SAS® Contextual Analysis In-Database Scoring 14.2 for Hadoop: User's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® Contextual Analysis In-Database Scoring 14.2 for Hadoop: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Contextual Analysis In-Database Scoring 14.2 for Hadoop: User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

December 2016

14.2-P1:ctxtindbug

Contents

SAS Contextual Analysis In-Database Scoring 14.2 for Hadoop: User's Guide

Audience

This book is for users who want to work with SAS Contextual Analysis scoring models in Hadoop. It is assumed that you know how to use SAS Contextual Analysis.

Requirements

You must have the following products licensed:

- SAS Contextual Analysis
- SAS In-Database Code Accelerator for Hadoop

Before you can use the In-Database Scoring for Hadoop, you must follow the steps in Chapters 2 and 3 of *SAS Contextual Analysis In-Database Scoring for Hadoop: Administrator's Guide*.

Chapter 1

Introduction

About SAS Contextual Analysis In-Database Scoring for Hadoop

This product enables you to run scoring models generated in SAS Contextual Analysis inside a Hadoop environment. Sample code that you would run is provided here. Please see the appropriate sections of this book for your needs.

Item to Score	See This Section
Concepts	"Using the Concept Model" on page 3
Categories	"Using the Category Model" on page 7
Sentiment	"Using the Sentiment Model" on page 11

Chapter 2

Using the Text Analytics Models

Accessing the Models

Text analytics models include score code, which is generated within SAS Contextual Analysis. To generate the score code, see the section "Viewing and Downloading Code" in Chapter 2 of *SAS Contextual Analysis: User's Guide*.

- To locate the binary files for concepts and categories, see "Locating the Rules Files (LI and MCO)" in *SAS Contextual Analysis: Administrator's Guide*.
- To locate language-specific SAM (binary) files that are used in SAS Contextual Analysis, see the file **\tktg\sasmisc*nn*-base.sam** (Windows) or **misc/tktg/*nn*-base.sam** (UNIX) under your SAS installation directory. Note that the first two characters (*nn*) of the .sam filename denote the licensed language. For example, in **en-base.sam** , the **en** denotes English.
- To locate sentiment (SAM) files that you created using SAS Sentiment Analysis Studio, see *SAS Sentiment Analysis Studio 12.2: User's Guide*.

Using the Concept Model

The following code can be executed in SAS Contextual Analysis In-Database Scoring for Hadoop for concept extraction.

```
/*****************************************************************************
 * Concept score code for Hadoop
 *
 * Copyright (C), 2016
 * SAS Institute Inc., Cary, N.C. 27513, U.S.A.  All rights reserved.
 *****************************************************************************/
```

```
/*****************************************************************************
 * Set SAS environment variables that specify the location of the Hadoop Java
 * client API and configuration files. They are used to access Hadoop services.
 * The Java client API is provided by the Hadoop vendor in the form of JAR files.
 *
 * NOTE: The SAS Contextual Analysis In-Database Scoring for Hadoop
 *       Administrator's Guide describes how to collect Hadoop Java client API jars
 *       and configuration files.
 *****************************************************************************/
options set=SAS_HADOOP_JAR_PATH="C:\path\to\Hadoop\jars";
options set=SAS_HADOOP_CONFIG_PATH="C:\path\to\Hadoop\conf";

/*****************************************************************************
 * Execute a LIBNAME statement to assign a library reference to associate with
 * a Hadoop HDFS or HIVE server.
 *
 * Please contact your IT administrator for the HIVE server name.
 *****************************************************************************/
libname gridlib hadoop server="hivenode.com" user=xxxxx password=xxxxxxx;

/*****************************************************************************
 * Execute a LIBNAME statement to assign a library reference to
 * the location of the local SAS data set to be scored.
 *
 * If all the data sets used for scoring are already in HDFS, the libary might
 * not be needed.
 *****************************************************************************/
libname home "C:\path\to\local\dataset";

/*****************************************************************************
 * Copy the data set to HDFS. Unique observation IDs are required and are
 * created as the field _document_id.
 *
 * If all the data sets used for scoring are already in HDFS, this step might
 * not be needed.
 *****************************************************************************/
data gridlib.input_dataset;
    set home.input_dataset;
    _document_id = _N_;
run;

/*****************************************************************************
 * Set input/output macro variables.
 *
 * input_ds:         Name of input data set in Hadoop, including the HDFS libref
 * document_id:      Name of the unique document ID column in the input data set
 * document_column:  Name of the column to process in the input data set
 * liti_binary_path: Path to the LITI binary
 * output_ds:        Name of the output data set, including the HDFS libref
 *
 * NOTE: The SAS Contextual Analysis In-Database Scoring for Hadoop
 *       Administrator's Guide describes how to deploy Text Analytics models
 *       into clusters.
 *****************************************************************************/
%let input_ds = gridlib.input_dataset;
%let document_id = _document_id;
```

```
%let document_column = text_to_process;
%let liti_binary_path = '/path/to/liti/binary.li';
%let output_ds = gridlib.concept_out;

/*****************************************************************************
 * Delete the output before starting (Optional)
 *****************************************************************************/
proc delete data=&output_ds; run;

/*****************************************************************************
 * Scores concepts in Hadoop
 *****************************************************************************/
proc ds2 ds2accel=yes xcode=warning;

    /* These packages are part of the Text Analytics add-on and are installed */
    /* in the EP                                                               */
    require package tkcat; run;
    require package tktxtanio; run;

    /* The output of the thread program is the input of the data program      */
    THREAD workerth / overwrite=YES;

        dcl package tkcat cat();
        dcl package tktxtanio txtanio();

        dcl binary(8)     _apply_settings;
        dcl binary(8)     _document;
        dcl binary(8)     _liti_binary;
        dcl binary(8)     _trans;

        dcl double        _status;
        dcl double        _num_matches;
        dcl double        _i;
        dcl double        _document_id;
        dcl varchar(1024) _name;
        dcl varchar(1024) _full_path;
        dcl double        _start_offset;
        dcl double        _end_offset;
        dcl varchar(1024) _term;
        dcl varchar(1024) _canonical_form;

        retain            _apply_settings;
        retain            _liti_binary;
        retain            _trans;

        /*********************************************************************
         * Initialization step. Only runs once when starting.
         *********************************************************************/
        method init();
            _apply_settings = cat.new_apply_settings();
            _liti_binary = txtanio.new_on_content_server(&liti_binary_path);

            _status = cat.set_apply_model(_apply_settings, _liti_binary);
            if _status NE 0 then put 'ERROR: set_apply_model fails';

            /* Match types are 0=ALL, 1=LONGEST or 2=BEST */
```

```
            _status = cat.set_match_type(_apply_settings, 0);
            if _status NE 0 then put 'ERROR: set_match_type fails';

            _status = cat.initialize_concepts(_apply_settings);
            if _status NE 0 then put 'ERROR: initialize_concepts fails';

            _trans = cat.new_transaction();
        end;

/*************************************************************************
 * Run step. The method runs per row of input.
 *************************************************************************/
 method run();
     set &input_ds(keep=(&document_column &document_id));

     /* Only process if document observation is not empty*/
     if &document_column NE ' ' then do;

         /* Initialize the document with the column data */
         _document = txtanio.new_document_from_string(&document_column);

         /* Set the document on the transaction so we're ready to process */
         _status = cat.set_document(_trans, _document);
         if _status NE 0 then put 'ERROR: set_document fails on obs:' &document_id;

         /* Apply the binary to the document */
         _status = cat.apply_concepts(_apply_settings, _trans);
         if _status NE 0 then put 'ERROR: apply_concepts fails on obs:' &document_id;

        /* Look for the concept matches */
        _num_matches = cat.get_number_of_concepts(_trans);
        _i = 0;
        do while (_i LT _num_matches);
            _name = cat.get_concept_name(_trans, _i);
            _full_path = cat.get_full_path_from_name(_trans, _name);
            _start_offset = cat.get_concept_start_offset(_trans, _i);
            _end_offset = cat.get_concept_end_offset(_trans, _i);
            _term = cat.get_concept(_trans, _i);
            _canonical_form = cat.get_concept_canonical_form(_trans, _i);

            output;

            _i = _i + 1;
        end;

  /* Now look for fact matches */
  _num_matches = cat.get_number_of_facts(_trans);
  _i = 0;
  _canonical_form = '';
  do while (_i LT _num_matches);
      _name = cat.get_fact_name(_trans, _i);
      _full_path = cat.get_full_path_from_name(_trans, _name);
      _start_offset = cat.get_fact_start_offset(_trans, _i);
      _end_offset = cat.get_fact_end_offset(_trans, _i);
      _term = cat.get_fact(_trans, _i);
```

```
        output;

        _i = _i + 1;
    end;
    _i = 0;

          /* Clean up resources */
          cat.clean_transaction(_trans);
          txtanio.free_object(_document);
       end;
  end;

  /*************************************************************************
   * Termination step that runs only once at the end.
   *************************************************************************/
   method term();
      /* clean up resources */
      cat.free_transaction(_trans);
      cat.free_apply_settings(_apply_settings);
      txtanio.free_object(_liti_binary);
   end;
endthread;
run;

/*****************************************************************************
 * Collect output data
 *****************************************************************************/
data &output_ds(
        keep=(
            &document_id
            _name
            _full_path
            _start_offset
            _end_offset
            _term
            _canonical_form
        )
        overwrite=yes
    );
    dcl THREAD workerth THRD;

    method run();
        set from THRD;
    end;

enddata;
run; quit;
```

Using the Category Model

The following code can be executed in SAS Contextual Analysis In-Database Scoring for Hadoop for context categorization.

```
/*****************************************************************************
```

```
 * Concept score code for Hadoop
 *
 * Copyright (C), 2016
 * SAS Institute Inc., Cary, N.C. 27513, U.S.A.  All rights reserved.
 ********************************************************************************/

/********************************************************************************
 * Set SAS environment variables that specify the location of the Hadoop Java
 * client API and configuration files. They are used to access Hadoop services.
 * The Java client API is provided by the Hadoop vendor in the form of JAR files.
 *
 * NOTE: The SAS Contextual Analysis In-Database Scoring for Hadoop
 *       Administrator's Guide denotes how to collect Hadoop Java client API jars
 *       and configuration files.
 ********************************************************************************/
options set=SAS_HADOOP_JAR_PATH="C:\path\to\Hadoop\jars";
options set=SAS_HADOOP_CONFIG_PATH="C:\path\to\Hadoop\conf";

/********************************************************************************
 * Execute a LIBNAME statement to assign a library reference to associate with
 * a Hadoop HDFS or HIVE server.
 *
 * Please contact your IT administrator for the HIVE server name.
 ********************************************************************************/
libname gridlib hadoop server="hivenode.com" user=xxxxx password=xxxxxxx;

/********************************************************************************
* Execute a LIBNAME statement to assign a library reference to
 * the location of the local SAS data set to be scored.
 *
 * If all the data sets used for scoring are already in HDFS, the libary may
 * not be needed.
 ********************************************************************************/
libname home "C:\path\to\local\dataset";

/********************************************************************************
 * Copy the data set to HDFS. Unique observation IDs are required and are
 * created as the field _document_id.
 *
 * If all the data sets used for scoring are already in HDFS, this step might
 * not be needed.
 ********************************************************************************/
data gridlib.input_dataset;
    set home.input_dataset;
    _document_id = _N_;
run;

/********************************************************************************
 * Set input/output macro variables.
 *
 * input_ds:         Name of input data set in Hadoop, including the HDFS libref
 * document_id:      Name of the unique document ID column in the input data set
 * document_column:  Name of the column to process in the input data set
 * liti_binary_path: Path to the LITI binary
 * output_ds:        Name of the output data set, including the HDFS libref
 *
```

```
 * NOTE: The SAS Contextual Analysis In-Database Scoring for
 *       Hadoop Administrator's Guide describes how to deploy Text Analytics models
 *       into clusters.
 ******************************************************************************/
%let input_ds = gridlib.input_dataset;
%let document_id = _document_id;
%let document_column = text_to_process;
%let mco_binary_path = '/path/to/mco/binary.mco';
%let output_ds = gridlib.category_out;

/******************************************************************************
 * Delete the output before starting (Optional)
 ******************************************************************************/
proc delete data=&output_ds; run;

/******************************************************************************
 * Score categories in Hadoop
 ******************************************************************************/
proc ds2 ds2accel=yes xcode=warning;

    /* These packages are part of the Text Analytics add-on and are installed */
    /* in the EP                                                               */
    require package tkcat; run;
    require package tktxtanio; run;

    /* The output of the thread program is the input of the data program      */
    THREAD workerth / overwrite=YES;

        dcl package tkcat cat();
        dcl package tktxtanio txtanio();

        dcl binary(8)     _apply_settings;
        dcl binary(8)     _document;
        dcl binary(8)     _mco_binary;
        dcl binary(8)     _trans;

        dcl double        _status;
        dcl double        _num_matches;
        dcl double        _num_terms;
        dcl double        _i;
        dcl double        _j;
        dcl double        _document_id;
        dcl varchar(1024) _name;
        dcl varchar(1024) _full_path;
        dcl double        _start_offset;
        dcl double        _end_offset;
        dcl varchar(1024) _term;

        retain            _apply_settings;
        retain            _mco_binary;
        retain            _trans;

  /*********************************************************************
   * Initialization step. Runs only once when starting.
   *********************************************************************/
  method init();
```

```
    _apply_settings = cat.new_apply_settings();
    _mco_binary = txtanio.new_on_content_server(&mco_binary_path);

    cat.set_categories_model(_apply_settings, _mco_binary);
    cat.set_return_match_positions_for_categories(_apply_settings, 1);
    cat.set_relevancy_type(_apply_settings, 1);

    _status = cat.initialize_categories(_apply_settings);
    if _status NE 0 then put 'ERROR: initialize_categories fails';

    _trans = cat.new_transaction();
 end;

/*********************************************************************
 * Run step. The method runs per row of input.
 *********************************************************************/
method run();
  set &input_ds(keep=(&document_column &document_id));

  /* Only process if document observation is not empty*/
  if &document_column NE ' ' then do;

    /* Initialize the document with the column data */
    _document = txtanio.new_document_from_string(&document_column);

    /* Set the document on the transaction so we're ready to process */
    _status = cat.set_document(_trans, _document);
    if _status NE 0 then put 'ERROR: set_document fails on obs:' &document_id;

    /* Apply the binary to the document */
    _status = cat.apply_categories(_apply_settings, _trans);
    if _status NE 0 then put 'ERROR: apply_categories fails on obs:' &document_id;

    /* Get the number of categories matches */
    _num_matches = cat.get_nb_matched_categories(_trans);
   _i = 0;
   do while (_i LT _num_matches);
     /* we used the name for both name & full_path for better alignment */
     /* with LITI code                                                  */
     _name = cat.get_category_name(_trans, _i);
     _full_path = _name;

     /* Get the number of term matches for current matched category */
     _num_terms = cat.get_nb_matched_terms_category(_trans, _i);
     _j = 0;
     do while (_j LT _num_terms);
         _start_offset = cat.get_matched_term_start_offset_category(_trans, _i, _j);
         _end_offset = cat.get_matched_term_end_offset_category(_trans, _i, _j);
         _term = cat.get_matched_term_category(_trans, _i, _j);

         output;

         _j = _j + 1;
      end;

      _i = _i + 1;
```

```
     end;

     /* Clean up resources */
     cat.clean_transaction(_trans);
     txtanio.free_object(_document);
   end;
end;

 /*************************************************************************
  * Termination step. Runs only once at end.
  *************************************************************************/
   method term();
      /* Clean up resources */
      cat.free_transaction(_trans);
      cat.free_apply_settings(_apply_settings);
      txtanio.free_object(_mco_binary);
   end;

endthread;
run;

/*****************************************************************************
 * Collect output data
 *****************************************************************************/
data &output_ds(
        keep=(
            &document_id
            _name
            _full_path
            _start_offset
            _end_offset
            _term
        )
        overwrite=yes
    );
    dcl THREAD workerth THRD;

    method run();
        set from THRD;
    end;

enddata;
run; quit;
```

Using the Sentiment Model

The following code can be executed in SAS Contextual Analysis In-Database Scoring for Hadoop for analyzing sentiment.

```
/*****************************************************************************
 * Sentiment score code for Hadoop
 *
 * Copyright (C), 2016
 * SAS Institute Inc., Cary, N.C. 27513, U.S.A.  All rights reserved.
```

```
 ****************************************************************************/

/****************************************************************************
 * Set SAS environment variables that specify the location of the Hadoop Java
 * client API and configuration files. They are used to access Hadoop services.
 * The Java client API is provided by the Hadoop vendor in the form of JAR files.
 *
 * NOTE: The SAS Contextual Analysis In-Database Scoring for Hadoop
 *       Administrator's Guide denotes how to collect Hadoop Java client API jars
 *       and configuration files.
 ****************************************************************************/
options set=SAS_HADOOP_JAR_PATH="C:\path\to\Hadoop\jars";
options set=SAS_HADOOP_CONFIG_PATH="C:\path\to\Hadoop\conf";

/****************************************************************************
 * Execute a LIBNAME statement to assign a library reference to associate with
 * a Hadoop HDFS or HIVE server.
 *
 * Please contact your IT administrator for the HIVE server name.
 ****************************************************************************/
libname gridlib hadoop server="hivenode.com" user=xxxxx password=xxxxxxx;

/****************************************************************************
 * Execute a LIBNAME statement to assign a library reference to
 * the location of the local SAS data set to be scored.
 *
 * If all the data sets used for scoring are already in HDFS, the libary might
 * not be needed.
 ****************************************************************************/
libname home "C:\path\to\local\dataset";

/****************************************************************************
 * Copy the data set to HDFS. Unique observation IDs are required and are
 * created as the field _document_id.
 *
 * If all the data sets used for scoring are already in HDFS, this step might
 * not be needed.
 ****************************************************************************/
data gridlib.input_dataset;
    set home.input_dataset;
    _document_id = _N_;
run;

/****************************************************************************
 * Set input/output macro variables.
 *
 * input_ds:         Name of input data set in Hadoop, including the HDFS libref
 * document_id:      Name of the unique document ID column in the input data set
 * document_column:  Name of the column to process in the input data set
 * sam_binary_path:  Path to the SAM binary
 * output_ds:        Name of the output data set, including the HDFS libref
 *
 * NOTE: The SAS Contextual Analysis In-Database Scoring for Hadoop
 *       Administrator's Guide describes how to deploy Text Analytics models
 *       into clusters.
 ****************************************************************************/
```

```
%let input_ds = gridlib.input_dataset;
%let document_id = _document_id;
%let document_column = text_to_process;
%let sam_binary_path = '/path/to/sam/binary.sam';
%let output_ds = gridlib.sentiment_out;

/*****************************************************************************
 * Delete output before starting (Optional)
 *****************************************************************************/
proc delete data=&output_ds; run;

/*****************************************************************************
 * Scores sentiment in Hadoop
 *****************************************************************************/
proc ds2 ds2accel=yes xcode=warning;

    /* These packages are part of the Text Analytics add-on and are installed */
    /* in the EP                                                               */
    require package tkling; run;
    require package tksent; run;
    require package tktxtanio; run;

    /* The output of the thread program is the input of the data program      */
    THREAD workerth / overwrite=YES;

        dcl package tkling ling();
        dcl package tksent sent();
        dcl package tktxtanio txtanio();

        dcl binary(8)  _apply_settings;
        dcl binary(8)  _document;
        dcl binary(8)  _sam;
        dcl binary(8)  _trans;

        dcl double     _status;
        dcl char(10)  _sentiment;
        dcl double    _sentiment_probability;

        retain        _apply_settings;
        retain        _sam_binary;
        retain        _trans;

 /*************************************************************************
 * Initialization step. Runs only once when starting.
 *************************************************************************/
  method init();
      _status = ling.set_language('english');
      if _status NE 0 then put 'ERROR: set_language fails';

      _status = ling.set_default_language_data();
      if _status NE 0 then put 'ERROR: set_default_language_data fails';

      _apply_settings = sent.new_apply_settings();

      _status = sent.set_threshold(_apply_settings, 0.5);
      if _status NE 0 then put 'ERROR: set_threshold fails';
```

```
      _sam = txtanio.new_on_content_server(&sam_binary_path);

      _status = sent.set_model(_apply_settings, _sam);
      if _status NE 0 then put 'ERROR: set_model fails';

      _trans = sent.new_transaction();

      _status = sent.initialize_sentiment(_apply_settings, _trans);
      if _status NE 0 then put 'ERROR: initialize_sentiment fails';
   end;

/*************************************************************************
* Run step. The method runs per row of input.
 *************************************************************************/
method run();
   set &input_ds(keep=(&document_column &document_id));

   /* Only process if document observation is not empty*/
      if &document_column NE ' ' then do;

      /* Initialize the document with the column data */
      _document = txtanio.new_document_from_string(&document_column);

     /* Set the document on the transaction so we're ready to process */
     _status = sent.set_document(_trans, _document);
     if _status NE 0 then put 'ERROR: set_document fails on document:' &document_id;

     /* Apply the binary to the document */
     _status = sent.apply_sentiment(_apply_settings, _trans);
     if _status NE 0 then put 'ERROR: apply_sentiment fails on document:' &document_id;

     /* Get current document sentiment polarity and probability */
     _sentiment = sent.get_sentiment(_trans);
     _sentiment_probability = sent.get_probability(_trans);

     output;

     /* Clean up resources */
               sent.clean_sentiment(_trans);
               txtanio.free_object(_document);
          end;
      end;

/*************************************************************************
* Termination step. Runs only once at end.
 *************************************************************************/
 method term();
    /* clean up resources */
    sent.free_transaction(_trans);
    sent.free_apply_settings(_apply_settings);
    txtanio.free_object(_sam);
 end;

endthread;
run;
```

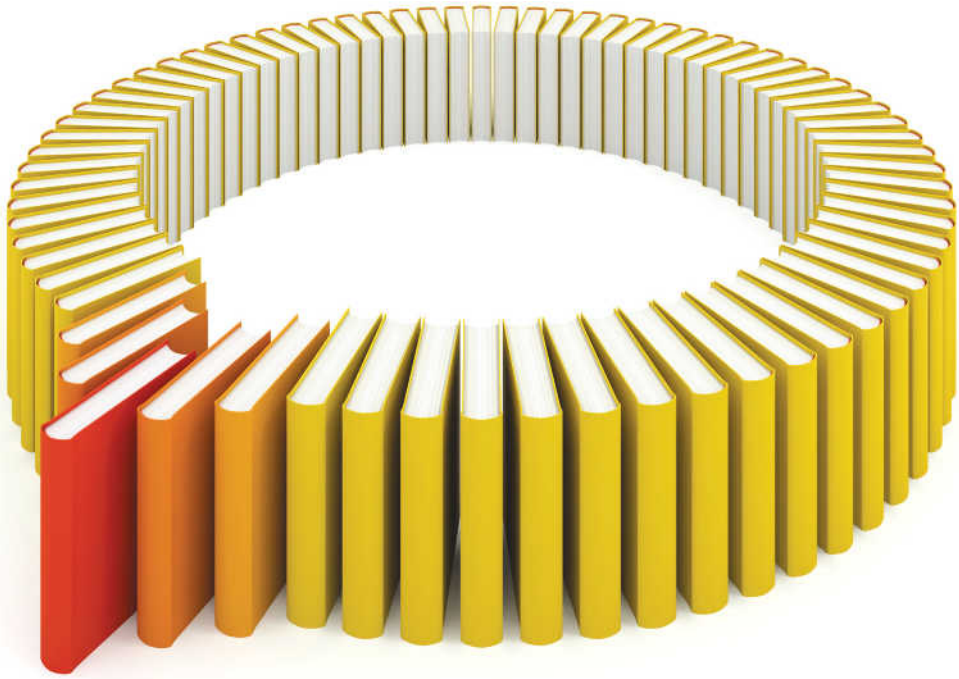
```
/*****************************************************************************
 * Collect output data
 *****************************************************************************/
data &output_ds(
        keep=(
            &document_id
            _sentiment
            _sentiment_probability
        )
        overwrite=yes
    );
    dcl THREAD workerth THRD;

    method run();
        set from THRD;
    end;

enddata;
run; quit;
```

Recommended Reading

- *SAS Contextual Analysis: Administrator's Guide*
- *SAS Contextual Analysis: User's Guide*
- *SAS Contextual Analysis In-Database Scoring for Hadoop: User's Guide*
- *SAS In-Database Products: Administrator's Guide*
- *SAS In-Database Products: User's Guide*
- SAS Global Forum Paper, "Exploring SAS Embedded Process Technologies on Hadoop"

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books



Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore
for additional books and resources.