for unique identification of records. Quarterize only Cash Flow items in Compustat and income statement items in Compustat are quarterly numbers. This macro defaults to all YTM Cash Flow variables in FUNDDQ dataset.

15. COMPOUND

Summary	The Compound macro calculates the continuously compound returns with an option to account for delisting returns. It uses returns in input file (INSET) provided at a given frequency (INFREQ) over the time intervals specified by OUTFREQ (annual, semi-annual, quarterly, monthly or weekly). It also provides maximum and minimum returns, number of total and missing observations into the output dataset OUTSET.
Invocation Line	%COMPOUND (INSET=, OUTSET=, INFREQ=, OUTFREQ=, DATEVAR=, DELIST=);
Example	%COMPOUND (INSET=data1, OUTSET=data2, INFREQ=d, OUTFREQ=q, DATEVAR=date, DELIST=) will compound returns quarterly from daily returns for each security in Data1 and will output them into Data2 dataset
SAS Code	View compound.sas code
Usage Notes	Users should have access to CRSP daily and monthly stock files

16. VW_AVGPRICE

Summary	Volume Weighted Average Trade Price Macro Generates a dataset of volume-weighted average trade prices for user-specified time intervals (e.g. minute-by-minute, 30 seconds, 5 minutes, etc.).
Invocation Line	%vw_avgprice(indsn=, outdsn=, begdate=, enddate=, beghms=, endhms=, inthms=, symlist=, symdsn=, help=);
Example	Creates dataset Data2 containing 4 industry classifications including Fama-French 48 industries (FFIND) using annual data from 1985 to 2008 inclusive based on the list of unique gvkeys specified in data1.
SAS Code	View vw_avgprice.sas code
Usage Notes	Users should have access to CRSP daily and monthly stock files

17. INDCLASS

Summary	Constructs 4 different industry classifications based on historical SIC, NAICS, GICS, and Fama-French industry classifications using Compustat data only and populates it into datasets of various frequencies (day, month, year).
Invocation Line	%INDCLASS (INSET=, OUTSET=, FFIND=, BEGDATE=, ENDDATE=, FREQ=);
Example	%INDCLASS (INSET=data1, OUSET=data2, FFIND=48, BEGDATE=01jan1985, ENDDATE=31dec2008, FREQ=year)Creates dataset Data2 containing 4 industry classifications including Fama-French 48 industries (FFIND) using annual data from 1985 to 2008 inclusive.
SAS Code	View indclass.sas code
Usage Notes	

18. INDUSTRY_RATIOS

Summary	Computes a broad range of financial ratios aggregated at the industry level using Fama-French industry classification
Invocation Line	%INDRATIOS (BEG_YR=, END_YR=, NIND=, AVR=, FREQ=, OUTSET_IND=, OUTSET_FIRM=);
Example	%INDRATIOS (BEG_YR=1980, END_YR=2009, NIND=12, AVR=Median, FREQ=A, OUTSET_IND=Ind_ratios, OUTSET_FIRM=firm_ratios) uses annually (freq=A) updated Compustat vintage to produce two SAS datasets, one called "Ind_ratios" containing a time-series of median (AVR) financial ratios for 12 industries (NIND) according to Fama-French classification, and another one called "firm_ratios" containing calculated ratios at the firm level between years 1980 (BEG_YR) and 2009 (END_YR). The output datasets will be saved to the user's home directory on WRDS
SAS Code	View indratios.sas code
Usage Notes	<p>The macro calculates the following ratios (only for companies incorporated in the US and reporting in USD) using annual data:</p> <p>Valuation Ratios :EPS, P/E Ratio, Price/Book, Price/Sales, Dividend Yield, Earnings Yield, Dividend Payout;</p> <p>Leverage Ratios : Debt/Market Equity Ratio, Debt/Book Equity Ratio, Financial Debt/Market Equity Ratio, Interest Coverage, Debt/Net Working Capital Ratio, Debt/Net Working Capital Ratio</p> <p>Profitability Ratios: Gross Profit Margin, Net Profit Margin, Pre-Tax Profit Margin, Operating Profit Margin, Return on Book Equity, Return on Sales, Return on Assets, Cash Flow Margin;</p> <p>Activity Ratios: Inventory Turnover, Receivables Turnover, Total Asset Turnover, Days of Inventories, Outstanding, Days of Payables Outstanding, Days of Receivables Outstanding;</p> <p>Liquidity Ratios:Current Ratio, Quick Ratio, Cash Ratio, Inventory to Net Working Capital;</p>

[Back to the List of WRDS Research Macros](#)

LINKING MACROS:

19. ICLINK

Summary	Creates a link table between IBES TICKER and CRSP PERMNO and scores links from 0 (best link) to 6 .
Invocation Line	%ICLINK (IBESID=IBES.ID,CRSPID=CRSP.STOCKNAMES,OUTSET=WORK.ICLINK);
Example	%ICLINK;
SAS Code	View iclink.sas code
Usage Notes	

20. CCM

Summary	Uses CRSP-Compustat Merged Table to Add Permno to Compustat Data, has options to remove secondary matches, and is flexible in allowing for a pre-specified overlap in the date conditions.
Invocation Line	%CCM (INSET=,DATEVAR=DATADATE,OUTSET=,LINKTYPE=LulC,REMDUPS=1,OVERLAP=0);
Example	%CCM(INSET=comp1, outset=comp2);
SAS Code	View ccm.sas code
Usage Notes	

21. OCLINK

Summary	Creates a link table between OptionMetrics SECID and CRSP PERMNO, using common CUSIP, and TICKER identifiers and scoring the links using company names
Invocation Line	%OCLINK (OPTIONMID=OPTIONM.SECNMD,CRSPID=CRSP.MSENAMES,OUTSET=WORK.OCLINK);
Example	%OCLINK;
SAS Code	View oclink.sas code
Usage Notes	

22. TCLINK

Summary	Create a link table between TAQ SYMBOL and CRSP PERMNO, using TAQ SYMBOL and Vintage Date combination as primary and unique identifier for TAQ securities, in the absence of primary key in TAQ data
Invocation Line	%TCLINK (BEGDATE=199301,ENDDATE=201012,OUTSET=WORK.TCLINK);
Example	%TCLINK;
SAS Code	View tclink.sas code
Usage Notes	

[Back to the List of WRDS Research Macros](#)

WRDS SEC PLATFORM MACROS:

23. LINEPARSE

Summary	WRDS-SEC Line-by-Line Parser for a given text or regular expression preserving tabular format preserving tabular format.
Invocation Line	%LINEPARSE(INSET=, OUTSET=, FNAME_Full=, TSTR=);
Example	<pre>data wrds_sec1; set sec.wforms; format FNAME2 \$100. ; where form like '%10%K%' and CIK in ('000078003') and fsize>0; FNAME2 = cats("/wrds/sec/archives/",FNAME); drop FINDEXDATE LINDEXDATE SOURCE FNAME ISIZE; run; %let string=Celebrex Lawsuit; %lineparse(inset=wrds_sec1,outset=wrds_sec2,fname_full=fname2,tstr=&string);</pre>
SAS Code	View lineparse.sas code
Usage Notes	HTML tags are cleaned before matching the search string. Search is case insensitive. Regular expressions should be compatible with the PRXMATCH function in SAS.

24. TEXTPARSE

Summary	Parses WRDS-SEC filings for any hits that match a given string or regular expression, and extracts in addition to the match line, a pre-specified number of preceding characters.
Invocation Line	%TEXTPARSE(INSET=, OUTSET=, FNAME_Full=, TSTR=, LN=);
Example	<pre>data wrds_sec1; set sec.wforms; format FNAME2 \$100. ; where form like '%10%K%' and CIK in ('0000789019') and fdate >= '01JAN2000'd and fsize>0; FNAME2 = cats("/wrds/sec/archives/",FNAME); drop FINDEXDATE LINDEXDATE SOURCE FNAME ISIZE; run; %let string=Windows 7 Windows XP; %let paragraph_length=300; %TEXTPARSE(inset=wrds_sec1,outset=wrds_sec2,fname_full=fname2,tstr=&string, ln=%paragraph_length);</pre>
SAS Code	View textparse.sas code
Usage Notes	HTML tags are cleaned before matching the search string. Search is case insensitive. Regular expressions should be compatible with the PRXMATCH function in SAS.

25. LINEPARAPARSE

Summary	Parses WRDS-SEC filings and extracts a paragraph with pre-specified number of text lines around a given line number.
Invocation Line	%LINEPARAPARSE(INSET=,OUTSET=,FNAME_Full=,LINE=,NLIN=);
Example	%LINEPARAParse(INSET=wrds_sec1,OUTSET=wrds_sec2,FNAME_Full=fname3,LINE=Line_num,NLINE=10);
SAS Code	View lineparaparse.sas code
Usage Notes	The macro is usually invoked within the PARAPARSE macro unless the user has the pre-specified lines around which the parsing will be done.

26. PARAPARSE

Summary	Parses WRDS-SEC filings and extracts a paragraph with pre-specified number of text lines around a given match string.
---------	---

Invocation Line	<code>%PARAPARSE(INSET=, OUTSET=, FNAME_FulL=, TSTR=, NLINE=);</code>
Example	<pre>data wrds_sec1; set sec.wforms; format FNAME2 \$100. ; where form like '%10%K%' and CIK in ('0000078003') and fdate >= '01JAN1998'd and fsize>0; FNAME2 = cats("/wrds/sec/archives/",FNAME); drop FINDEXDATE LINDEXDATE SOURCE FNAME ISIZE; run; %let string=Lipitor; %PARAPARSE(INSET=wrds_sec1,OUTSET=wrds_sec2,FNAME_FulL=fname2,TSTR=&string,NLINE=20);</pre>
SAS Code	View paraparse.sas code
Usage Notes	This macro parses for a text string first, and then invokes the LINEPARAPARSE macro to get text around hits. HTML tags are cleaned before matching the search string. Search is case insensitive. Regular expressions should be compatible with the PRXMATCH function in SAS. Do not use for more than 10,000 filings as it is very slow compared with other parsing macros.

[Back to the List of WRDS Research Macros](#)