

Regression Analysis Made Easy Using SAS® Studio

Kirk Paul Lafler, sasNerd

Zheyuan “Walter” Yu, Optimus Dental Supply

Nuoer “Norry” Lu, University of North Carolina, Chapel Hill

Juncheng Yi, University of Washington, Seattle

Yanzhang “Gavin” Chen, University of California, Los Angeles

Kai Kang, University of California, Berkeley

Yixuan “Jason” Xiang, University of California, Davis

Zhaowen “Daniel” Qian, Ten Square International Inc.

Swallow Xiaozhe Yan, US Education Without Borders

Abstract

SAS® OnDemand for Academics (ODA) provides students, faculty, and SAS learners with free access to SAS software and the SAS® Studio user interface using a web browser. SAS Studio provides a comprehensive and customizable integrated development environment (IDE) for all SAS users. To showcase SAS Studio’s many features, numerous techniques will be introduced to access, clean, transform, analyze, and visualize data using the point-and-click features found in SAS Studio’s Navigation Pane’s Tasks and Utilities. Plus, we’ll demonstrate the generated SAS code that is automatically produced from the point-and-click techniques. To obtain a high-level understanding of the datasets being used, we’ll demonstrate tasks associated with exploratory data analysis (EDA) to identify missing values, explore outliers, and evaluate trends in the data. Two types of regression will be demonstrated – simple linear regression where one independent variable is used to explain or predict the outcome of the dependent variable and multiple linear regression where two or more independent variables to explain or predict the outcome of the dependent variable to assist with decision-making activities. Key takeaways will be provided to assist in learning regression analysis techniques using effective examples.

Introduction

SAS® OnDemand for Academics (ODA) can be freely used by students, faculty, and anyone who wants to learn how to use SAS software. With SAS ODA’s cloud-based user interface, SAS Studio, users can access data and perform amazing extract, transform, and load (ETL) activities, data analysis, statistical analysis, reporting and data visualization, and other processing using SAS ODA with a common web browser. This paper introduces exploratory data analysis (EDA), data cleaning, data transformation, and regression analysis techniques using SAS ODA and SAS Studio’s powerful point-and-click user interface; the Navigation Pane consisting of Files and Folders, Tasks and Utilities, and Libraries; the SAS Programmer window; data access for accessing and retrieving data from SAS (SAS7BDAT) datasets; predefined tasks and utilities to perform exploratory data analysis (EDA) techniques to understand missing data, data anomalies, and trends in data; and perform regression analysis to help with decision-making activities, allocate resources more efficiently, identify the factors (or variables) that matter most and which can be ignored, and turning the collected data into actionable insights.

Datasets Used in the Examples

The example datasets and data files used in this paper include the Heart and Heart_MedCenter data files and SAS datasets. (For a thorough explanation of the step-by-step point-and-click operations associated with accessing and reading different types of data files (e.g., TSV, CSV, XLSX, JSON, and SAS7BDAT). The Heart and Heart_MedCenter datasets were presented in an earlier paper called, [Data Access Made Easy Using SAS® Studio](#).

File Type Definition	
SAS Dataset (SAS7BDAT)	A proprietary SAS (SAS7BDAT) data format that contains data values that are created, organized, and stored as a table of rows and columns in a SAS library (e.g., WORK, SASUSER, and User-assigned) where processing is performed by SAS software.

HEART_MEDCENTER Data Files and SAS Dataset (5 Rows and 5 Variables)

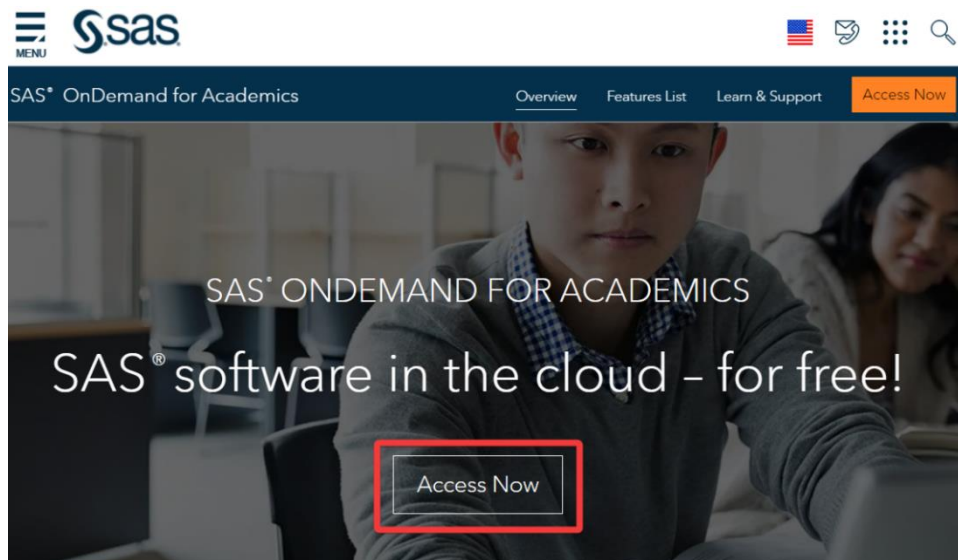
MedCtrID	MedicalCenter	City	State	Zip
CA92101	San Diego Medical Center	San Diego	CA	92101
CA92037	La Jolla Heart Institute	La Jolla	CA	92037
CA90025	Los Angeles Medical Center	Los Angeles	CA	90025
CA94105	San Francisco Medical Center	San Francisco	CA	94105
NV89109	Las Vegas Health Center	Las Vegas	NV	89109

HEART Data Files and SAS Dataset (5,209 Rows and 18 Variables)

MedCtrID	Status	DeathCause	AgeCHDdiag	Sex	AgeAtStart	Height	Weight	Diastolic	Systolic	MRW	Smoking	AgeAtDeath	Cholesterol	Chol_Status	BP_Status	Weight_Status	Smoking_Status
CA94105	Dead	Other	.	Female	29	62.50	140	78	124	121	0	55	.	.	Normal	Overweight	Non-smoker
CA94105	Dead	Cancer	.	Female	41	59.75	194	92	144	183	0	57	181	Desirable	High	Overweight	Non-smoker
CA94105	Alive		.	Female	57	62.25	132	90	170	114	10	.	250	High	High	Overweight	Moderate (6-15)
CA92307	Alive		.	Female	39	65.75	158	80	128	123	0	.	242	High	Normal	Overweight	Non-smoker
CA90025	Alive		.	Male	42	66.00	156	76	110	116	20	.	281	High	Optimal	Overweight	Heavy (16-25)
CA92307	Alive		.	Female	58	61.75	131	92	176	117	0	.	196	Desirable	High	Overweight	Non-smoker
CA94105	Alive		.	Female	36	64.75	136	80	112	110	15	.	196	Desirable	Normal	Overweight	Moderate (6-15)
CA90025	Dead	Other	.	Male	53	65.50	130	80	114	99	0	77	276	High	Normal	Normal	Non-smoker
CA92307	Alive		.	Male	35	71.00	194	68	132	124	0	.	211	Borderline	Normal	Overweight	Non-smoker
CA90025	Dead	Cerebral Vascular Disease	.	Male	52	62.50	129	78	124	106	5	82	284	High	Normal	Normal	Light (1-5)
NV89109	RIP		.	Male	39	66.25	179	76	128	133	30	.	225	Borderline	Normal	Overweight	Very Heavy (> 25)
CA92307	Alive		57	Male	33	64.25	151	68	108	118	0	.	221	Borderline	Optimal	Overweight	Non-smoker
CA92307	Alive		55	Male	33	70.00	174	90	142	114	0	.	188	Desirable	High	Overweight	Non-smoker
CA90025	Alive		79	Male	57	67.25	165	76	128	118	15	.	.	.	Normal	Overweight	Moderate (6-15)
NV89109	RIP		66	Male	44	69.00	155	90	130	105	30	.	292	High	High	Normal	Very Heavy (> 25)
CA94105	Alive		.	Female	37	64.50	134	76	120	108	10	.	196	Desirable	Normal	Normal	Moderate (6-15)
NV89109	RIP		.	Male	40	66.25	151	72	132	112	30	.	192	Desirable	Normal	Overweight	Very Heavy (> 25)
CA90025	Dead	Cancer	56	Male	56	67.25	122	72	120	87	15	72	194	Desirable	Normal	Under	Moderate (6-15)
CA94105	Alive		.	Female	42	67.75	162	96	138	119	1	.	200	Borderline	High	Overweight	Light (1-5)
NV89109	RIP	Coronary Heart Disease	74	Male	46	66.50	157	84	142	116	30	76	233	Borderline	High	Overweight	Very Heavy (> 25)
CA94105	Alive		.	Female	37	66.25	148	78	110	112	15	.	192	Desirable	Optimal	Overweight	Moderate (6-15)

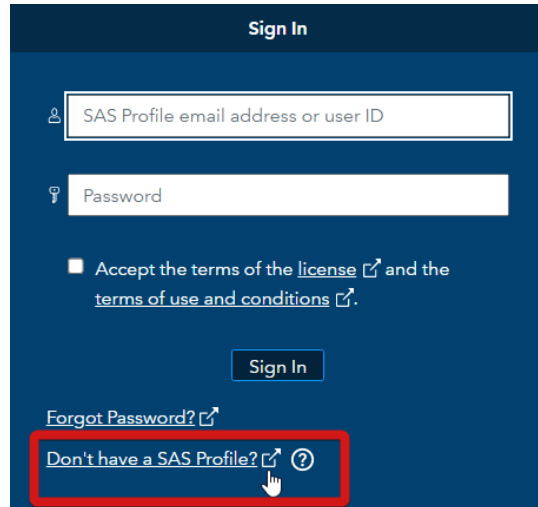
SAS Studio – A Cloud-based Integrated Development Environment (IDE)

SAS OnDemand for Academics (ODA) provides learners and educators with a comprehensive cloud- and web-based user interface called SAS Studio. SAS Studio provides numerous user-friendly features to help users become more productive while using the SAS ODA. To begin, open one of the supported web browsers (e.g., Google Chrome, Mozilla Firefox or Apple Safari) to access SAS ODA by clicking the following hyperlink, https://www.sas.com/en_us/software/on-demand-for-academics.html, and then clicking the “Access Now” as shown, below.

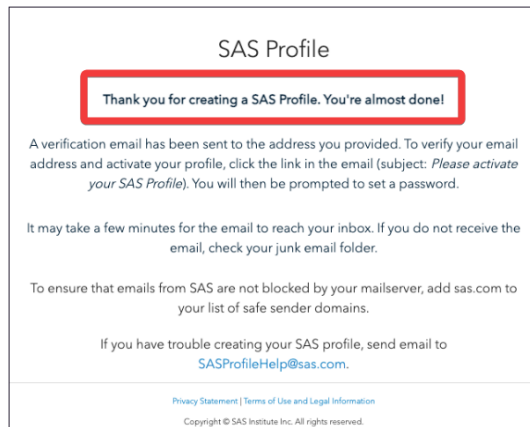


Step 1 – Create a SAS Profile

The SAS OnDemand for Academics (ODA) Sign In dialog window will display as shown, below. Before accessing SAS ODA, you will need to create a SAS Profile. If you are already a SAS user and have set up a SAS profile account, then you can proceed to register to use SAS ODA. By entering your SAS Profile email address or user ID along with your Password in the designated boxes. If you are a new SAS user or have never created a SAS Profile then you will need to click the “Don’t have a SAS Profile?” hyperlink shown, below.

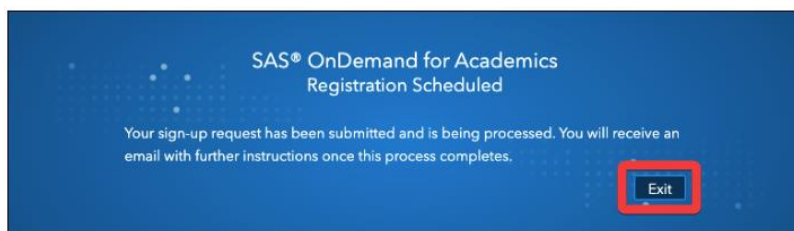


After entering the requested information to create your SAS Profile, a message will display on your screen, below.



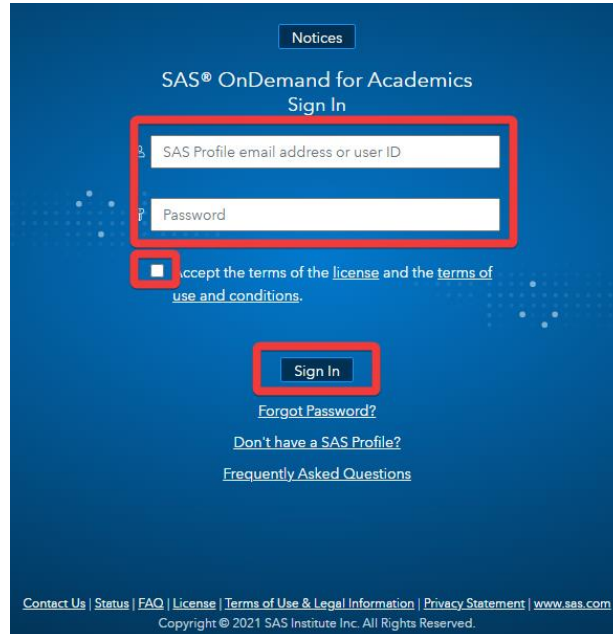
Step 2 – Register to Use SAS OnDemand for Academics (ODA)

After successfully creating a SAS Profile, you can register to use SAS OnDemand for Academics (ODA). You should then return to the SAS OnDemand for Academics (ODA) page where you will be prompted to select your home region and click **Submit**. A confirmation page will then appear like the one shown below, allowing you to finalize the process by clicking the **Exit** button. SAS will also send a follow-up email with your User ID so you can then enter this User ID or email address to access SAS ODA and SAS Studio software.

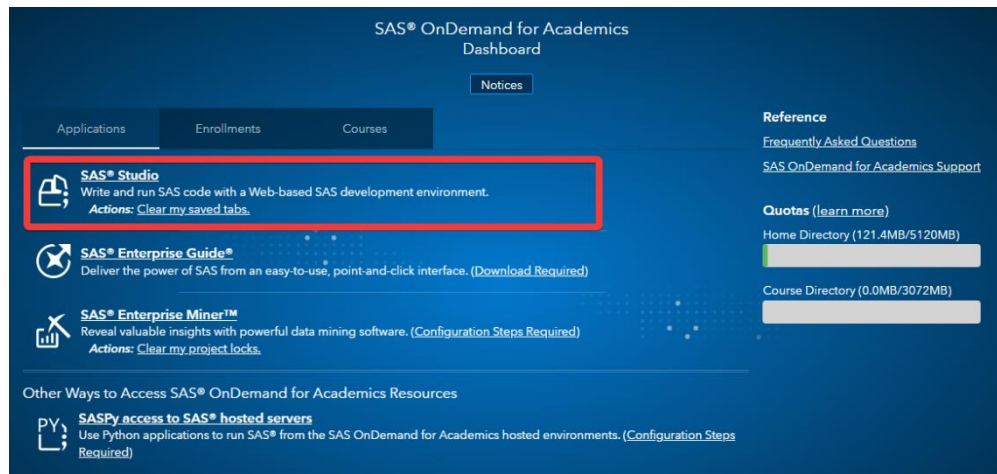


Signing Into SAS OnDemand for Academics (ODA) and Accessing SAS Studio

After successfully registering to use SAS OnDemand for Academics (ODA), you can then sign in with your User ID and password credentials in the appropriate fields, check the box associated with accepting the terms of the license and the terms of use and conditions, and click the **Sign In** button as shown, below.

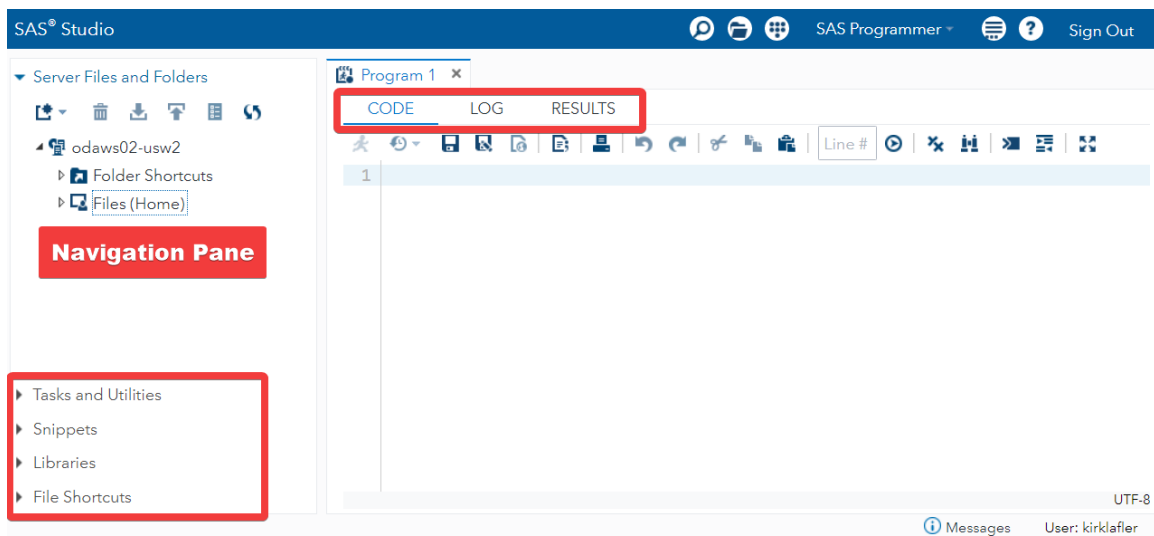


The SAS ODA dashboard will then display with important information about your account including permissions, enrollments, courses, self-help references, and storage space quotas. When ready, click the **SAS Studio** hyperlink shown, below.



SAS Studio User Interface

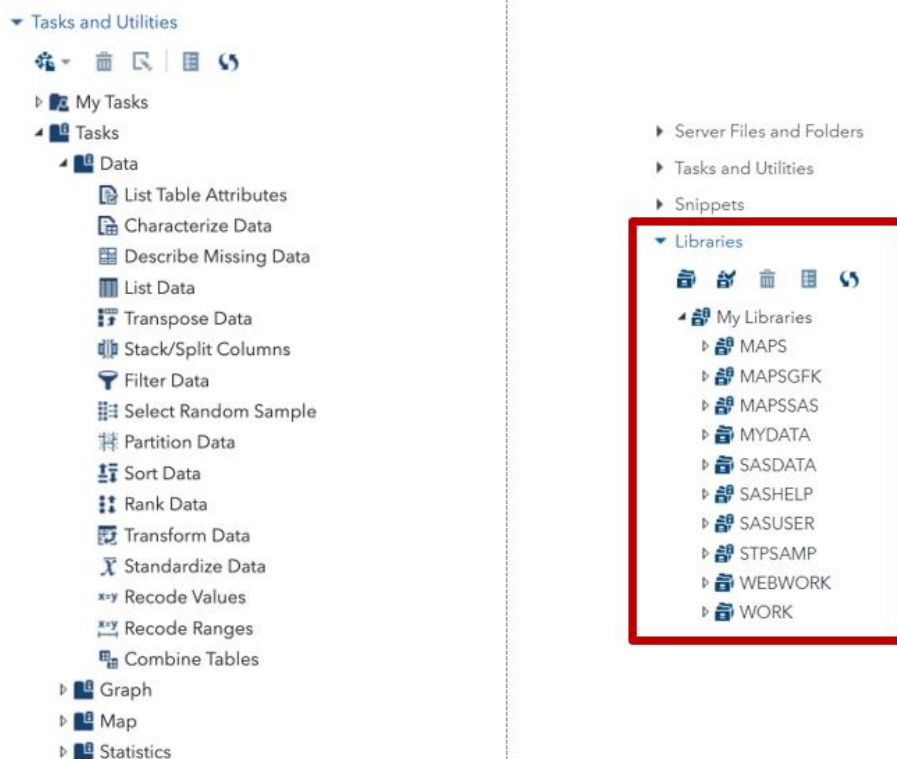
SAS Studio's powerful and easy-to-use interface provides users with a comprehensive integrated development environment (IDE). The SAS Studio interface is divided into several parts that help make user interaction easier, Navigation pane, and Work area more convenient. Let's explore the different parts of SAS Studio to better understand what they're used for. After signing into SAS Studio, **Server Files and Folders** provide users with the ability to upload local data files. There are four more dropdown menus below Server Files and Folders, two of which will be emphasized, **Tasks and Utilities**, and **Libraries**.



Navigation Pane

When clicking on the Navigation pane's drop-down arrow next to Tasks, more options expand as shown, below. SAS Studio's built-in point-and-click interface helps make working with SAS datasets, text-delimited data files, CSV data files, Excel data files, JSON data files, and program code easier with a powerful toolkit of predefined tasks that enable users to list table attributes, characterize data, identify missing data and outliers, transform datasets using merge/join and transpose processes, perform data analytics, and several other tasks.

Another Navigation Pane drop-down is Libraries. A SAS library is a collection of one or more SAS datasets that are stored, referenced, and processed by SAS software. Specifically, the SASHELP library stores a variety of SAS-supplied datasets for use by students, faculty, and SAS learners to explore and learn from. We will demonstrate using a modified version of the SASHELP library HEART dataset along with a user-created HEART_MEDCENTER dataset in several examples in this paper.

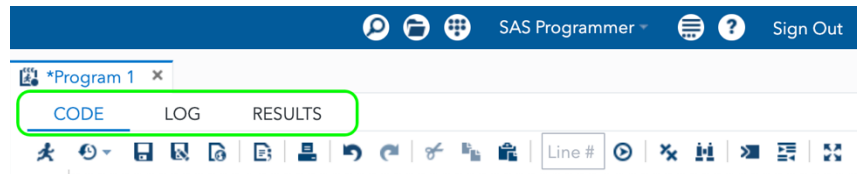


Program Window: Code, Log, and Results

The SAS Studio Program window provides users with Code, Log, and Results tabs. A description of each tab appears below.

Code Editor Tab

SAS Studio includes a color-coded, syntax-checking editor for editing new or existing SAS programs. The editor includes a wide variety of features such as autocompletion, automatic formatting, and pop-up syntax help. With the code editor, you can write, run, and save SAS programs.

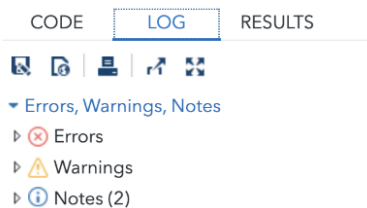


There are a variety of icons in the Code tab toolbar. Most of these icons will display tooltips or their functionality when hovering the mouse on them. Below are descriptions of some commonly used SAS Studio-specific icons:

Icon	Tooltip	Execution
	Run all or selected codes	Executes all lines or highlighted lines of codes in the Code window.
	Submission history	Displays a history of executed statements and will rerun the code once selected on the previous statement.
	Save program	Save all codes.
	Program summary	An HTML file that opens in a separate browser tab includes information about the program execution, the complete SAS source code, the complete SAS log, and the results.
	Clear all code	Clears all code in the current program's code editor.

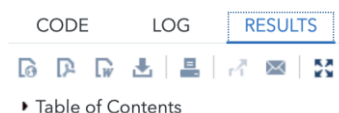
Log Tab

It is crucial to develop a routine habit of checking the Log tab after each code execution as it is a tremendous tool for helping users during troubleshooting. After executing the program code, the SAS Log tab provides useful information about Errors, Warnings, and Notes in corresponding red, yellow, and blue colors.



Results Tab

By clicking the Results tab, you can view any output results from output-producing procedures. SAS software automatically produces HyperText Markup Language (HTML) results as the “default” output format, along with any graphical, tabular, and statistical information when it be requested, as shown, below.



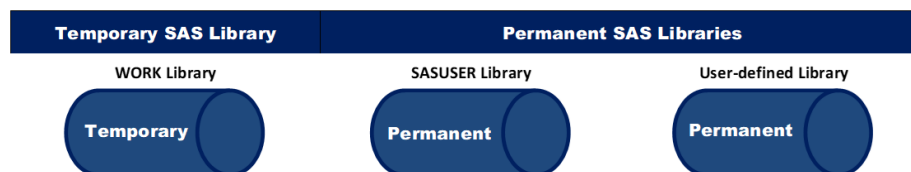
The MEANS Procedure

Analysis Variable : Smoking							
Sex	Smoking Status	N Obs	N	Mean	Std Dev	Minimum	Maximum
Female	Heavy (16-25)	339	339	20.3834808	1.3325105	20.0000000	25.0000000
	Light (1-5)	422	422	4.1279621	1.6535358	1.0000000	5.0000000
	Moderate (6-15)	340	340	12.6764706	2.4974393	10.0000000	15.0000000
	Non-smoker	1682	1682	0	0	0	0
	Very Heavy (> 25)	73	73	33.9726027	4.7110172	30.0000000	45.0000000
Male	Heavy (16-25)	707	707	20.7001414	1.7363103	20.0000000	25.0000000
	Light (1-5)	157	157	4.4649682	1.3659254	1.0000000	5.0000000
	Moderate (6-15)	236	236	12.9449153	2.4653202	10.0000000	15.0000000
	Non-smoker	819	819	0	0	0	0
	Very Heavy (> 25)	398	398	36.7336683	7.7107287	30.0000000	60.0000000

Temporary versus Permanent SAS Datasets

In the SAS world, the location of your data is everything. This concept is essential for SAS users to understand when using SAS OnDemand for Academics (ODA), or any other SAS product. But what does it mean? Data can be stored on a variety of fixed or removable storage devices including CDs, DVDs, Blu-ray, USB flash drives, tape, external hard drives, NAS storage, and in the cloud. The data access demonstrations presented in this paper use data that is stored in the cloud.

Another important concept that users should become familiar with is which SAS library a dataset is stored in. The library where a SAS dataset is stored determines if the dataset is temporary or permanent. If this sounds a bit confusing, then the good news is that, in time and with practice, your comfort level working with temporary and permanent datasets will become easier with use. The SAS WORK library is classified as temporary, and all temporary SAS datasets are automatically removed (or deleted) at the end of a SAS session. A SAS dataset that is stored in either the SASUSER library or in a user-defined folder in SAS Studio is classified as permanent and, as a result, is accessible after the end of a SAS session, from one session to another, or until the SAS dataset is removed (or deleted).

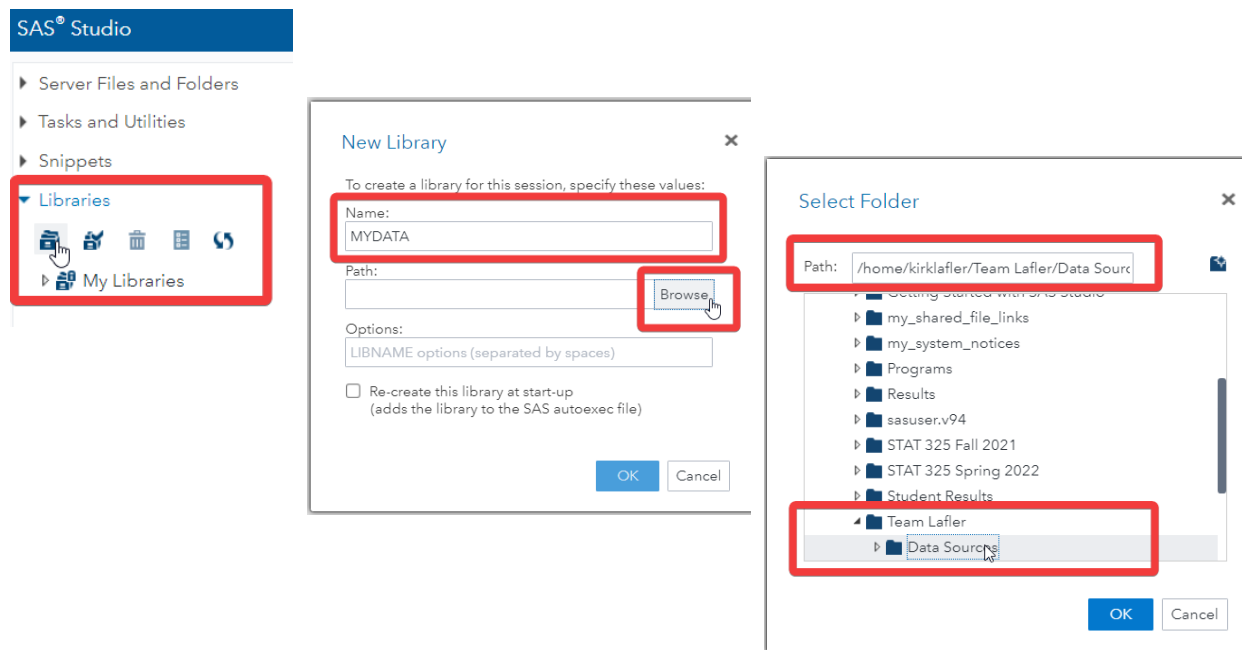


SAS Studio’s Point-and-Click Navigation

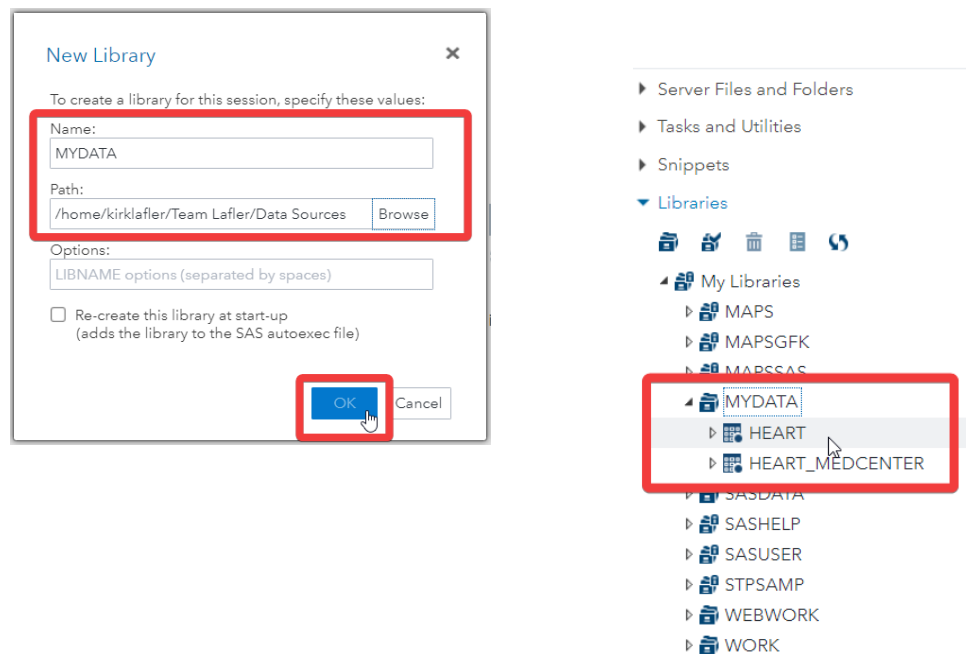
SAS Studio’s point-and-click navigation provides users with a powerful, flexible, and easy to use approach to auto-generating SAS code for all types of SAS processing. The objective of this paper is to demonstrate the many capabilities that SAS OnDemand for Academics (ODA) and SAS Studio offers users including creating new SAS libraries; establishing library references (LIBREFs); uploading SAS datasets, tab-delimited, CSV, and Excel data files in the cloud; importing tab-delimited, CSV, and Excel data files to SAS datasets using tasks and utilities; and producing results using the Navigation pane.

Assigning a New SAS Library

Using the Navigation pane’s point-and-click features, select **Libraries** → **New Library icon** → **Import Data** to Using the Navigation pane’s point-and-click features, users can assign a new SAS library, a libref, and the path to where the data is in the cloud, as shown, below.



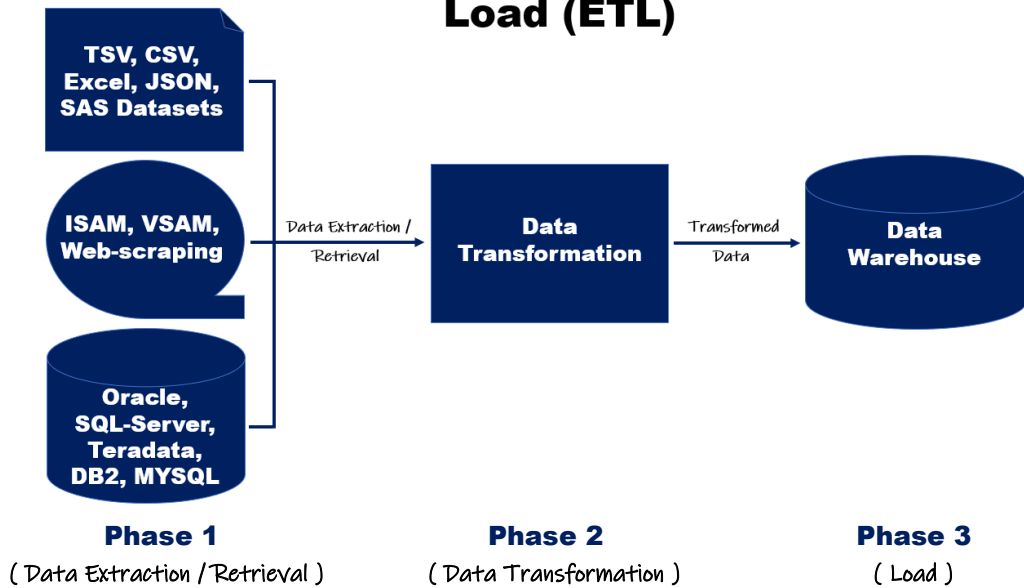
User-assigned library references (LIBREFs) along with their specific paths were specified using the **New Library** window. Specifically, the LIBREF, **MYDATA**, along with its path to identify where the Heart and Heart_MedCenter datasets are stored in the cloud are assigned, as shown, below.



The Extract, Transform, and Load (ETL) Process

The extract, transform, and load (ETL) process involves moving / migrating data from various sources into a data warehouse. The best way to understand how ETL works is to examine what happens in each phase of the process. The ETL process and its three phases are displayed in the figure below.

Extract, Transform, and Load (ETL)

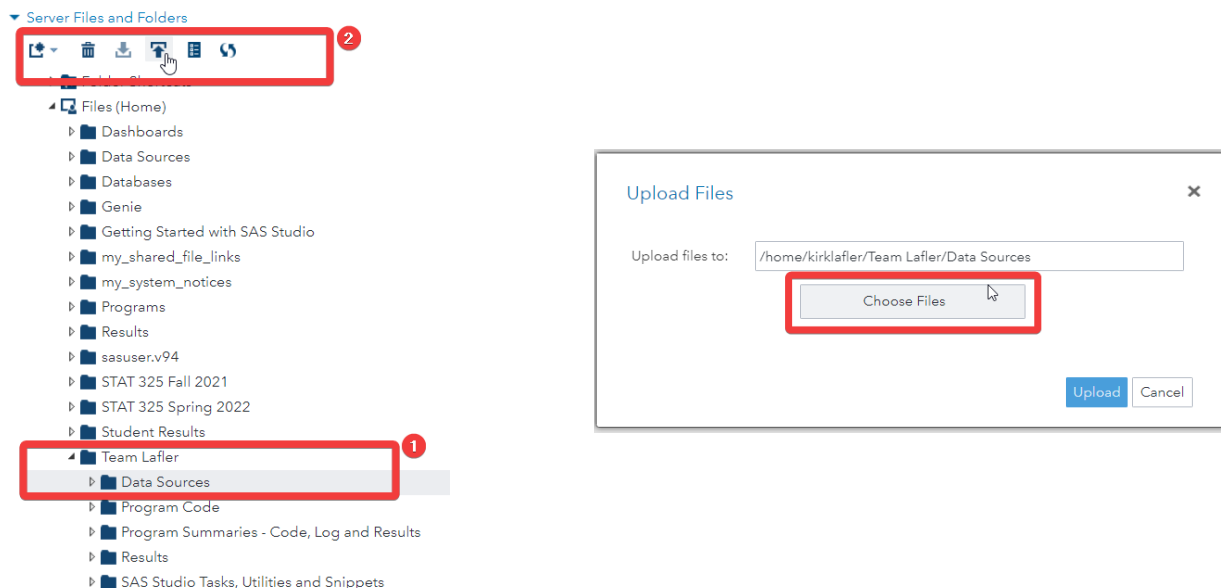


Data Extraction / Retrieval Phase

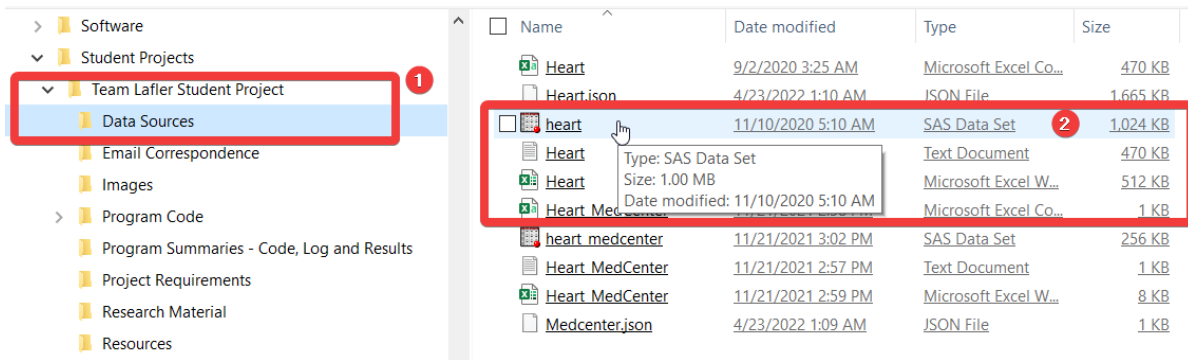
The first phase of the ETL process involves data extraction / retrieval from various sources (e.g., SAS datasets; text data files including tab-separated value (TSV) and comma-separated values (CSV); Excel; JavaScript Object Notation (JSON) data files and SAS datasets; legacy systems (e.g., ISAM, VSAM, etc.); web scraping; RDBMS tables (e.g., Oracle, SQL-Server, Teradata, DB2, MYSQL, etc.); cloud-based repositories; and other data file management systems. Specifically, the objective of the data extraction / retrieval phase is to access and extract the desired data from the various into a consistent and usable format enabling successful data transformation.

Task: Uploading SAS Datasets and Other Data Files to the Cloud

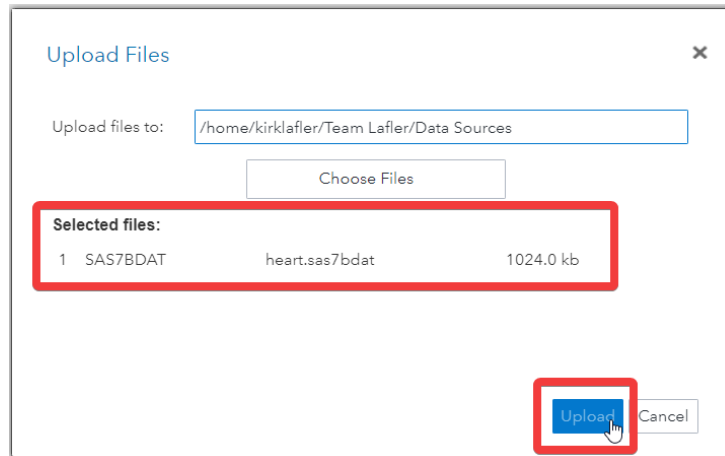
To upload SAS datasets and data files to SAS Studio in the cloud, the following point-and-click steps can be followed, as shown, below. In step #1, click the desired folder / sub-folder where you want a SAS dataset or data file uploaded to. In step #2, click the Upload control tool to display the **Upload Files** window. Then, click the **Choose Files** button, as shown, below.



After clicking the **Choose Files** button, navigate to where your data is stored (step #1), and then select the SAS dataset you want to upload (step #2), as shown below.



After selecting the SAS dataset from the list of data files you want uploaded to the cloud, the **Upload Files** window will then display the name of the selected dataset. Finally, clicking the **Upload** button launches the upload process, as shown, below.



Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is the process of conducting an initial investigation of the data including the visualization of general patterns in the data such as:

- the identification of each variable's discrete (or unique) values by characterizing data using PROC FREQ to display frequency distributions.
- the production of summary descriptive statistics such as MIN, MAX, RANGE, MEAN, MEDIAN, MODE, STD, and VARIANCE using PROC MEANS and PROC UNIVARIATE.
- the number of missing values (NMISS) using PROC FORMAT and PROC MEANS.
- the display of outliers with BoxPlots using PROC SGPLOT.

Task: Characterize Data

The Characterize Data task allows users to describe data in manners useful for data mining purposes and to better understand what is in the data. SAS ODA and SAS Studio provide users with point-and-click capabilities to auto-generate PROC FREQ code for data characterization, as shown below.

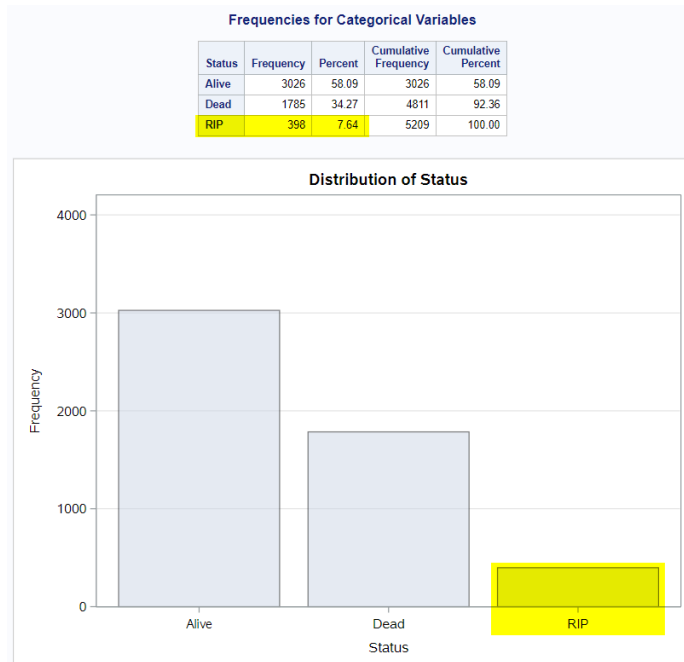
The screenshot shows the SAS Studio interface for the 'Characterize Data' task. On the left, the 'DATA' tab is selected, showing the dataset 'MYDATA.HEART_WITH_MESSY_DATA'. Under 'AUTOMATIC CHARACTERIZATION', several variables are listed: MedCtrID, Status, DeathCause, Sex, Chol_Status, BP_Status, and Weight_Status. The 'CUSTOM CHARACTERIZATION' section shows 'Categorical variables' set to 'Column'. On the right, the 'CODE' tab displays the generated SAS code:

```

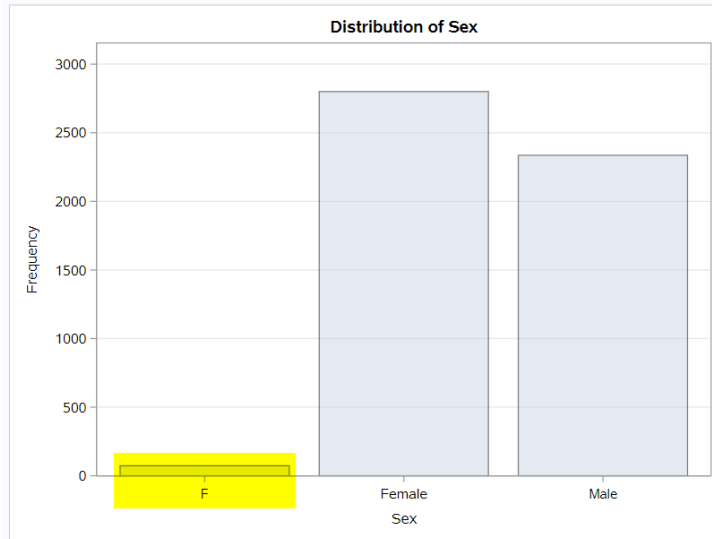
1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '6/10/23, 4:00 PM'
6 * Generated by 'kirklafler'
7 * Generated on server 'ODAWS04-USW2.ODA.SAS.COM'
8 * Generated on SAS platform 'Linux LIN X64 3.10.0-1062.9.1.e17.x86_64'
9 * Generated on SAS version '9.04.01M7P08062020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
11 * Generated on web client 'https://odamid-usw2.oda.sas.com/SASStudio/main?loc
12 *
13 */
14
15 ods noproctitle;
16
17 /*** Analyze categorical variables ***/
18 title "Frequencies for Categorical Variables";
19
20 proc freq data=MYDATA.HEART_WITH_MESSY_DATA;
21     tables MedCtrID Status DeathCause Sex Chol_Status BP_Status Weight_Status
22           Smoking_Status / plots=(freqplot) missing;
23 run;
    
```

The screenshot shows the 'OPTIONS' tab for the 'Characterize Data' task. Under 'CATEGORICAL VARIABLES', the following options are checked: 'Frequency table', 'Frequency chart', and 'Treat missing values as valid level'. The 'Limit categorical values' option is unchecked. Under 'NUMERIC VARIABLES', 'Descriptive statistics' and 'Histogram' are checked. The 'DATE VARIABLES' section is collapsed.

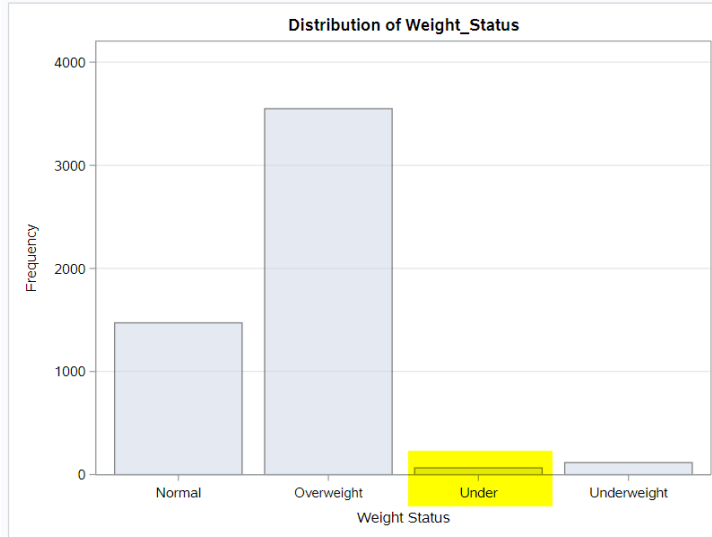
Results:



Sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	73	1.40	73	1.40
Female	2800	53.75	2873	55.15
Male	2336	44.85	5209	100.00



Weight Status				
Weight_Status	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Normal	1472	28.29	1472	28.29
Overweight	3550	68.23	5022	96.52
Under	65	1.25	5087	97.77
Underweight	116	2.23	5203	100.00
Frequency Missing = 6				



Task: Summary Descriptive Statistics

The Summary Statistics task allows users to summarize or describe the characteristics of a dataset. SAS ODA and SAS Studio provide users with point-and-click capabilities to auto-generate PROC MEANS and PROC UNIVARIATE code to produce three basic categories of measurements: 1) measures of central tendency, 2) measures of variability (or spread), and 3) frequency distributions.

The screenshot displays the SAS Studio interface for the Summary Statistics task. On the left, the 'Tasks and Utilities' pane shows the 'Summary Statistics' task selected under the 'Statistics' category. The main workspace is divided into several panes:

- DATA:** MYDATA.HEART_WITH_MESSY_DATA
- ROLES:** Analysis variables: Smoking
- Classification variables:** Column
- CODE:** Contains the following SAS code:


```

proc means data=MYDATA.HEART_WITH_MESSY_DATA chartype mean std min max median n
miss var mode range vardef=df qmethod=os;
var Smoking;
run;

proc univariate data=MYDATA.HEART_WITH_MESSY_DATA vardef=df noprint;
var Smoking;
  histogram Smoking;
run;

/* Graph template to construct combination histogram/boxplot */
proc template;
  define statgraph histogram;
    dynamic AVAR ByVarInfo;
    begingraph;
    entrytitle "Distribution of " AVAR ByVarInfo;
    layout lattice / rows=2 columndatarange=union rowgutter=0 rowweights=(0.75
0.25);
    layout overlay / yaxisopts=(offsetmax=0.1) xaxisopts=(display=none);
    histogram AVAR /;
    endlayout;
    layout overlay /;
    BoxPlot Y=AVAR / orient=horizontal;
    endlayout;
    endlayout;
    endgraph;
end;
run;
            
```
- OPTIONS:** Shows the configuration for the task:
 - STATISTICS:**
 - Basic Statistics: Mean, Standard deviation, Minimum value, Maximum value, Median, Number of observations, Number of missing values
 - Additional Statistics: Standard error, Variance, Mode, Range, Sum, Sum of weights, Confidence limits for the mean, Coefficient of variation, Skewness, Kurtosis
 - Percentiles: (None selected)
 - PLOTS:**
 - Histogram
 - Add normal density curve
 - Add kernel density estimate
 - Add inset statistics
 - Position at: Lower right
 - Histogram and box plot

Task: Describe Missing Data

The Describe Missing Data task allows users to examine missing data or missing values in a dataset. SAS ODA and SAS Studio provide users with point-and-click capabilities to auto-generate PROC FORMAT, PROC FREQ, and PROC PRINT code to produce the number (or frequency) of missing and non-missing data.

```

17 proc format;
18     value $_cmisprint " =" " other="Non-missing";
19 run;
20
21 proc freq data=MYDATA.HEART_WITH_MESSY_DATA;
22     title3 "Missing Data Frequencies";
23     title4 h=2 "Legend: ., A, B, etc = Missing";
24     format MedCtrID Status DeathCause Sex Chol_Status BP_Status Weight_Status
25           Smoking_Status $_cmisprint.;
26     tables MedCtrID Status DeathCause Sex Chol_Status BP_Status Weight_Status
27           Smoking_Status / missing nocum;
28 run;
29
30 proc freq data=MYDATA.HEART_WITH_MESSY_DATA noprint;
31     table MedCtrID * Status * DeathCause * Sex * Chol_Status * BP_Status *
32           Weight_Status * Smoking_Status / missing out=Work_MissingData;
33     format MedCtrID Status DeathCause Sex Chol_Status BP_Status Weight_Status
34           Smoking_Status $_cmisprint.;
35 run;
36
37 proc print data=Work_MissingData_noobs label;
38     title3 "Missing Data Patterns across Variables";
39     title4 h=2 "Legend: ., A, B, etc = Missing";
40     format MedCtrID Status DeathCause Sex Chol_Status BP_Status Weight_Status
41           Smoking_Status $_cmisprint.;
42     label count="Frequency" percent="Percent";
43 run;
44

```

Results:

Missing Data Frequencies
Legend: ., A, B, etc = Missing

Status	Frequency	Percent
Non-missing	5209	100.00

Sex	Frequency	Percent
Non-missing	5209	100.00

Weight Status		
Weight_Status	Frequency	Percent
.	6	0.12
Non-missing	5203	99.88

Smoking	Frequency	Percent
.	36	0.69
Non-missing	5173	99.31

Smoking Status		
Smoking_Status	Frequency	Percent
.	36	0.69
Non-missing	5173	99.31

Missing Data Patterns across Variables
Legend: ., A, B, etc = Missing

Status	Sex	Weight Status	Smoking	Smoking Status	Frequency	Percent
Non-missing	Non-missing	Non-missing	Non-missing	Non-missing	6	0.1152
Non-missing	Non-missing	Non-missing	.	Non-missing	36	0.6911
Non-missing	Non-missing	Non-missing	Non-missing	Non-missing	5167	99.1937

Data Cleaning / Standardization

Data cleaning is the process of remediating or removing incorrect, corrupted, duplicate, incomplete, or incorrectly formatted data in a dataset.

Task: Recode (or Standardize) Values for Status

The Recode (or Standardize) Values task allows users to remediate or remove incorrect, corrupted, duplicate, incomplete, or incorrectly formatted values for the Status variable. SAS ODA and SAS Studio provide users with point-and-click capabilities to auto-generate DATA step code to remediate or remove (cleanup) identified Status data issues for improved standardization.

```

1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '6/10/23, 8:18 PM'
6 * Generated by 'kirklafler'
7 * Generated on server 'ODAWS04-USW2.ODA.SAS.COM'
8 * Generated on SAS platform 'Linux LIN X64 3.10.0-'
9 * Generated on SAS version '9.04.01M7P08062020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10
11 * Generated on web client 'https://odamid-usw2.oda
12 *
13 */
14
15 data WORK.HEART_CLEANED_STATUS;
16     length Status $ 5;
17     set MYDATA.HEART_WITH_MESSY_DATA;
18
19     select (Status);
20         when ('RIP') Status='Dead';
21         otherwise Status=Status;
22     end;
23 run;
    
```

Recode values: (minimum 1 row)	
Old value	New value
RIP	Dead

Task: Recode (or Standardize) Values for Sex

The Recode (or Standardize) Values task allows users to remediate or remove incorrect, corrupted, duplicate, incomplete, or incorrectly formatted values for the Sex variable. SAS ODA and SAS Studio provide users with point-and-click capabilities to auto-generate DATA step code to remediate or remove (cleanup) identified Sex data issues for improved standardization.

The screenshot shows the SAS Studio interface for the 'Recode Values' task. The left sidebar contains a 'Tasks and Utilities' menu with 'Recode Values' selected. The main window is divided into three tabs: 'DATA', 'VALUES', and 'INFORMATION'. The 'DATA' tab is active, showing the input data set 'MYDATA.HEART_WITH_MESSY_DATA' and the output data set 'WORK.HEART_CLEANED_SEX'. The 'ROLES' section shows 'Sex' as a character variable. The 'OUTPUT DATA SET' section shows the option to 'Write to another data set'. The 'CODE' tab shows the generated SAS DATA step code:

```

1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '6/10/23, 9:25 PM'
6 * Generated by 'kirklafler'
7 * Generated on server 'ODAWS04-USW2.ODA.SAS.COM'
8 * Generated on SAS platform 'Linux LIN X64 3.10.0-
9 * Generated on SAS version '9.04.01M7P08062020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 16
11 * Generated on web client 'https://odamid-usw2.oda
12 *
13 */
14
15 data WORK.HEART_CLEANED_SEX;
16   length Sex $ 6;
17   set MYDATA.HEART_WITH_MESSY_DATA;
18
19   select (Sex);
20     when ('F') Sex='Female';
21     otherwise Sex=Sex;
22 end;
23 run;
    
```

The screenshot shows the 'VALUES' tab in the 'Recode Values for Sex.ctlk' task. It displays a table for mapping old values to new values:

Old value	New value
F	Female

Task: Recode (or Standardize) Values for Weight_Status

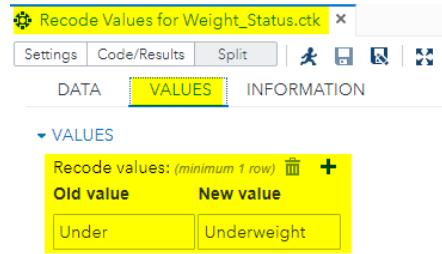
The Recode (or Standardize) Values task allows users to remediate or remove incorrect, corrupted, duplicate, incomplete, or incorrectly formatted values for the Weight_Status variable. SAS ODA and SAS Studio provide users with point-and-click capabilities to auto-generate DATA step code to remediate or remove (cleanup) identified Weight_Status data issues for improved standardization.

The screenshot shows the SAS Studio interface for the 'Recode Values' task. The left sidebar contains a 'Tasks and Utilities' menu with 'Recode Values' selected. The main window is divided into three tabs: 'DATA', 'VALUES', and 'INFORMATION'. The 'DATA' tab is active, showing the input data set 'MYDATA.HEART_WITH_MESSY_DATA' and the output data set 'WORK.HEART_CLEANED_Weight_Status'. The 'ROLES' section shows 'Weight_Status' as a character variable. The 'OUTPUT DATA SET' section shows the option to 'Write to another data set'. The 'CODE' tab shows the generated SAS DATA step code:

```

1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '6/10/23, 9:36 PM'
6 * Generated by 'kirklafler'
7 * Generated on server 'ODAWS04-USW2.ODA.SAS.COM'
8 * Generated on SAS platform 'Linux LIN X64 3.10.0-1062.9.1.e17.x86_64'
9 * Generated on SAS version '9.04.01M7P08062020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) Apple
11 * Generated on web client 'https://odamid-usw2.oda.sas.com/SASStudio/ma
12 *
13 */
14
15 data WORK.HEART_CLEANED_Weight_Status;
16   length Weight_Status $ 11;
17   set MYDATA.HEART_WITH_MESSY_DATA;
18
19   select (Weight_Status);
20     when ('Under') Weight_Status='Underweight';
21     otherwise Weight_Status=Weight_Status;
22 end;
23 run;
    
```

Task: Recode (or Standardize) Values for Weight_Status, continued



Data Transformation Phase

Data transformation is the process of converting, cleaning, and structuring data from one format to a more usable format to enable processing and analysis tasks. Examples of data transformation techniques customarily performed by users include:

- Sorting.
- Data Deduplication.
- Data Aggregation.
- Splitting / Consolidating Data.
- Data Cleaning / Smoothing / Standardization.
- Data Normalization.
- Data Filtering.
- Data Concatenation.
- Combining / Integrating Data.

SAS Studio offers users several ways to transform data by producing DATA step and/or PROC SQL syntax. In this paper, we will illustrate the point-and-click data transformation techniques to combine two data sets (or tables) together. The following figure illustrates merge / join techniques using Venn diagrams, see below.

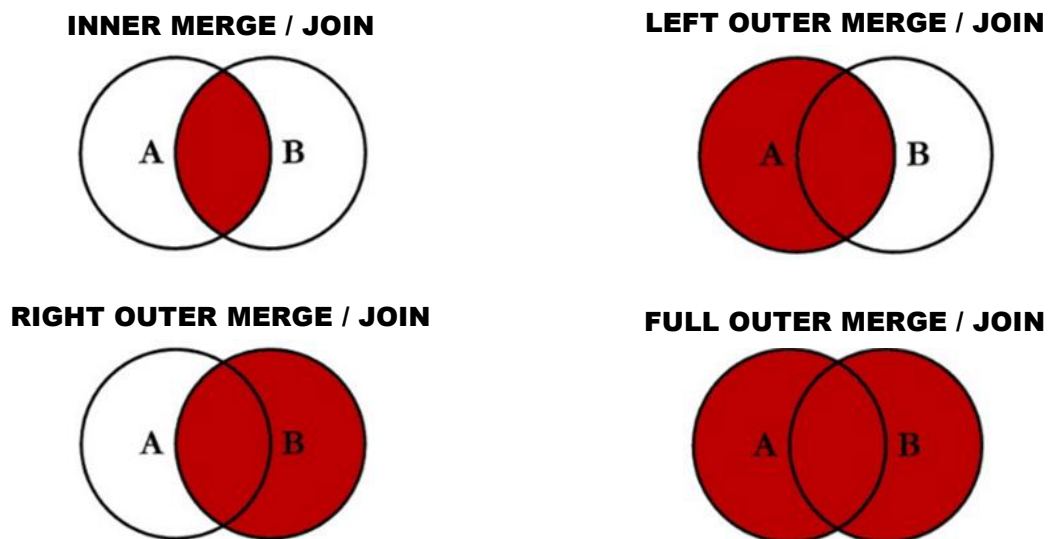
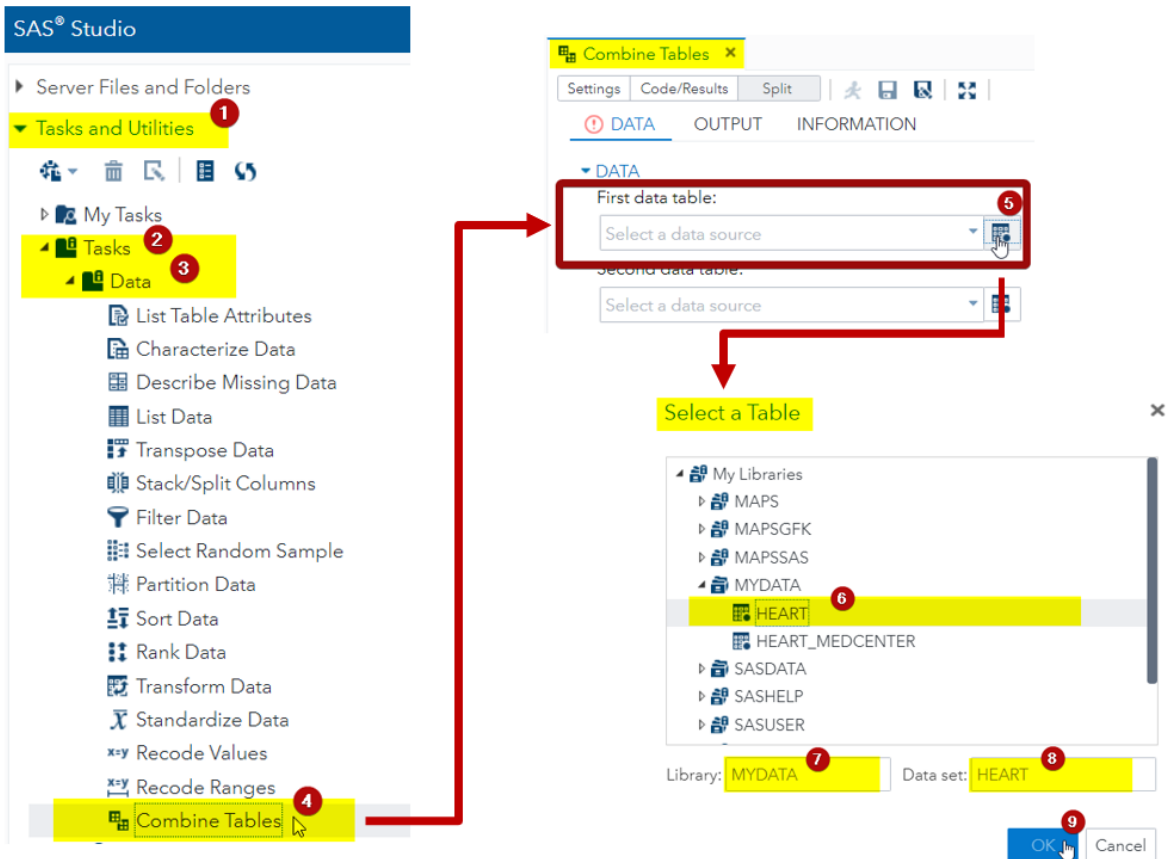


Illustration of Merges / Joins using Venn Diagrams

Task: Combining (Joining) Tables with All Matching and Non-Matching Rows

The Combining (or joining) tables with all matching and non-matching rows task allows users to perform a full outer join (or merge). SAS ODA and SAS Studio provide users with point-and-click capabilities to auto-generate DATA step or PROC SQL code to perform the operation.



Task: Combining (Joining) Tables with All Matching and Non-Matching Rows, continued

The image displays two screenshots of the SAS Studio 'Combine Tables' dialog box, illustrating the configuration for joining two tables. The first screenshot shows the 'DATA' tab where the first data table is 'MYDATA.HEART' and the second data table is 'Select a data source' (highlighted with a red box and number 10). A 'Select a Table' dialog box is open, showing a tree view of libraries. The 'MYDATA' library is selected (12), and the 'HEART_MEDCENTER' table is chosen (11). The 'Library' field is set to 'MYDATA' (12) and the 'Data set' field is set to 'HEART_MEDCENTER' (13). The 'OK' button is highlighted (14).

The second screenshot shows the 'METHOD' tab. The 'Select combination method' is set to 'Match merge (default)' (15). The 'Merge type' is set to 'All matching and nonmatching rows' (16). The 'Use PROC SQL' radio button is selected (17). The 'Required match columns' list is empty (18). A 'Columns' dialog box is open, showing a list of variables. The 'MedCtrID' variable is selected (19). The 'OK' button is highlighted (20).

Task: Combining (Joining) Tables with All Matching and Non-Matching Rows, continued

The screenshot shows the SAS Studio interface for the 'Combine Tables' task. On the left, the 'METHOD' section is expanded, showing the 'Match merge (default)' method selected. Under 'Merge type', 'All matching and nonmatching rows' is selected. The 'Use PROC SQL' radio button is checked. Under '*Required match columns:', 'MedCtrID' is listed. The main pane shows the generated SAS code, with lines 18-22 highlighted in yellow. A red circle with the number '21' is next to line 21.

```

1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '4/16/23, 1:55 AM'
6 * Generated by 'kirklafler'
7 * Generated on server 'ODAWS01-USW2.ODA.SAS.COM'
8 * Generated on SAS platform 'Linux LIN X64 3.10.0-1062.9.1.el7.x86_64'
9 * Generated on SAS version '9.04.01M7P08062020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
11 * Generated on web client 'https://odamid-usw2.oda.sas.com/SASStudio/main?locale=en_US
12 *
13 */
14
15 /* The DATA step and PROC SQL methods produce the same results when the match column */
16 /* values uniquely identify each row and the tables have no other columns with the same
17 /* Otherwise, the results may differ. */
18 proc sql noprint;
19     create table work.combine as select coalesce(a.MedCtrID, b.MedCtrID) as
20         MedCtrID, a.*, b.* from MYDATA.HEART as a full join MYDATA.HEART_MEDCENTER as
21         b on a.MedCtrID=b.MedCtrID;
22 quit;

```

This screenshot shows the same SAS code as above, but with the 'Edit' button highlighted in yellow. A red circle with the number '22' is next to the 'Edit' button. The code is identical to the previous screenshot.

```

1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '4/16/23, 1:55 AM'
6 * Generated by 'kirklafler'
7 * Generated on server 'ODAWS01-USW2.ODA.SAS.COM'
8 * Generated on SAS platform 'Linux LIN X64 3.10.0-1062.9.1.el7.x86_64'
9 * Generated on SAS version '9.04.01M7P08062020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (Kl
11 * Generated on web client 'https://odamid-usw2.oda.sas.com/SASStudio/main?locale=en_US&
12 *
13 */
14
15 /* The DATA step and PROC SQL methods produce the same results when the match column */
16 /* values uniquely identify each row and the tables have no other columns with the same
17 /* Otherwise, the results may differ. */
18 proc sql noprint;
19     create table work.combine as select coalesce(a.MedCtrID, b.MedCtrID) as
20         MedCtrID, a.*, b.* from MYDATA.HEART as a full join MYDATA.HEART_MEDCENTER as
21         b on a.MedCtrID=b.MedCtrID;
22 quit;

```

Task: Combining (Joining) Tables with All Matching and Non-Matching Rows, continued

```

CODE LOG RESULTS OUTPUT DATA
1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '4/16/23, 1:55 AM'
6 * Generated by 'kirklafler'
7 * Generated on server 'ODAWS01-USW2.ODA.SAS.COM'
8 * Generated on SAS platform 'Linux LIN X64 3.10.0-1062.9.1.e17.x86_64'
9 * Generated on SAS version '9.04.01M7P08062020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
11 * Generated on web client 'https://odamid-usw2.oda.sas.com/SASStudio/main?locale=en_US&zone=GMT-07
12 *
13 */
14
15 /* The DATA step and PROC SQL methods produce the same results when the match column */
16 /* values uniquely identify each row and the tables have no other columns with the same name. */
17 /* Otherwise, the results may differ. */
18
19 proc sql noprint ;
20 create table WORK.Heart_Heart_MedCenter as
21 select coalesce(a.MedCtrID, b.MedCtrID) as MedCtrID
22 , a.*
23 , b.*
24 from MYDATA.HEART as a
25 full join
26 MYDATA.HEART_MEDCENTER as b
27 on a.MedCtrID=b.MedCtrID ;
28 quit ;

```

```

CODE LOG RESULTS OUTPUT DATA
24
1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '4/16/23, 1:55 AM'
6 * Generated by 'kirklafler'
7 * Generated on server 'ODAWS01-USW2.ODA.SAS.COM'
8 * Generated on SAS platform 'Linux LIN X64 3.10.0-1062.9.1.e17.x86_64'
9 * Generated on SAS version '9.04.01M7P08062020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, lik
11 * Generated on web client 'https://odamid-usw2.oda.sas.com/SASStudio/main?locale=en_US&zone=GMT-
12 *
13 */
14
15 /* The DATA step and PROC SQL methods produce the same results when the match column */
16 /* values uniquely identify each row and the tables have no other columns with the same name. */
17 /* Otherwise, the results may differ. */
18
19 proc sql noprint ;
20 create table WORK.Heart_Heart_MedCenter as
21 select coalesce(a.MedCtrID, b.MedCtrID) as MedCtrID
22 , a.*
23 , b.*
24 from MYDATA.HEART as a
25 full join
26 MYDATA.HEART_MEDCENTER as b
27 on a.MedCtrID=b.MedCtrID ;
28 quit ;

```

Task: Combining (Joining) Tables with All Matching and Non-Matching Rows, continued

```

CODE | LOG | RESULTS | OUTPUT DATA
-----|-----|-----|-----
[Icons]
▼ Errors, Warnings, Notes 25
  ▶ ❌ Errors
  ▶ ⚠ Warnings (2)
  ▶ ⓘ Notes (3)
91
92     proc sql noprint ;
93         create table WORK.Heart_Heart_MedCenter as
94             select coalesce(a.MedCtrID, b.MedCtrID) as MedCtrID
95                 , a.*
96                 , b.*
97             from MYDATA.HEART as a
98                 full join
99                 MYDATA.HEART_MEDCENTER as b
100            on a.MedCtrID=b.MedCtrID ; 26
WARNING: Variable MedCtrID already exists on file WORK.HEART_HEART_MEDCENTER.
WARNING: Variable MedCtrID already exists on file WORK.HEART_HEART_MEDCENTER.
NOTE: Table WORK.HEART_HEART_MEDCENTER created, with 5210 rows and 22 columns.
101     quit ;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.01 seconds
      user cpu time       0.01 seconds
      system cpu time     0.01 seconds
      memory              16722.09k
      OS Memory           43000.00k
      Timestamp           04/16/2023 09:40:39 AM
    
```

Results – Snapshot Data Listing of WORK.HEART_HEART_MEDCENTER Dataset:

The resulting SAS dataset produced from the data transformation phase consists of 5,210 observations and 22 variables and is shown below.

MedCtrID	Status	DeathCause	AgeCHDdiag	Sex	AgeAtStart	Height	Weight	Diastolic	Systolic	MRW	Smoking	AgeAtDeath	Cholesterol	Chol_Status	BP_Status	Weight_Status	Smoking_Status	MedicalCenter	City	State	Zip
CA94105	Dead	Other		Female	29	62.50	140	78	124	121	0	55			Normal	Overweight	Non-smoker	San Francisco Medical Center	San Francisco	CA	94105
CA94105	Dead	Cancer		Female	41	59.75	194	92	144	183	0	57	181	Desirable	High	Overweight	Non-smoker	San Francisco Medical Center	San Francisco	CA	94105
CA94105	Alive			Female	57	62.25	132	90	170	114	10		250	High	High	Overweight	Moderate (6-15)	San Francisco Medical Center	San Francisco	CA	94105
CA90025	Alive				42	66.00	156	76	110	116	20		281	High	Optimal	Overweight	Heavy (16-25)	Los Angeles Medical Center	Los Angeles	CA	90025
CA94105	Alive			Female	36	64.75	136	80	112	110	15		196	Desirable	Normal	Overweight	Moderate (6-15)	San Francisco Medical Center	San Francisco	CA	94105
CA90025	Dead	Other		Male	53	65.50	130	80	114	99	0	77	276	High	Normal	Normal	Non-smoker	Los Angeles Medical Center	Los Angeles	CA	90025
CA90025	Dead	Cerebral Vascular Disease		Male	52	62.50	129	78	124	106	5	82	284	High	Normal	Normal	Light (1-5)	Los Angeles Medical Center	Los Angeles	CA	90025
NV89109	RIP			Male	39	66.25	179	76	128	133	30		225	Borderline	Normal	Overweight	Very Heavy (> 25)	Las Vegas Health Center	Las Vegas	NV	89109
CA90025	Alive			79 Male	57	67.25	165	76	128	118	15				Normal	Overweight	Moderate (6-15)	Los Angeles Medical Center	Los Angeles	CA	90025
NV89109	RIP			66 Male	44	69.00	155	90	130	105	30		292	High	Normal	Very Heavy (> 25)		Las Vegas Health Center	Las Vegas	NV	89109
CA94105	Alive			Female	37	64.50	134	76	120	108	10		196	Desirable	Normal	Normal	Moderate (6-15)	San Francisco Medical Center	San Francisco	CA	94105
NV89109	RIP			Male	40	66.25	151	72	132	112	30		192	Desirable	Normal	Overweight	Very Heavy (> 25)	Las Vegas Health Center	Las Vegas	NV	89109

Correlation Analysis using SAS Studio Point-And-Click Features

Process

1. Import your data into SAS Studio: First, you'll need to import the data you want to transform into SAS Studio. You can do this by clicking on "File" > "Import Data" and selecting the file you want to import.
2. Explore your data: Once you've imported your data, you can explore it by clicking on the data set in the "Explorer" pane. This will open the "Data" tab, where you can see the structure of your data, including the variable names, data types, and values.
3. Choose "Correlation Analysis": In the "Explorer" panel on the left, click on the "Tasks" tab. And then click on the "Correlation Analysis" under the tab of "Statistics".
4. Specify Variables: In the "Correlation Analysis" task window, you will typically find a section to specify the variables you want to analyze. You may have the option to choose variables from your dataset using dropdown menus or by typing in the variable names.
5. Set Correlation Method and Options: Depending on the options provided in the task window, you can choose the correlation method (e.g., Pearson, Spearman) and other options such as confidence intervals if available.
6. Interpret your results: After pressing the "Run" button, you will produce results including correlation coefficients, p-values, and other relevant statistics. You can typically view your results in various formats, such as tables and graphs. Explore the output to interpret the results of your correlation analysis.

Interpretation of Results

In SAS Studio's output, you will typically see a table that contains correlation coefficients. These coefficients measure the strength and direction of the relationship between pairs of variables.

The correlation coefficient can range from -1 to 1:

- A value of 1 indicates a perfect positive correlation, meaning that as one variable increases, the other increases proportionally.
- A value of -1 indicates a perfect negative correlation, meaning that as one variable increases, the other decreases proportionally.
- A value close to 0 suggests a weak or no linear relationship between the variables.
- Correlation does not imply causation. Even if two variables are strongly correlated, it does not mean that changes in one variable cause changes in the other.
- Outliers in your data can influence correlation coefficients. Examine your data for outliers and consider their impact on the analysis.

Example of Correlation Analysis and Interpretation:

In the following example, we chose Height, Diastolic, Smoking, Weight, Systolic, and AgeCHDdiag as our target analysis variables, and MRW, AgeAtDeath, AgeAtStart, and Cholesterol as the correlated variables. Therefore, we got total $6 * 4 = 24$ correlation coefficients to interpret.

According to the results, the evident and logical conclusion here is that there exists a robust and positive correlation between "AgeCHDdiag" and "AgeAtDeath." This relationship aligns with our intuitive expectations. A positive correlation suggests that individuals who are diagnosed with CHD at a later age ("AgeCHDdiag") tend to live longer ("AgeAtDeath"). This could be because those diagnosed could have a milder form of CHD or more effective treatments available to them. It may indicate that individuals who live longer are more likely to develop CHD at some point in their lives. This is consistent with the idea that as people age, they become more susceptible to age-related health conditions, including CHD.

The point-and-click steps are shown below.

The screenshot shows the SAS Studio interface with the following elements:

- Left Panel (Navigation):**
 - Tasks and Utilities (1)
 - My Tasks (2)
 - Statistics (3)
 - Correlation Analysis (4)
- Central Workspace:**
 - DATA:** _TEMP1.HEART_LARGE (5)
 - ROLES:**
 - Analysis variables: Height, Diastolic, Smoking, Weight, Systolic, AgeCHDdiag (6)
 - Correlate with: MRW, AgeAtDeath, AgeAtStart, Cholesterol (7)
- Right Panel (RESULTS):**
 - Table of Contents
 - 4 With Variables: MRW AgeAtDeath AgeAtStart Cholesterol
 - 6 Variables: Height Diastolic Smoking Weight Systolic AgeCHDdiag
 - Pearson Correlation Coefficients**

	Height	Diastolic	Smoking	Weight	Systolic	AgeCHDdiag
MRW	-0.13629 5199	0.38511 5203	-0.12524 5167	0.76717 5203	0.36257 5203	0.00942 1447
AgeAtDeath	-0.13650 1990	0.01094 1991	-0.28525 1971	0.00460 1988	0.19217 1991	0.74811 894
AgeAtStart	-0.13173 5203	0.27540 5209	-0.16743 5173	0.69352 5203	0.37938 5209	0.55091 1449
Cholesterol	-0.07959 5051	0.18336 5057	-0.01178 5049	0.67243 5051	0.19935 5057	0.00353 1408
 - Red text box: "Here is correlation coefficients between these variables. It ranges between -1 and 1, with +1 indicating a perfect positive correlation, -1 indicating a perfect negative correlation, and 0 indicating no line"

(Correlation Analysis)

Introduction to Regression Analysis

In this paper we will primarily focus on two types of regression analysis: Simple Linear Regression and Multiple Linear Regression. Regression analysis is a statistical method used for investigating the relationship between one or more independent variables and a dependent variable. The dependent variable is the object we are trying to predict, whereas the independent variables are the factors that might have an impact on the dependent variable. The primary objective is to model and quantify the relationship between the dependent and independent variables and sort out which of these variables have a significant impact. We will discuss them individually and provide examples using SAS Studio later.

1. **Simple Linear Regression:** Evaluates the relationship between a single independent variable and the dependent variable. It produces a linear equation of the form: $Y = \beta_0 + \beta_1 X_1 + \epsilon$ where:
 - Y is the dependent variable.
 - β_0 is the y-intercept.
 - β_1 is the slope, representing the change in Y for a one-unit change in X_1 .
 - X_1 is the independent variable.
 - ϵ represents the random error.
2. **Multiple Linear Regression:** Incorporates two or more independent variables. The equation becomes: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon$

Each β coefficient indicates the change in the dependent variable for a one-unit change in the corresponding independent variable, holding other variables constant.

In addition to the two key formulas illustrated above, we will include a few more concepts here which are the critical to understand for linear regression:

- **Coefficient:** Represents the change in the dependent variable for a one-unit change in an independent variable.
- **R-squared:** A statistical measure indicating the proportion of the variance in the dependent variable that's predictable from the independent variables. It ranges from 0 to 1, with higher values denoting a better fit.
- **p-value:** In the context of regression, it assesses the significance of each coefficient. A small p-value (typically ≤ 0.05) indicates that you can reject the null hypothesis, suggesting a meaningful addition of the variable to the model.

Assumptions of Simple Linear Regression

Simple linear regression makes certain assumptions about the data and are presented below:

1. **Linearity:**
The relationship between the independent variable X and the dependent variable of Y is linear. We can observe a straight line through the data points visually.
2. **Homoscedasticity:**
The variance of residual is the same for any value of the independent variable X . This means that the spread of the residuals should be roughly the same throughout the range of the independent variable.
3. **Independence:**
Observations in the data set are independent of each other, meaning there is no relationship among observations.
4. **Normality:**
The data is approximately normally distributed.

Example:

Go to Task and Utilities Tasks Graph Histogram.

Select the desired dataset under DATA and click the plus icon under ROLES. Select the variable from the dataset you wish to inspect for normality (we will select Weight here), then click the Run icon, and click RESULTS.

Assumptions of Binary Logistic Regression

Binary logistic regression makes certain assumptions about the data. The assumptions for binary logistic regression are:

1. No Multicollinearity:

Multicollinearity occurs when two or more explanatory variables are highly correlated to each other, such that they do not provide unique or independent information in the regression model. If the degree of correlation is high enough between variables, it can cause problems when fitting and interpreting the model.

2. Homoscedasticity:

The variance of the residual is the same for any value of the independent variable X. This means that the spread of the residuals should be roughly the same throughout the range of the independent variable.

3. Linearity:

The relationship between the independent variable X and the dependent variable of Y is linear. We can observe a straight line through the data points visually.

4. Large Sample Size:

A minimum of 10 cases with the least frequent outcome for each explanatory variable is required.

5. No Extreme Outliers:

The most common way to test for extreme outliers and influential observations in a dataset is to calculate Cook's distance for each observation. If there are indeed outliers, you can choose to (1) remove them, (2) replace them with a value like the mean or median, or (3) simply keep them in the model but make a note about this when reporting the regression results.^[1]

Simple Linear Regression using SAS OnDemand for Academics (ODA)

Background Knowledge

Simple linear regression is a type of data analysis method that attempts to find some relationships between two variables by fitting a linear equation ($y_i = \beta_0 + \beta_1 x_i + \epsilon$) to observed data. Among the two variables, one is the explanatory variable (independent variable), the other is response variable (dependent variable). The response variable is the focus of a question in a study or experiment. An explanatory variable is one that explains changes in the response variable, and it can be anything that might affect that variable.[1]

Typically, the primary purpose of simple linear regression is to create a linear model to predict how independent variables affect the dependent variable.[2] And this can be achieved by finding the regression line's y-intercept and slope in 2D coordinate plane, and the model's R (correlation coefficient) and R-square (coefficient of determination). Luckily, the SAS ODA can perform a simple linear regression for datasets and provide the wanted results for the users. Therefore, a step-by-step example is illustrated on how to use SAS ODA to perform a simple linear regression with the cleaned and transformed *HEART* dataset for analysis.

SAS ODA Tutorial

Before fitting the model, it is important to understand the dataset. Therefore, the first step of linear regression analysis is summary statistics. As shown below, the Summary Statistics Task can be found under Tasks and Utilities -> Tasks -> Statistics on the left side panel. After selecting Summary Statistics, you should select your dataset (step 2) and interested variables (step 3). After doing these, your codes for summary statistics are generated.

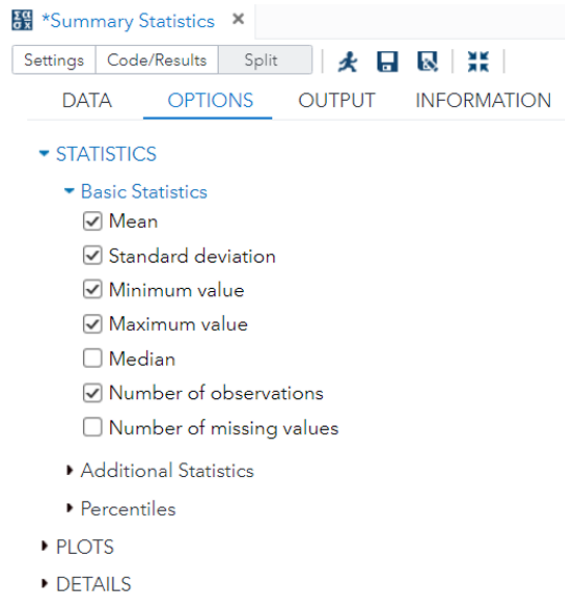
The screenshot displays the SAS Studio interface for the Summary Statistics task. On the left, the 'Tasks and Utilities' panel shows 'Summary Statistics' selected. The main workspace is divided into three panels: DATA, OPTIONS, and OUTPUT. The DATA panel shows the dataset 'MYDATA.HEART_HMC_JOINED' selected. The OPTIONS panel shows 'Analysis variables' selected, with a list of variables including AgeCHDdiag, AgeAtStart, Height, Weight, Diastolic, Systolic, and MRW. The OUTPUT panel shows the generated SAS code, which includes a PROC MEANS statement for the selected dataset and variables. Red boxes and numbers 1 through 4 highlight key steps: 1. Selecting 'Summary Statistics' in the left panel; 2. Selecting the dataset 'MYDATA.HEART_HMC_JOINED'; 3. Selecting the variables for analysis; 4. The generated SAS code block.

```

1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '9/16/23, 12:31 PM'
6 * Generated by 'u49996933'
7 * Generated on server 'ODAWS01-USW2.ODA.SAS.COM'
8 * Generated on SAS platform 'Linux X64 3.10.0-1062.9.1.e
9 * Generated on SAS version '9.04.01M7P08062020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64;
11 * Generated on web client 'https://odamid-usw2.oda.sas.com/S
12 *
13 */
14
15 ods noproctitle;
16 ods graphics / imagemap=on;
17
18 proc means data=MYDATA.HEART_HMC_JOINED chartype mean std min
19 var AgeCHDdiag AgeAtStart Height Weight Diastolic Systoli
20 AgeAtDeath Cholesterol;
21 run;
    
```

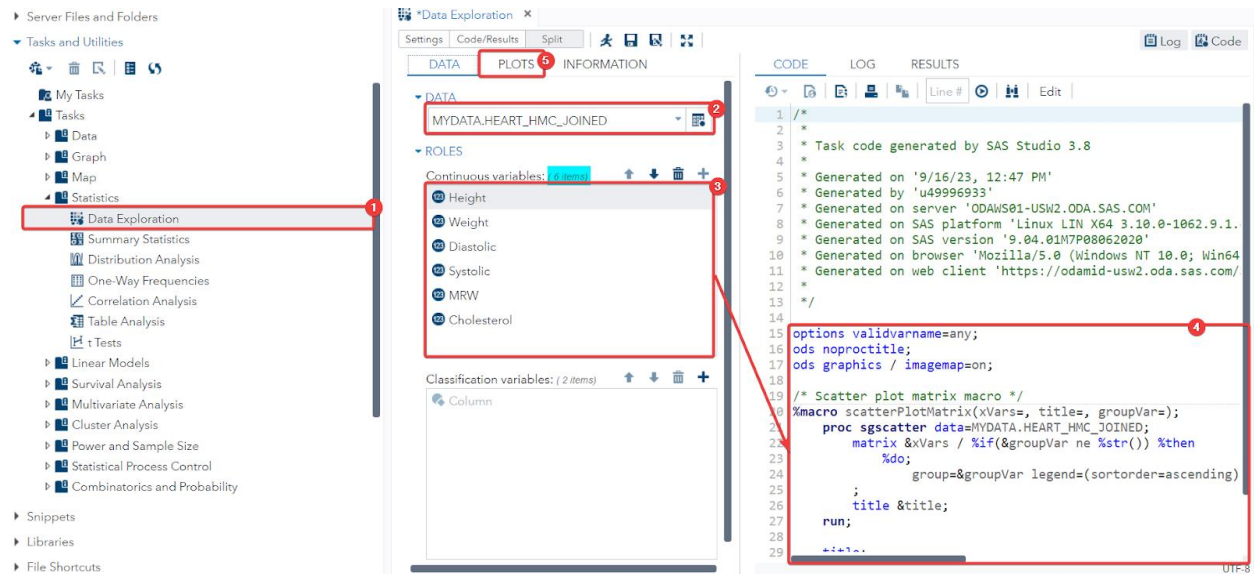
(Summary Statistics Task)

You can select what statistics to be presented under OPTIONS panel, as shown below.



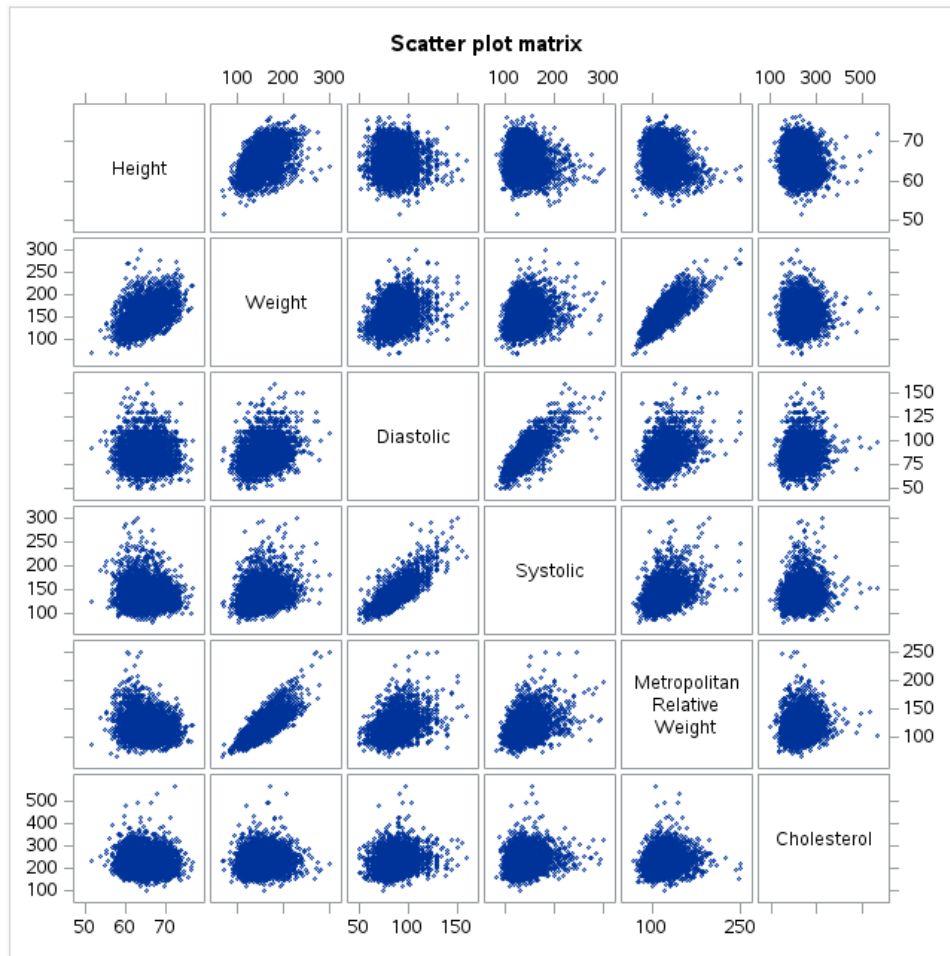
(Summary Statistics Options)

It is important to eliminate multicollinearity within independent variables. Using the Scatter Plot Matrix is a very intuitive way to discover correlations between variables. As shown below, the Scatter Plot Matrix can be created using the Data Exploration Task under Tasks and Utilities -> Tasks -> Statistics. You can select up to 6 continuous variables and 2 classification variables to create the plot. After selecting the variables, you can go to the plot panel to select the desired plot. The Scatter Plot Matrix has been selected by default.



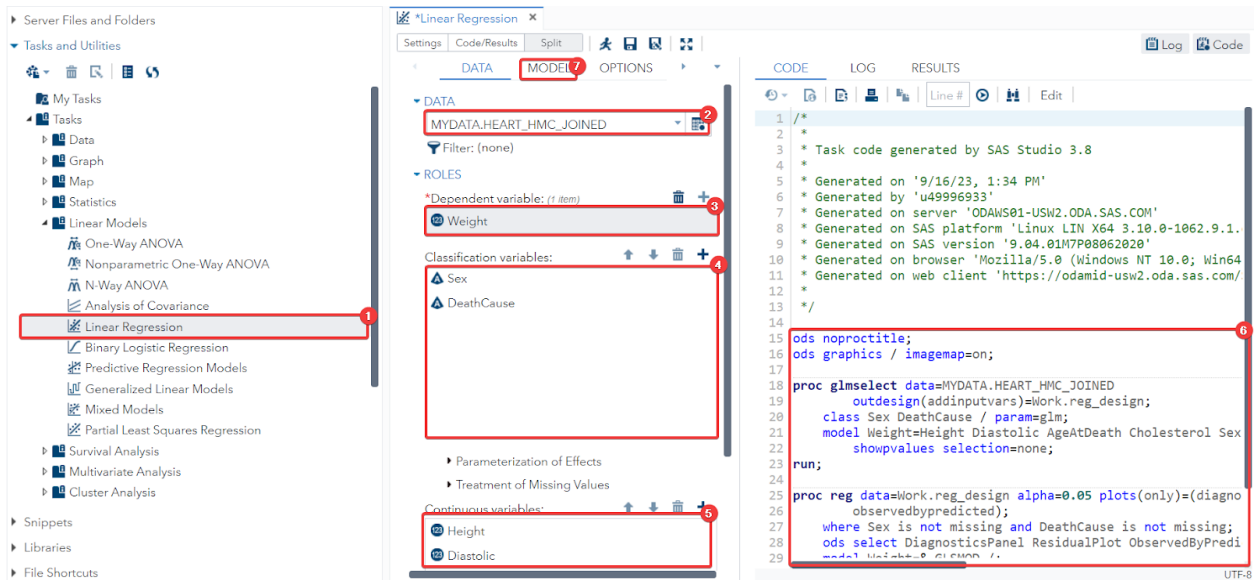
(Summary Statistics Options)

As shown below, Diastolic and Systolic are correlated, and Weight and Metropolitan Relative Weight are correlated. Therefore, Diastolic or Systolic and Weight or Metropolitan Relative Weight will not be fitted to the model.



(Scatter Plot Matrix for Autocorrelation)

The next step is to create the Linear Regression Model. The model can be created using Linear Regression Task under Tasks and Utilities -> Tasks -> Linear Models. As shown below, you need to select your dataset, dependent variable, and independent variables. In this scenario, variable Weight is selected as the dependent variable. For independent variables, it is important to put continuous variables under the continuous variable panel, even though it is allowed to be selected under the classification variable panel. If continuous variables are selected under the classification panel, the model will become unnecessarily complex. Since we have known that Weight and MRW, and Diastolic and Systolic are correlated, Systolic and MRW are not selected as predictive variables in this model.



(Linear Regression Task)

After selecting the variable, you should go to the MODEL panel to select their effect, as shown below.

Model Effects Builder



(Selecting Model Effects)

After clicking the run icon to fit the model, it is important to diagnose the model. The table of Parameter Estimates, shown below, which shows how much each variable affects the prediction. The p value in the last column shows the significance of each variable. As you can see, the variables Cholesterol and Death Cause are not significant to estimate the Weight. Therefore, the improved model should eliminate these two variables.

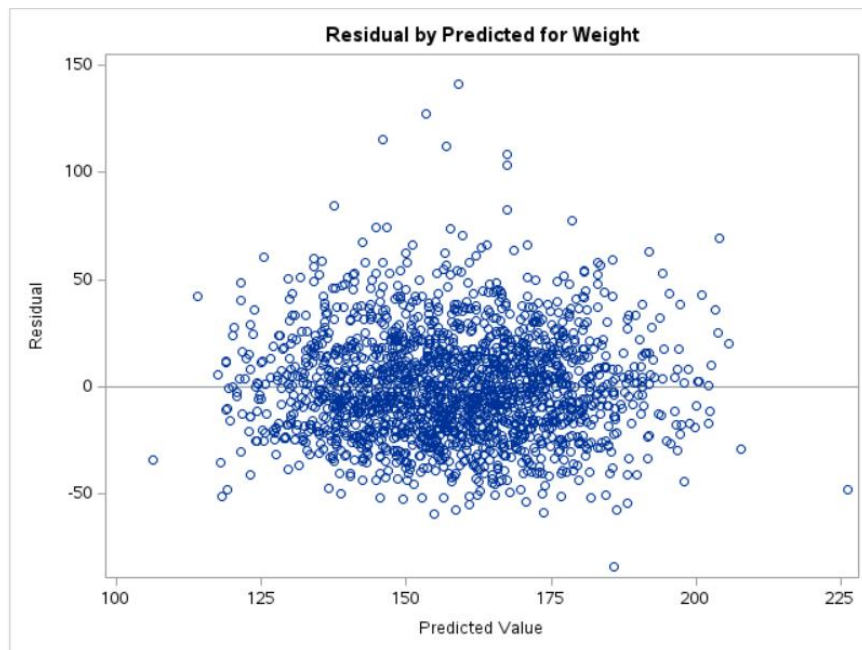
Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Pr > t
Intercept	1	-156.887283	16.452022	-9.54	<.0001
Height	1	3.743718	0.218737	17.12	<.0001
Diastolic	1	0.621934	0.039803	15.63	<.0001
AgeAtDeath	1	0.173634	0.054261	3.20	0.0014
Cholesterol	1	0.014488	0.012445	1.16	0.2445
Sex Female	1	-3.834785	1.568385	-2.45	0.0146
Sex Male	0	0	.	.	.
DeathCause Cancer	1	1.359924	2.605021	0.52	0.6017
DeathCause Cerebral Vascular Disease	1	2.406609	2.684368	0.90	0.3701
DeathCause Coronary Heart Disease	1	4.138062	2.576667	1.61	0.1084
DeathCause Other	1	-0.083558	2.702353	-0.03	0.9753
DeathCause Unknown	0	0	.	.	.

(Parameter Estimates)

The Residual vs Predicted Value plot is frequently used to test the assumptions of a linear model. Both assumption of linearity and assumption of homoscedasticity can be tested by a Residual vs Predicted Value plot.

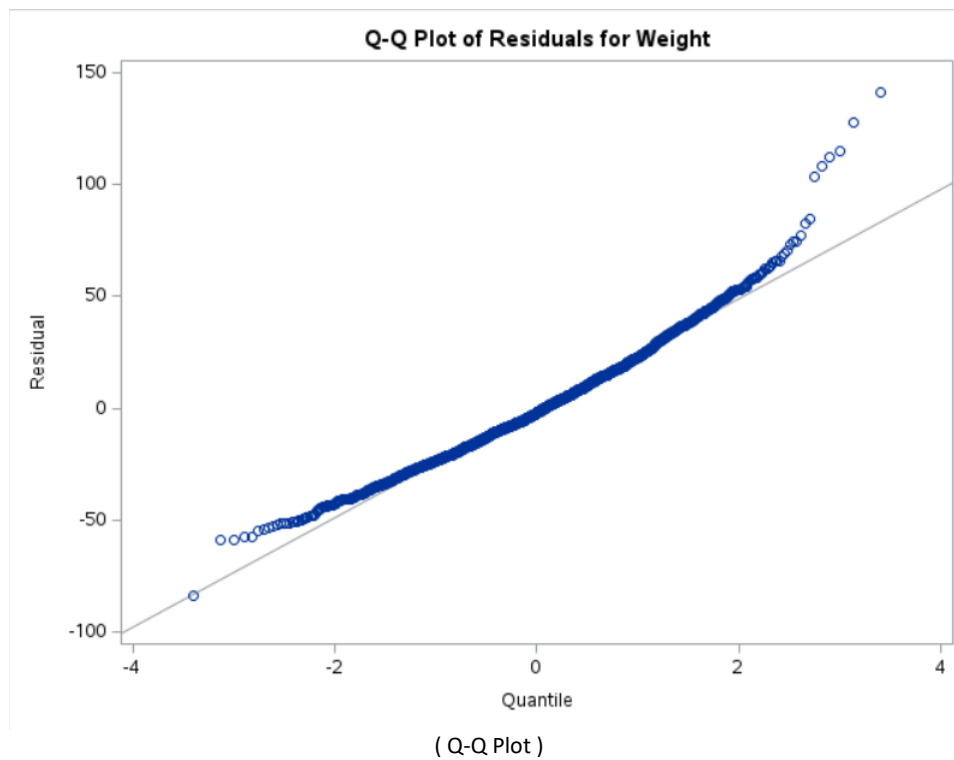
As shown below, the residuals make no pattern in the plot and bounce randomly around the 0 line. This implies that the assumption of linearity is valid.

The residuals do not change from left to right of the Residual vs Predicted Value plot and roughly form a "horizontal band" around the 0 line. This implies that the assumption of homoscedasticity is valid.



(Residual vs Predicted Value Plot)

The Q-Q plot, or quantile-quantile plot, can be used to test if the data fit to a theoretical distribution, such as normal distribution. We can use it to test if our model fits the normality assumption. As shown below, the points deviate significantly from the straight diagonal line and curve upward. This implies the assumption of normality is violated and the data slightly skewed to the left.



Result Interpretation

Some values in the produced table reveal the regression model's features and accuracy, and here we will try to explain their meanings. First, we can see the number of observations read and used. If these two numbers are different, it implies that some observations have unexpected data (like NAN) and it's better to do the data cleaning beforehand. Secondly, the p-value can be found in the "Analysis of Variance" table, the Pr > F. The p-value tells us how likely the linear relationship we observe in the sample also exists in the larger population. [3] If $p < \alpha$, there's enough evidence to assert the linear relationship exists in the larger population. Additionally, the estimated slope and intercept can be found in the Parameter Estimates table, and we can also add and subtract the standard error of it to get a corresponding confidence interval.

Model: MODEL1
Dependent Variable: Height

Number of Observations Read	5209
Number of Observations Used	5203
Number of Observations with Missing Values	6

Unexpected data?

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	1158.73675	1158.73675	91.85	<.0001
Error	5201	65613	12.61547		
Corrected Total	5202	66772			

P-value

Root MSE	3.55183	R-Square	0.0174
Dependent Mean	64.81318	Adj R-Sq	0.0172
Coeff Var	5.48010		

Parameter Estimates						
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	Intercept	1	67.23932	0.25789	260.73	<.0001
AgeAtStart	Age at Start	1	-0.05506	0.00574	-9.58	<.0001

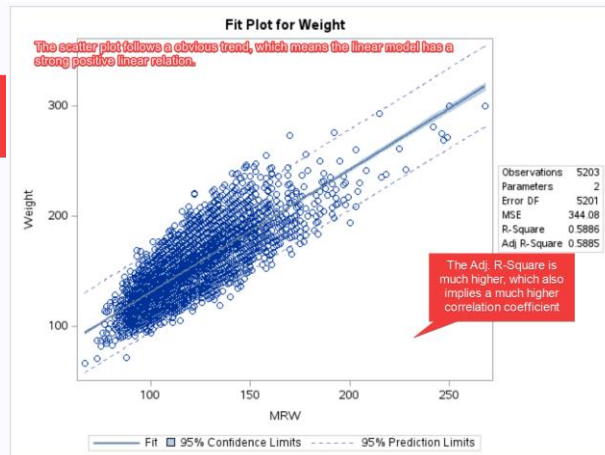
Intercept & slope

(Useful facts in produced tables)

If the default options are selected, only one plot will be outputted: the Fit Plot for the response (dependent) variable. We present useful information from the plot, and they are illustrated below:



(Fitness plots – Best-fit line doesn't fit)



(Best-fit line strongly fits)

To produce alternative plots for additional information, change the selection in the OPTIONS panel. We will explain those options' meaning here:

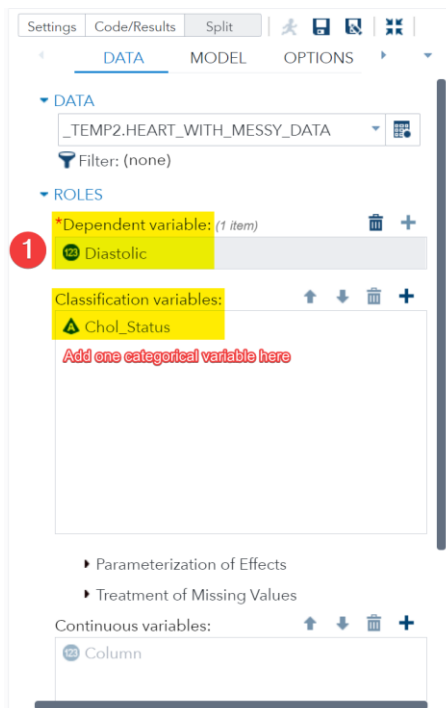
(In the OPTIONS panel, decide on the desired output)

There are 3 main aspects to select the produced results: Methods, Statistics, Plots. Methods contains only one choice: the confidence level, which is used to compare with the p-value and determines whether the linear regression model is significantly explainable. Although the default choice is 95% (as $\alpha=0.05$), we can select any confidence level we want by using “custom value”. There are many choices in the Statistics, like Type I or Type II squares, predicted values, collinearity analysis. All these options will output a statistical table here. However, producing too many or some too long tables will cause unreadable outputs. For instance, Selecting Diagnostics is strongly not recommended. This is because regression diagnostics in simple linear regression’s objective is to investigate if the calculated model and the assumptions we made about the data and the model, are consistent with the recorded data.[4] So, each observation (recorded data) will be analyzed individually with each of the model’s predicted value, and this will create a table with rows the same number of observations, which is unbelievably to read and get information from it.

In the Plots section, instead, the Diagnostic plots is a better choice to select to examine the distribution of difference between predicted value and recorded data, because it produces a residual plot instead of a super-long table. To visualize the linear model, there’s many choices in the “Scatter Plots”, and the most convenient default one is the “Fit plot for a single continuous variable”.

Categorical Variable Case Tutorial

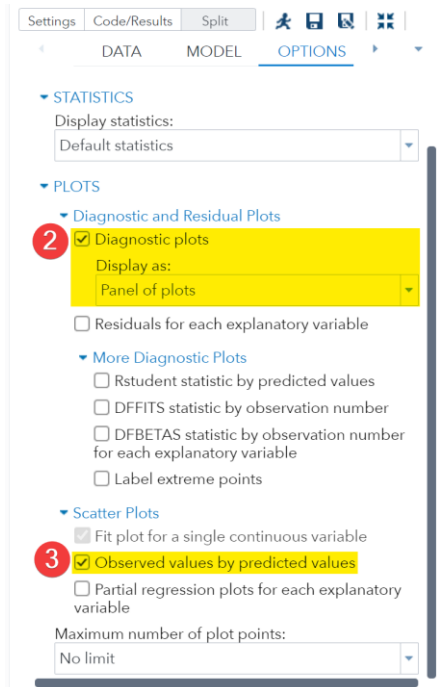
The only operation difference between using categorical variable and continuous variable as independent variable in simple linear regression is to add the variable to the correct panel, and all other steps are the same.



(Select variables from dataset)

Similarly, we can select intended tables and plots to produce. Because the independent variable is not continuous, it’s impossible to produce a fit plot, but an “Observed values by predicted values” is an alternative option. Also, Diagnostic plots are recommended to visualize residual distributions. (Note: Select “Display as Panel of plots”, or 9 separate plots will be produced.)

Looking at the results, the most important information to focus on is p-value and the Parameter Estimates. The categorical simple linear regression usually selects one level as starting point (in this case the “High”), and other level’s estimates are the changes of dependent variable’s value.



(OPTIONS panel, select the intended output)

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	26127	13063	79.91	<.0001
Error	5054	826254	163.48525		
Corrected Total	5056	852381			

Root MSE	12.78614
Dependent Mean	85.39213
R-Square	0.0307
Adj R-Sq	0.0303
AIC	30836
AICC	30836
SBC	25797

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Pr > t
Intercept	1	87.987158	0.302128	291.22	<.0001
Chol_Status Borderline	1	-2.705589	0.423237	-6.39	<.0001
Chol_Status Desirable	1	-5.756553	0.455677	-12.63	<.0001
Chol_Status High	0	0	.	.	.

(Useful facts in produced tables)

Multiple Linear Regression (MLR) using SAS OnDemand for Academics

Background Knowledge

Multiple Linear Regression (MLR) is a statistical analysis method that builds upon the foundation of Linear Regression (LR). It allows us to simultaneously consider the linear relationship between multiple independent variables and a dependent variable. Unlike Simple Linear Regression, which involves only one independent variable, Multiple Linear Regression can include multiple independent variables, providing a more accurate representation of complex real-world scenarios.

The key aspects associated with how Multiple Linear Regression extends the capabilities of Linear Regression:

- **Increased Number of Independent Variables:** In Simple Linear Regression, we have only one independent variable to predict the dependent variable. However, in Multiple Linear Regression, we can introduce multiple independent variables that may have an impact on the dependent variable. This allows us to comprehensively consider the influence of multiple factors.
- Like Simple Linear Regression, Multiple Linear Regression requires model evaluation to ensure its reliability. Evaluation methods include checking the distribution of residuals, using statistical tests to verify the significance of coefficients, and employing techniques such as cross-validation to assess the model's generalization performance.
- In summary, Multiple Linear Regression is an extended analysis tool that allows us to consider multiple independent variables when predicting and explaining the dependent variable. While it requires more data and analysis work than Simple Linear Regression, it provides richer insights and more accurate predictive capabilities.

Logistic Regression using SAS OnDemand for Academics

Background Knowledge

Logistic regression is used to estimate the probability of an event occurring or not occurring, based on a combination of inputting independent variables.[5] Any other regression methods we’ve shown before are not working when we want to estimate probability. This is because the probability is a numerical value between 0 and 1, but linear (simple and multiple) and non-linear regressions is one quantitative variable predicting another, which the predicted value can be beyond the interval [0, 1]. Also, the dependent variable is binary and thus does not have a normal distribution, which is a required assumption for linear regression. However, computing the inverse log odds converts all the predicted outputs into the interval [0, 1].

For all binary logistic regression, the dependent variable is in the Bernoulli distribution with an unknown probability p (of an event occurs). The most common link function is “Logit”:

$\text{Logit}(p) = \ln[p/(1-p)]$, and take its inverse function to become $\text{logit}^{-1}(x) = e^x/(1+e^x)$.

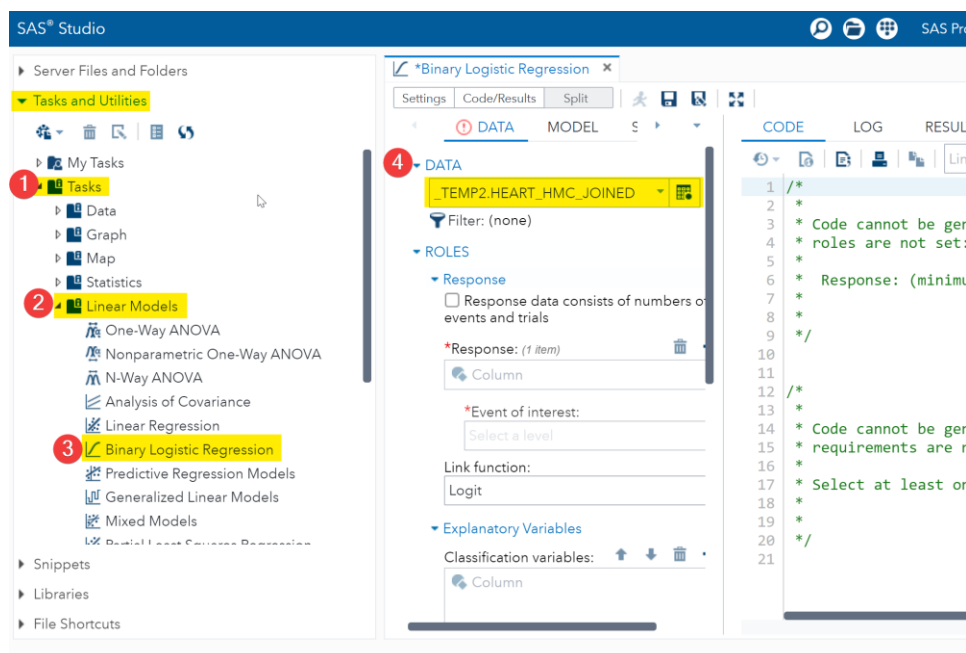
Finally, we can calculate the estimated probability by solving the equation between linear combination of independent variables and inverse logit function.

In addition to the estimated probability, the odds ratio is important to consider because it tells us two more stories: 1. The increase of a certain independent variable is increasing or decreasing the probability of the selected event happening; 2. How large each unit of a certain independent variable can change the probability of the selected event happening. To ensure a certain independent variable is significantly impacting the probability of the dependent variable, the confidence interval (usually at confidence level 0.05) should not contain 1.

The seemingly complex binary logistic regression can be performed using SAS OnDemand through a point-and-click process, and the content like odds ratio and estimated probability can be read from the results. Like other regression methods, a tutorial for Logistic Regression is presented.

SAS ODA Tutorial

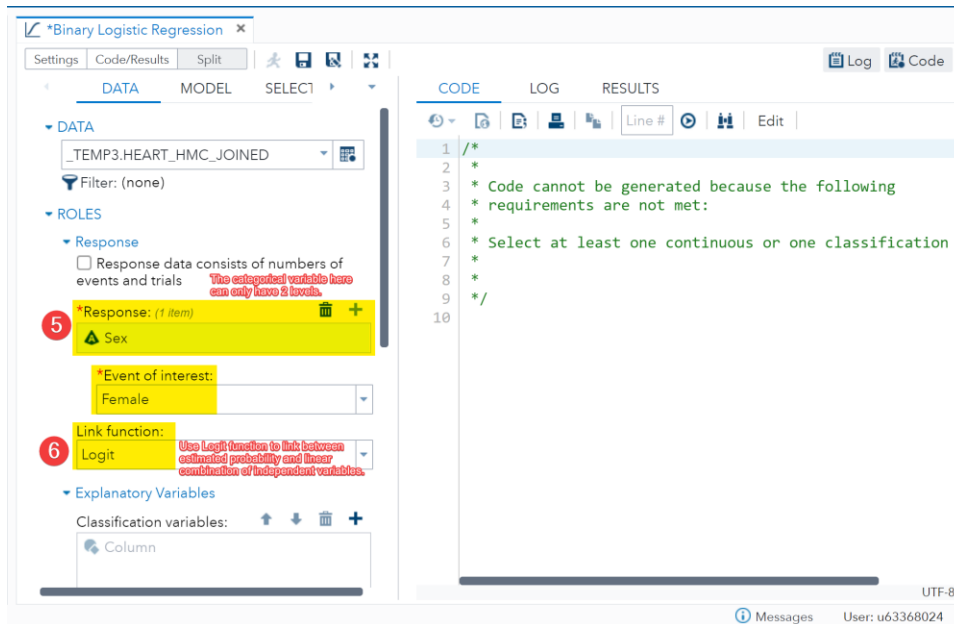
We can start a binary logistic regression by clicking the “Tasks and Utilities” in the left sidebar. Then, click “Tasks” → “Linear Models” → “Binary Logistic Regression”. Next, browsing and selecting the dataset, as *HEART_HMC_JOINED* in our example, in the DATA panel.



(Starting analysis and selecting datasets)

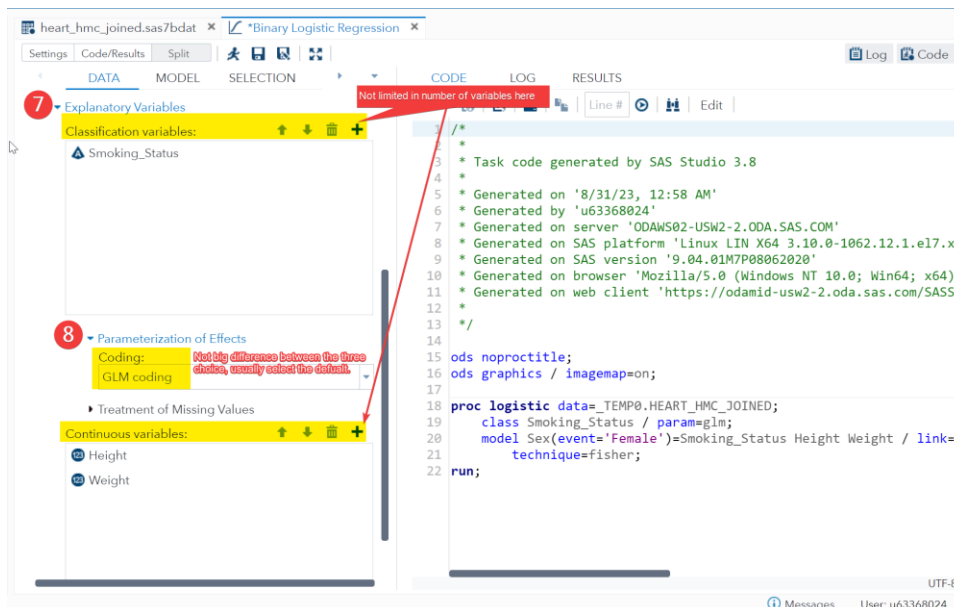
The next step is to select the response variable, the one we want to estimate the probability that one of its levels happen based on other variables. We can only choose categorical variables with 2 levels in this place by binary logistic regression’s definition. Click the plus sign on the upper right and choose the one we want in the generated list. After that, select the categorical level at “Event of interest”, then the model’s output is this level’s estimated occurrence probability.

Below the response variable panel, we need to choose the link function for our model, which is usually the “Logit” function that links the estimated probability and the linear combination of independent variables.



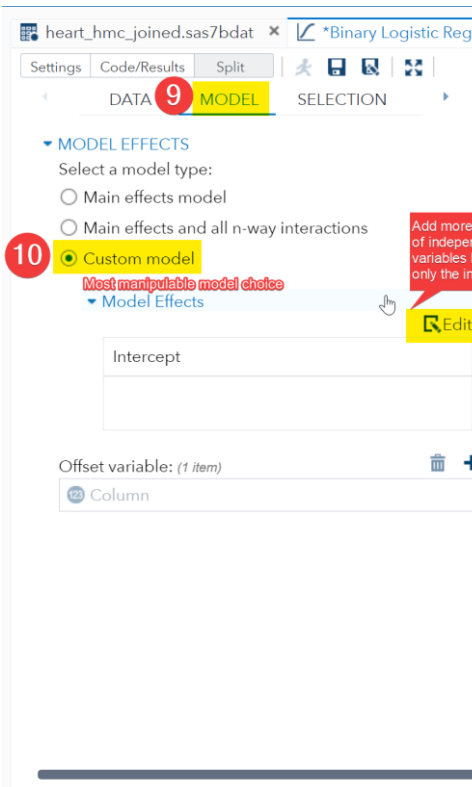
(Setting response variable and function)

As we see, there’s a place to select independent (explanatory) variables. Similarly, click the “plus sign” to add variables from the generated lists to either categorical or continuous sections. For the “Parameterization of Effects”, there’s no obvious difference between the 3 coding manners, so selecting an arbitrary one here is fine and the output will be logically the same.

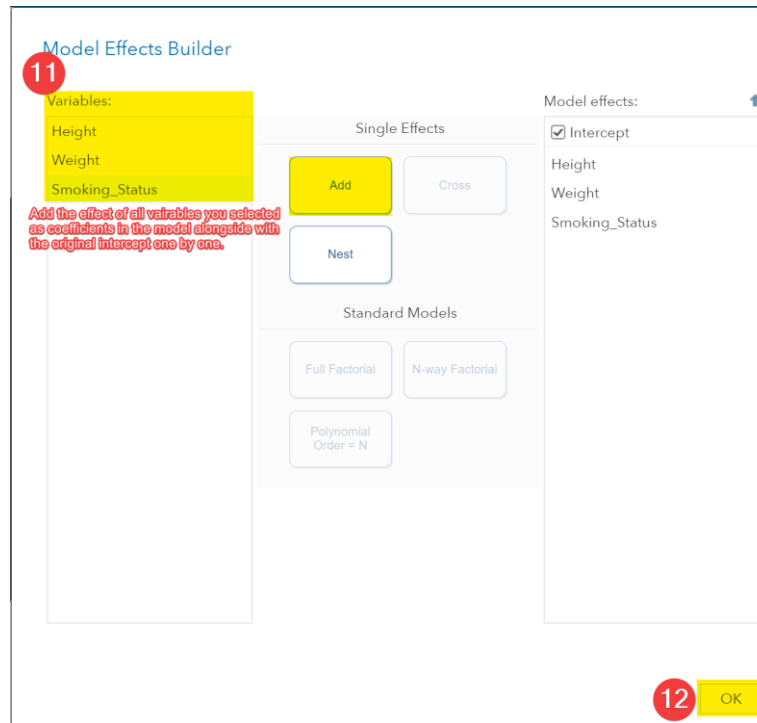


(Setting independent variables)

Next, click the MODEL panel, select “Custom model” in “MODEL EFFECTS” so that we have the maximum level of freedom to manipulate our model. Then, click the “Edit” button to add each independent variable as an effect into the model, and after the “Model Effects Builder” opens, click “the independent variable’s name” → “Add” → “Intercept” → “OK”.

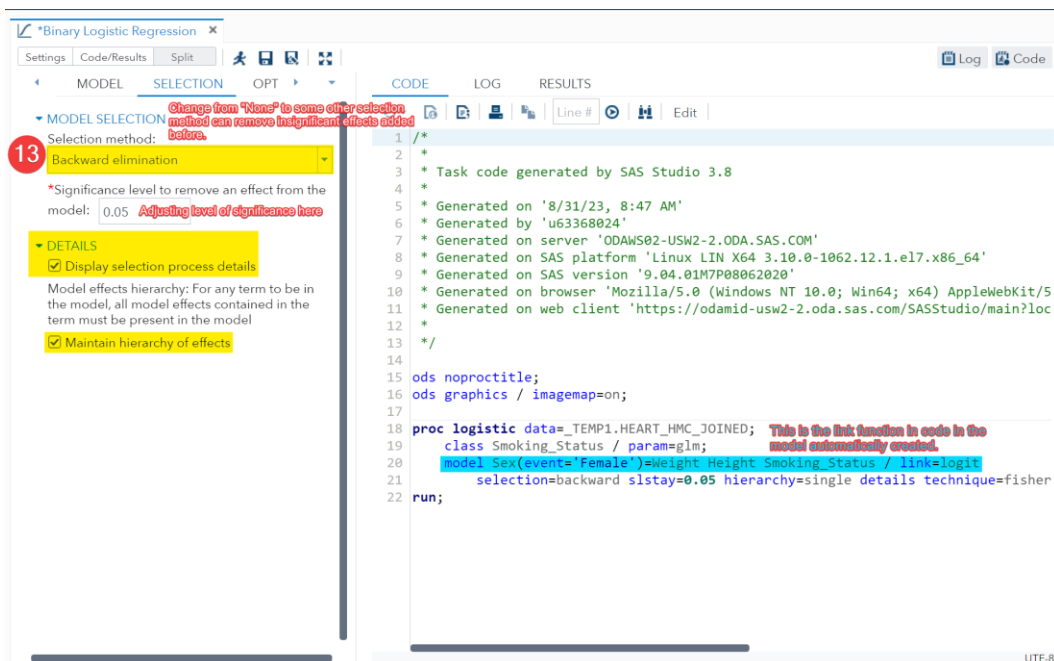


(Adjusting model effect)



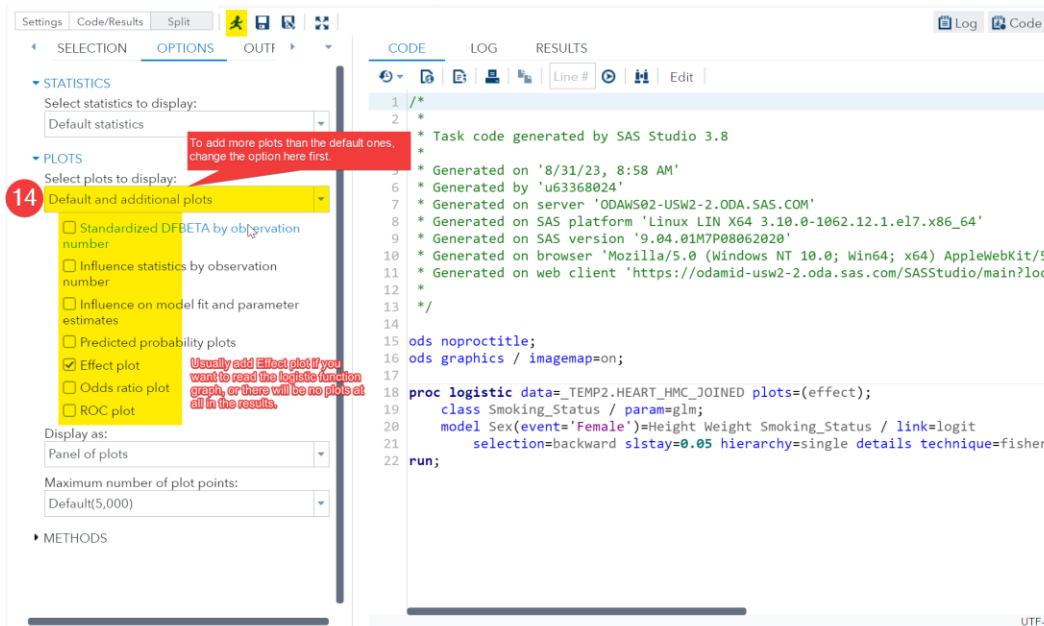
(Adding coefficients in effect)

Moving to the “SELECTION” panel, we can adjust the selection method and significance level here to remove the insignificant explanatory variables selected before if their p-value is higher than the significance level.[6] If we want to see how the variables are eliminated, just select all options in the “DETAILS” below.



(Effect selection)

Move to the “OPTIONS” panel after adjusting the selection and elimination method. Here, like “OPTIONS” in other regression analysis, you will need to select the type of tables and plots to see in the output results. The default plots of binary logistic regression contain nothing, so if you want to see the “S-shaped” function graph, which is recommended, select “Default and additional plots” is necessary.



(Results Options)

Result Interpretation

The most important value for a binary logistic regression is the odds ratio for each of the independent variables because they reveal whether or an independent variable influences the probability of the categorical level that we are interested and in what direction it influences. In this SAS Program, all the categorical explanatory variables will be cut into finer variables that each new variable represents one of their categories. For example, the “Smoking_Status” variable are separated into 5 new variables in the results with each responding to one of the smoking statuses.

Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	44.3742	1.6220	748.4611	<.0001
Smoking_Status	Heavy (16-25)	1	0.9500	0.1919	24.5147	<.0001
Smoking_Status	Light (1-5)	1	2.5431	0.2131	142.4376	<.0001
Smoking_Status	Moderate (6-15)	1	1.8761	0.2072	81.9980	<.0001
Smoking_Status	Non-smoker	1	1.7706	0.1963	81.3306	<.0001
Smoking_Status	Very Heavy (> 25)	0	0	.	.	.
Height		1	-0.6641	0.0254	681.4187	<.0001
Weight		1	-0.0181	0.00212	72.6481	<.0001

Check if the P-value smaller than the significance level you choose

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
Smoking_Status Heavy (16-25) vs Very Heavy (> 25)	2.586	1.775	3.766
Smoking_Status Light (1-5) vs Very Heavy (> 25)	12.719	8.377	19.312
Smoking_Status Moderate (6-15) vs Very Heavy (> 25)	6.528	4.350	9.798
Smoking_Status Non-smoker vs Very Heavy (> 25)	5.874	3.998	8.631
Height	0.515	0.490	0.541
Weight	0.982	0.978	0.986

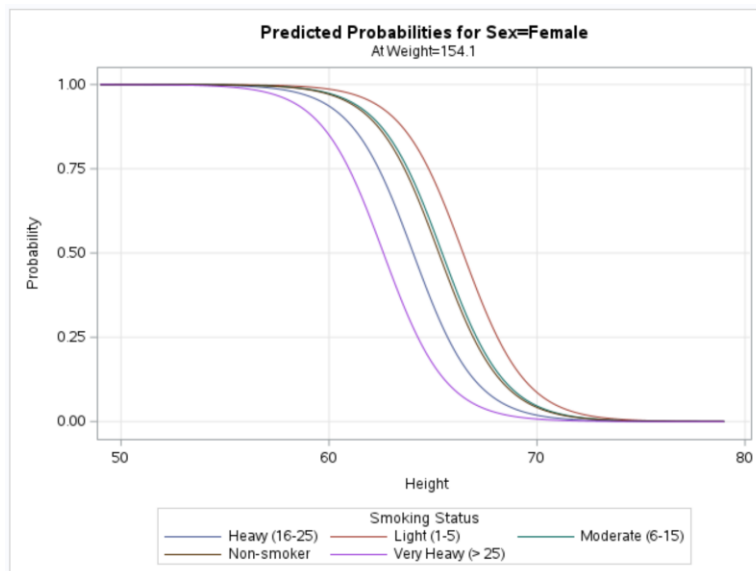
Check if confidence interval contains 1 here

(Interpreting results: odds ratio in the table)

How to interpret Odds Ratio? Looking at the Table “Odds Ratio Estimates”, the odds ratio for “Smoking_Status Light vs Very Heavy” is 12.719, which means “Light” being the smoking status will increase the odds ($p/1-p$) for the level of interest (in this case, being a female). Also, if the estimated odds ratio is smaller than 1, the odds for the level of interest will decrease. Thus, we can use odds ratio’s value to determine whether as independent variable increases or decreases the chance the level of interest occurs (for discrete independent variable, how its categories change that probability).

In addition, the confidence limit is also worthy to notice, since if the confidence limits have one greater than 1 and another smaller than 1, the confidence interval contains 1, which infers the variable could both increase and decrease the probability. So, that variable should be ignored at this significance level. An alternative way to check this is the p-value in the Table “Analysis of MLE” to see each variable separately whether their p-value is smaller than the chose significant level.

The effect plot is represented below. Unfortunately, the current SAS Studio can only output one single effect plot using the first categorical and the first continuous explanatory variable you selected. If we want to view the effect plots with another 2 variables’ impact, we must change the sequence of variable selected at the beginning and run the program again.



(Logistic regression’s effect plot considering only 2 variables)

Conclusion

With SAS® OnDemand for Academics (ODA) and SAS Studio, students, users, and SAS learners everywhere can learn SAS software’s many powerful features for performing data analysis and other important tasks. Our primary objective for this paper is to introduce the ease and simplicity of performing regression analysis by using point-and-click features found in SAS ODA and SAS Studio software. We demonstrated SAS Studio’s powerful point-and-click user interface to perform exploratory data analysis (EDA), data cleaning, data transformation, and regression analysis using the Navigation Pane consisting of Files and Folders, Tasks and Utilities, and Libraries. We also showed the SAS Program window consisting of Code, Log, and Results; data access to SAS (SAS7BDAT) datasets; predefined tasks and utilities to perform exploratory data analysis (EDA) to identify and better understand missing data, data anomalies, and data trends; and to conduct correlation and regression analysis so an organization can achieve greater insights, improve decision-making activities, and predict future events.

Reference

1. <https://www.statology.org/assumptions-of-logistic-regression/>
2. <https://www.statisticshowto.com/probability-and-statistics/types-of-variables/explanatory-variable/>
3. Answered by Lei Huaye, phd: <https://www.quora.com/What-is-the-primary-purpose-of-linear-regression-analysis>
4. <https://statisticsbyjim.com/regression/interpret-coefficients-p-values-regression/>

5. J. Ferré, in Comprehensive Chemometrics, 2009: <https://www.sciencedirect.com/topics/mathematics/regression-diagnostics#:~:text=Regression%20diagnostics%20is%20the%20part,consistent%20with%20the%20recorded%20data.>
6. What is logistic regression? <https://www.ibm.com/topics/logistic-regression#:~:text=Resources-What%20is%20logistic%20regression%3F,given%20dataset%20of%20independent%20variables.>
7. Our Learners Also Ask: <https://www.simplilearn.com/what-is-backward-elimination-technique-in-machine-learning-article#:~:text=Backward%20elimination%20is%20a%20method,is%20removed%20from%20the%20model.>

Acknowledgments

The authors thank the WUSS 2023 Conference Committee, particularly the Analytics & Statistics Section Chair, Kirby Sinclair, for accepting our abstract and paper; the WUSS 2023 Academic Chair, Lida Gharibvand, and the Operation Chair, Julie Kilburn, for organizing and supporting a great “live” conference event; SAS Institute Inc. for providing SAS users with wonderful software; and SAS users everywhere for being the nicest people anywhere!

Trademark Citations

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brands and product names are trademarks of their respective companies.

About the Authors

Kirk Paul Lafler is a consultant, developer, programmer, educator, and data scientist; and teaches SAS Programming and Data Management in the Statistics Department at San Diego State University. Kirk also provides project-based consulting and programming services to client organizations in a variety of industries including healthcare, life sciences, and business; and teaches “virtual” and “live” SAS, SQL, Python, Database Management Systems (DBMS) technologies (e.g., Oracle, SQL-Server, Teradata, MySQL, MongoDB, PostgreSQL, AWS), Excel, R, cloud-based technologies as well as other software and tools. Currently, Kirk serves as the Western Users of SAS Software (WUSS) Executive Committee (EC) Open-Source Advocate and Coordinator and is actively involved with several proprietary and open-source software, DBMS, machine learning, cloud-computing user groups and conference committees. Kirk is the author of several books including the popular PROC SQL: Beyond the Basics Using SAS, Third Edition (SAS Press, 2019), along with other technical books and publications. He is also an Invited speaker, educator, keynote, and leader; and is the recipient of 28 “Best” contributed paper, hands-on workshop (HOW), and poster awards.

Zheyuan “Walter” Yu is a skilled data analyst with a Master of Science degree in Biostatistics from the University of California, Davis. With expertise in data analysis tools like Excel, R, and SAS, he has honed his analytical abilities through his work in the dental supply industry.

Nuoer Lu is a thinker, innovator, and passionate about making a difference. Nuoer currently is pursuing a master’s degree at the University of North Carolina, Chapel Hill. Previously, Nuoer worked as a Supply Chain Systems Analyst at UC San Diego Health and has earned a bachelor’s degree in mathematics and economics at the University of California, San Diego.

Juncheng Yi is a Mathematics Undergraduate and Statistics Analytic Researcher at the University of Washington, Seattle. He has been involved in 3 statistical projects that explore machine learning and probability models (2 finished, 1 failed). His skills and interests include using Python (Programming Language), Machine Learning, Markov Chain Monte Carlo, Bootstrap, Data Collection and Visualization, conditional probability, Regression analysis.

Yanzhang “Gavin” Chen is an aspiring data analyst with innate ability to translate data into meaningful insights to aid data-driven decision making across the organization. Self-starter who is passionate about data exploration utilizing various programming languages and tools. Experience in analyzing broad concepts to distill into detailed recommendations.

Kai Kang graduated from UCLA with a degree in applied mathematics, a minor in statistics, and a specialty in computing. Currently, he is studying at UC Berkeley majoring in Master of Analytics. Kai has been involved with many projects to understand that data can facilitate our lives. And during his first internship at ByteDance, he engaged in the product development of a math solver app named Gauthmath by creating an annotation guidebook and optimizing the algorithms about solution steps on user feedback and improved user satisfaction by 10%.

Yixuan “Jason” Xiang is a Managerial Economics Undergraduate and Marketing Research Intern at the University of California, Davis. His skills and interests include using Python, Microsoft Office (Word, Excel, PowerPoint), Google Drive (Docs, Sheets, Slides), and he drafted 30+ reports dedicated to the stock market, analyzing candlestick charts to identify stocks with potential growth.

Zhaowen “Daniel” Qian, graduated with a bachelor’s degree in mechanical engineering from the University of Washington. He can write intro level code in multiple languages including Java, Python, MATLAB, SAS, and RStudio. He also has experience in writing research reports, video editing and 2D design, and is good at solving and communicating technical issues / content.

Swallow Xiaozhe Yan is President of US Education Without Borders. His organization reaches across cultures to foster and support the growth of students in their pursuit of international achievements. He is also the Chairman of the Presidential Youth Leadership Initiative in Des Moines, Iowa. He received a master’s degree in computer engineering and sociology from Iowa State University.

Comments and suggestions can be sent to:

Kirk Paul Lafler, sasNerd
Consultant, Developer, Programmer, Data Scientist, Educator, and Author
Specializing in SAS® / Python / SQL / Database Management Systems / Excel / R /
AWS / Cloud-based Technologies
E-mail: KirkLafler@cs.com
LinkedIn: <https://www.linkedin.com/in/KirkPaulLafler/>
Twitter: @sasNerd



Zheyuan “Walter” Yu
Data Analyst and Biostatistics Professional at Optimus Dental Supply in Grimes, Iowa
E-mail: zywyu@ucdavis.edu
LinkedIn: <https://www.linkedin.com/in/zheyuan-yu-25926b1b2/>



Nuoer “Norry” Lu
MS in Information Science at the University of North Carolina at Chapel Hill,
Chapel Hill, NC
E-mail: nuoerlu@gmail.com
LinkedIn: <https://www.linkedin.com/in/norry-lu/>



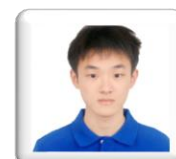
Juncheng Yi
University of Washington, Seattle
E-mail: jyi3@uw.edu
LinkedIn: <https://www.linkedin.com/in/nicklas-yee-b95a15210/>



Yanzhang “Gavin” Chen
University of California, Los Angeles
E-mail: gavinchenca@gmail.com
LinkedIn: <https://www.linkedin.com/in/gavin-chen-669815244/>



Kai Kang
Analytics Major at the University of California, Berkeley
E-mail: kangkai0518@berkeley.edu
LinkedIn: <https://www.linkedin.com/in/kaikang0518/>



Yixuan “Jason” Xiang
University of California, Davis
E-mail: yixiang@ucdavis.edu

LinkedIn: <https://www.linkedin.com/in/yixuan-xiang-05b9511b7/>



Zhaowen “Daniel” Qian
Daniel is a Mechanical Engineer for Ten Square International Inc. in West Des
Moines, Iowa

E-mail: danielqian98@gmail.com

LinkedIn: <https://www.linkedin.com/in/danielqianuw/>



Swallow Xiaozhe Yan
President, US Education Without Borders
E-mail: swallowxyan@yahoo.com

LinkedIn: <https://www.linkedin.com/in/swallow-xiaozhe-yan-588b0b8/>

