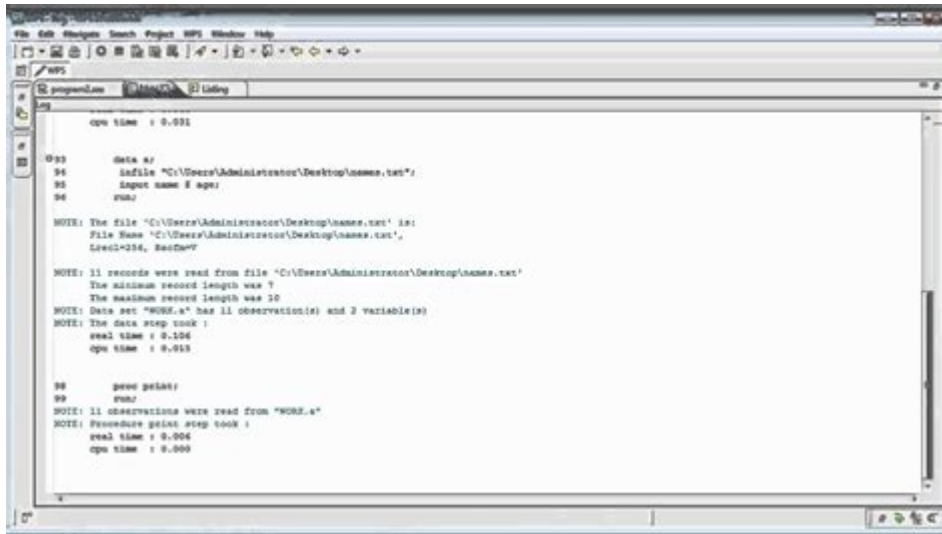


# Sas Programming Language Example



```

cpu time : 0.031

000      data a;
001      infile 'C:\Users\Administrator\Desktop\names.txt';
002      input name $ age;
003      run;

NOTE: The file 'C:\Users\Administrator\Desktop\names.txt' is:
File Name 'C:\Users\Administrator\Desktop\names.txt',
Length=256, Record=7

NOTE: 11 records were read from file 'C:\Users\Administrator\Desktop\names.txt'.
The minimum record length was 7
The maximum record length was 10
NOTE: Data set 'WORK.A' has 11 observation(s) and 2 variable(s)
NOTE: The data step took:
      real time : 0.106
      cpu time : 0.033

004      proc print;
005      run;

NOTE: 11 observations were read from 'WORK.A'
NOTE: Procedure print step took:
      real time : 0.006
      cpu time : 0.009

```

SAS programming language example: SAS, or Statistical Analysis System, is a software suite used for advanced analytics, business intelligence, data management, and predictive analytics. It is widely utilized in various industries for statistical analysis and data manipulation. This article will explore the fundamentals of SAS programming, providing examples and insights into its functionalities, syntax, and applications.

## Introduction to SAS Programming

SAS programming is a powerful tool for data analysis and reporting. The language is designed to handle, analyze, and visualize data effectively. It provides a comprehensive environment for data manipulation, statistical analysis, and graphical representation of data. Here, we will delve into the basic components of SAS programming that make it a go-to choice for data analysts and statisticians.

## Key Components of SAS

1. Data Steps: These are used for data manipulation. Data steps allow users to read, modify, and create datasets.

2. Procedures: Known as PROC steps, procedures are utilized to perform specific analyses or generate reports. Each procedure has a specific purpose, such as PROC PRINT for displaying data, PROC MEANS for statistical summaries, and PROC REG for regression analysis.
3. Macros: Macros in SAS help automate repetitive tasks, making code more efficient and easier to manage.
4. Formats and Informats: These are used to control how data is displayed and read into SAS. Formats change the appearance of data, while informats control how data is read into the program.

## Getting Started with SAS Programming

To begin programming in SAS, you first need to have access to the SAS software. Once installed, you can create a new program by opening a new window in the SAS interface. Below is a basic example to illustrate the structure of SAS code.

### Basic Structure of SAS Code

A typical SAS program consists of two main steps: the DATA step and the PROC step. Below is a simple example that demonstrates how to create a dataset and then print it.

```
``sas  
  
/ Create a dataset /  
  
data example_data;  
input Name $ Age Height Weight;  
datalines;  
  
John 25 175 70  
  
Jane 30 160 60  
  
Tom 22 180 80  
  
Lucy 28 165 55  
  
;
```

```
run;
```

```
/ Print the dataset /
```

```
proc print data=example_data;
```

```
title 'Example Data';
```

```
run;
```

```
***
```

Explanation of the Code:

#### 1. DATA Step:

- The ``data example_data;`` statement begins the creation of a new dataset named ``example_data``.
- The ``input`` statement specifies the variables to be included in the dataset. Here, ``Name``, ``Age``, ``Height``, and ``Weight`` are defined. The ``$`` symbol indicates that ``Name`` is a character variable.
- The ``datalines;`` statement allows you to input data directly into the program. Each line represents a new observation in the dataset.

#### 2. PROC Step:

- The ``proc print data=example_data;`` statement calls the PRINT procedure to display the dataset created in the DATA step.
- The ``title`` statement provides a title for the output.

## Data Manipulation in SAS

Data manipulation is a crucial aspect of data analysis. SAS provides various functions and techniques to manipulate and transform data efficiently.

# Common Data Manipulation Techniques

1. Subsetting Data: You can create a new dataset that contains only a subset of the original data based on specific conditions.

```
```sas
data young_adults;
set example_data;
if Age < 30;
run;
```
```

2. Creating New Variables: You can calculate new variables based on existing ones.

```
```sas
data example_data;
set example_data;
BMI = Weight / (Height2);
run;
```
```

3. Sorting Data: SAS allows you to sort your dataset based on one or more variables.

```
```sas
proc sort data=example_data;
by Age;
run;
```
```

4. Merging Datasets: You can combine two datasets based on a common variable.

```
```\nsas  
  
data merged_data;  
merge dataset1 dataset2;  
by common_variable;  
run;  
```\n
```

## Statistical Analysis with SAS

SAS is renowned for its statistical capabilities. The language offers a variety of procedures to perform different types of statistical analyses.

### Examples of Statistical Procedures

1. Descriptive Statistics: The PROC MEANS procedure provides summary statistics for numeric variables.

```
```\nsas  
  
proc means data=example_data;  
var Age Height Weight;  
run;  
```\n
```

2. Regression Analysis: The PROC REG procedure is used for linear regression analysis.

```
```\nsas  
  
proc reg data=example_data;  
model Weight = Height Age;  
run;  
```\n
```

```
```
```

3. ANOVA: The PROC ANOVA procedure is used to compare means across different groups.

```
```sas  
proc anova data=example_data;  
class Name; / Grouping variable /  
model Weight = Name;  
run;  
```
```

4. Frequency Analysis: The PROC FREQ procedure generates frequency tables for categorical variables.

```
```sas  
proc freq data=example_data;  
tables Name;  
run;  
```
```

## Data Visualization in SAS

Data visualization is an essential part of data analysis, helping to communicate findings effectively. SAS provides various options for creating visualizations.

### Creating Graphs in SAS

1. Basic Bar Chart: You can create a simple bar chart to visualize the frequency of a categorical variable.

```
```sas  
proc sgplot data=example_data;  
vbar Name / response=Weight stat=mean;  
run;  
```
```

2. Scatter Plot: A scatter plot can help visualize the relationship between two numeric variables.

```
```sas  
proc sgscatter data=example_data;  
plot WeightHeight;  
run;  
```
```

3. Box Plot: Box plots are useful for visualizing the distribution of a numeric variable across different categories.

```
```sas  
proc sgplot data=example_data;  
vbox Weight / category=Name;  
run;  
```
```

## Advanced SAS Programming Techniques

As you become more familiar with SAS programming, you may want to explore advanced techniques that can enhance your analyses.

# Using Macros for Automation

Macros can significantly reduce the amount of repetitive code you write. Below is an example of a simple macro that prints a dataset.

```
```sas  
  
%macro print_data(data);  
  
proc print data=&data;  
  
run;  
  
%mend print_data;  
  
  
%print_data(example_data);  
```
```

Explanation of the Macro:

- The ``%macro`` statement defines a macro named ``print\_data`` that takes one parameter, ``data``.
- The ``&data`` syntax inside the macro allows you to reference the dataset passed as an argument when calling the macro.
- The ``%mend`` statement marks the end of the macro definition.

## Conclusion

The SAS programming language example provided in this article illustrates the versatility and power of SAS for data analysis. From data manipulation and statistical analysis to data visualization and advanced programming techniques, SAS is a comprehensive tool for handling data. As you practice and explore more features within SAS, you will find it an invaluable asset for your data analysis tasks, whether in academia, healthcare, finance, or any other industry. With its robust capabilities and user-friendly interface, SAS continues to be a leader in the field of data analytics.



# Frequently Asked Questions

## What is SAS programming language used for?

SAS programming language is primarily used for data analytics, statistical analysis, data management, and predictive analytics in various industries.

## Can you provide a simple example of a SAS program?

Sure! A basic example of a SAS program to read a dataset and print it would look like this:

```
```\nsas\n\nDATA mydata;\nINPUT name $ age;\n\nDATALINES;\n\nJohn 25\nJane 30\n\n;\n\nRUN;\n\n\nPROC PRINT DATA=mydata;\n\nRUN;\n\n```\n
```

## What are some common procedures in SAS programming?

Common procedures in SAS include PROC PRINT for displaying data, PROC MEANS for summary statistics, PROC FREQ for frequency tables, and PROC REG for regression analysis.

## How does data input work in SAS?

Data input in SAS can be done using the DATA step with the INPUT statement for raw data, or using PROC IMPORT for importing data from external files like CSV or Excel.



如何 安装python和R语言 - SAS SPSS Stata ...

python和SAS stata 99.99% 安装python和SAS python ...

SAS R Python matlab spss stata 安装 ...

Jun 27, 2015 · SAS SPSS R 安装 SAS SPSS R 安装 MATLAB 安装 STATA 安装 SAS 安装 ...

如何 安装python和R语言 - 如何

如何 安装python和R语言 www.behance.net (Sankey Diagram) 如何 安装python和R语言 ...

SAS 安装 - 如何

99.99% SAS 安装 ...

SaaS 安装 SAAS 安装 - 如何

如何 SaaS 安装 SAAS 安装 SaaS 安装 SaaS 安装 ...

NVMe SAS SATA 安装 ...

NVMe SAS SATA 安装 NVMe SAS SATA 安装 ... 1. NVMe Non-Volatile Memory Express ...

如何 安装SAS? - 如何

如何 安装SAS 如何 安装SAS 如何 安装SAS 1. 如何 安装SAS 如何 安装SAS ...

sas和sata 安装 - 如何

1. 如何 安装SAS 如何 安装SAS 2. 如何 安装SAS 10000 SAS 7200 5400 3. 如何 安装SAS ...

如何 安装SSD/SATA/SAS 安装 - 如何

如何 安装600GB SAS SATA 如何 安装SSD ...

如何 安装sas 安装 - 如何

2019 9 SAS 安装 SAS 安装 ...

Explore practical SAS programming language examples that enhance your data analysis skills. Discover how to implement key features effectively. Learn more!

[Back to Home](#)