Working in SAS Cloud involves using SAS software hosted in the cloud, such as SAS Studio or SAS Viya. These platforms allow you to run SAS code, manage data, and perform analytics without needing to install SAS on your local machine.

1. Accessing SAS Cloud

1. **Sign Up or Log In**:
   - If you don't already have access, you may need to sign up for a SAS Cloud service like SAS OnDemand for Academics (for students and educators) or a commercial SAS Viya instance.
   - Log in to your SAS Cloud account.
2. **Launching SAS Studio**:
   - Once logged in, you will typically be directed to a dashboard or a landing page.
   - Launch SAS Studio, which is an integrated development environment (IDE) for writing and running SAS code.

2. Getting Started with SAS Studio in SAS Cloud

1. **Workspace Overview**:
   - **Code Editor**: This is where you write and execute your SAS code.
   - **Explorer**: On the left, you'll find the Explorer, which shows the file system, libraries, and datasets available to you.
   - **Log**: Displays logs from your code executions, including errors, warnings, and other messages.
   - **Results**: Shows the output of your SAS programs, including tables, graphs, and other reports.
2. **Setting Up Libraries**:
   - Libraries in SAS are references to data sources. To create a new library, use the `LIBNAME` statement:

   ```
   libname mylib '/path/to/your/data';
   ```

   - In SAS Cloud, paths may refer to cloud storage locations, or you can use pre-configured libraries.
3. **Uploading Data**:
   - **Via Drag and Drop**: You can upload files (e.g., CSV, Excel) directly into SAS Cloud by dragging them into the Explorer pane.
   - **Using Code**: You can also load data using `PROC IMPORT`:

   ```
   proc import datafile='/path/to/your/file.csv'
       out=work.mydata
       dbms=csv
       replace;
   run;
   ```

4. **Writing and Running Code**:

- Write your SAS code in the Code Editor.
- To run your code, click the "Run" button or press `F3`. The output will appear in the Results tab, and any messages will be in the Log tab.

3. Performing Common Tasks

1. **Data Management**:
   - Use `DATA` steps to create and manipulate datasets:

```
data work.newdata;
    set work.mydata;
    /* Add your data transformations here */
run;
```

2. **Data Analysis**:
   - Use `PROC` steps to perform analyses, such as correlation:

```
proc corr data=work.mydata;
    var var1 var2 var3;
run;
```

3. **Creating Visualizations**:
   - Use `PROC SGPLOT` or other `PROC` steps to create visualizations:

```
proc sgplot data=work.mydata;
    scatter x=var1 y=var2;
run;
```

4. Saving and Sharing Your Work

1. **Saving Code**:
   - Save your SAS programs by clicking the "Save" button in the Code Editor. You can organize your code into folders in the Explorer.
2. **Exporting Results**:
   - You can export datasets, tables, or graphs generated by your code by right-clicking the output in the Results tab and selecting export options.
3. **Collaborating**:
   - SAS Cloud often supports collaboration features, allowing you to share projects, datasets, or code with other users.

5. Learning Resources

1. **Documentation**:
   - SAS provides comprehensive documentation and tutorials on their website.
2. **SAS Community**:
   - Engage with the SAS community through forums, webinars, and user groups for tips and best practices.
3. **Courses and Tutorials**:

- Take advantage of free and paid SAS courses available through SAS Academy or other educational platforms.

Summary

SAS Cloud allows you to perform all your SAS programming tasks in a web-based environment, making it accessible from any location with internet access. By understanding the basics of navigating SAS Studio, managing data, and running code, you can efficiently carry out your analytics projects in SAS Cloud

To perform correlation analysis in SAS Cloud (using SAS Studio or similar SAS cloud environments), you can use the `PROC CORR` procedure. This procedure computes the Pearson correlation coefficients between variables. Below is an example of how to write code for correlation analysis in SAS:

Example: Correlation between Multiple Variables

Suppose we have a dataset named `work.mydata` with the variables `var1`, `var2`, and `var3`. Here's how you can calculate the correlations between these variables:

```
/* Calculate correlations between variables */
proc corr data=work.mydata;
   var var1 var2 var3;
run;
```

Explanation:

- **`proc corr data=work.mydata;`**: This line starts the `PROC CORR` procedure, specifying the dataset `work.mydata` as the input.
- **`var var1 var2 var3;`**: This line specifies the variables for which you want to calculate the correlation coefficients.
- **`run;`**: This ends the procedure and executes the code.

Output:

- The output will include a correlation matrix, which shows the Pearson correlation coefficients for each pair of variables.
- It will also include other statistics such as the number of observations and p-values to test the hypothesis that the correlation is zero.

Example: Saving the Correlation Matrix to a Dataset

If we want to save the correlation matrix to a dataset for further analysis, you can use the `outp=` option:

```
/* Save the correlation matrix to a dataset */
proc corr data=work.mydata outp=work.corr_matrix;
   var var1 var2 var3;
run;
```

Explanation:

- **`outp=work.corr_matrix;`**: This option saves the correlation matrix to a new dataset called `corr_matrix` in the `work` library.

This code can be executed in SAS Cloud environments like SAS Studio. The output dataset `corr_matrix` can be used for further analysis, visualization, or reporting.

Example Problem: Analyzing the Correlation Between Variables

Problem Statement: A researcher has collected data on the height (in cm), weight (in kg), and age (in years) of 10 individuals. The researcher wants to determine if there is a correlation between these variables.

Dataset: The dataset contains the following variables:

- `Height`: The height of the individuals in centimeters.
- `Weight`: The weight of the individuals in kilograms.
- `Age`: The age of the individuals in years.

Here is the sample data:

Subject	Height (cm)	Weight (kg)	Age (years)
1	170	65	25
2	160	58	22
3	175	75	30
4	180	85	28
5	165	62	24
6	155	55	21
7	168	70	26
8	172	68	27
9	158	60	23
10	178	80	29

Step 1: Input the Data

First, input the data into a SAS dataset:

```
/* Create a dataset named 'people_data' */
```

```
data work.people_data;
   input Subject Height Weight Age;
   datalines;
1 170 65 25
2 160 58 22
3 175 75 30
4 180 85 28
5 165 62 24
6 155 55 21
7 168 70 26
8 172 68 27
9 158 60 23
10 178 80 29
;
run;
```

Step 2: Perform Correlation Analysis

Use the `PROC CORR` procedure to compute the correlation coefficients between `Height`, `Weight`, and `Age`:

```
/* Calculate correlations between Height, Weight, and Age */
proc corr data=work.people_data;
   var Height Weight Age;
run;
```

Example Output (Hypothetical):

Assume the output shows the following correlations:

- **Height and Weight**: 0.95 (strong positive correlation)
- **Height and Age**: 0.93 (strong positive correlation)
- **Weight and Age**: 0.90 (strong positive correlation)

These results suggest that there is a strong relationship between an individual's height and weight, and a moderate relationship between height and age, as well as weight and age.

Example Problem: Correlation Analysis of Test Scores

Problem Statement: A teacher wants to understand the relationship between students' scores in three different subjects: Mathematics, Science, and English. The teacher has data for 10 students and wants to determine if there is a significant correlation between the scores in these subjects.

Dataset: The dataset contains the following variables:

- `Math`: Scores in Mathematics (out of 100).
- `Science`: Scores in Science (out of 100).
- `English`: Scores in English (out of 100).

Here is the sample data:

Student	Math	Science	English
1	85	78	92
2	90	88	95
3	76	85	80
4	92	91	89
5	70	75	78
6	88	82	85
7	95	94	96
8	65	70	72
9	78	80	83
10	82	85	87

Step 1: Input the Data

First, input the data into a SAS dataset:

```
/* Create a dataset named 'scores' */
data work.scores;
   input Student Math Science English;
   datalines;
1 85 78 92
2 90 88 95
3 76 85 80
4 92 91 89
5 70 75 78
6 88 82 85
7 95 94 96
8 65 70 72
9 78 80 83
10 82 85 87
;
run;
```

Step 2: Perform Correlation Analysis

Use the `PROC CORR` procedure to calculate the correlation coefficients between `Math`, `Science`, and `English` scores:

```
/* Calculate correlations between Math, Science, and English scores */
proc corr data=work.scores;
   var Math Science English;
run;
```

Output Interpretation:

The output will include a correlation matrix showing the Pearson correlation coefficients between `Math`, `Science`, and `English` scores. Additionally, it will provide p-values to test the significance of these correlations.

- **Math vs. Science**: A positive correlation indicates that students who score well in Math tend to score well in Science.
- **Math vs. English**: A positive correlation suggests that students who perform well in Math also perform well in English.
- **Science vs. English**: A positive correlation indicates a relationship between Science and English scores.

Hypothetical Output Interpretation:

Assume the output shows the following correlations:

- **Math and Science**: 0.86 (strong positive correlation)
- **Math and English**: 0.92 (strong positive correlation)
- **Science and English**: 0.78 (moderate positive correlation)

These results suggest that there is a strong relationship between Math and English, as well as Math and Science scores, and a moderate relationship between Science and English scores.

Step 3: Optional - Save Correlation Matrix to a Dataset

If you want to save the correlation matrix for further analysis, you can modify the code as follows:

```
/* Save the correlation matrix to a dataset */
proc corr data=work.scores outp=work.corr_matrix;
   var Math Science English;
run;
```

- **`outp=work.corr_matrix;`**: This saves the correlation matrix to a dataset named `corr_matrix` in the `work` library.

Example Problem: Correlation Between Advertising and Sales

Problem Statement: A company wants to understand the relationship between its advertising expenditure in three different media channels (TV, Radio, and Newspaper) and the resulting sales figures. The company has data from 10 different campaigns. The goal is to determine if there is a significant correlation between the amount spent on each type of advertising and the sales generated.

Dataset: The dataset contains the following variables:

- `TV_Ad`: Advertising expenditure on TV (in thousand dollars).

- `Radio_Ad`: Advertising expenditure on Radio (in thousand dollars).
- `Newspaper_Ad`: Advertising expenditure on Newspapers (in thousand dollars).
- `Sales`: Sales generated (in thousand units).

Here is the sample data:

Campaign	TV_Ad	Radio_Ad	Newspaper_Ad	Sales
1	230	37	69	22.1
2	44	39	45	10.4
3	17	45	69	9.3
4	151	41	58	18.5
5	180	10	30	12.9
6	8	13	15	4.2
7	57	32	54	11.8
8	120	49	25	18.9
9	194	23	19	15.0
10	280	30	40	25.4

Step 1: Input the Data

First, input the data into a SAS dataset:

```
/* Create a dataset named 'ad_data' */
data work.ad_data;
   input Campaign TV_Ad Radio_Ad Newspaper_Ad Sales;
   datalines;
1 230   37      69      22.1
2 44    39      45      10.4
3 17    45      69      9.3
4 151   41      58      18.5
5 180   10      30      12.9
6 8     13      15      4.2
7 57    32      54      11.8
8 120   49      25      18.9
9 194   23      19      15.0
10 280  30      40      25.4
;
run;
```

Step 2: Perform Correlation Analysis

Use the `PROC CORR` procedure to calculate the correlation coefficients between `TV_Ad`, `Radio_Ad`, `Newspaper_Ad`, and `Sales`:

```
/* Calculate correlations between advertising expenditures and sales */
proc corr data=work.ad_data;
   var TV_Ad Radio_Ad Newspaper_Ad Sales;
```

```
run;
```

Output Interpretation:

The output will include a correlation matrix showing the Pearson correlation coefficients between `TV_Ad`, `Radio_Ad`, `Newspaper_Ad`, and `Sales`. It will also provide p-values to test the significance of these correlations.

- **TV_Ad vs. Sales**: A positive correlation indicates that higher spending on TV ads is associated with higher sales.
- **Radio_Ad vs. Sales**: A positive correlation suggests that more spending on radio ads is associated with higher sales.
- **Newspaper_Ad vs. Sales**: A positive correlation indicates that spending more on newspaper ads is associated with higher sales.

Hypothetical Output Interpretation:

Assume the output shows the following correlations:

- **TV_Ad and Sales**: 0.87 (strong positive correlation)
- **Radio_Ad and Sales**: 0.34 (weak positive correlation)
- **Newspaper_Ad and Sales**: 0.24 (weak positive correlation)

These results suggest that TV advertising has the strongest relationship with sales and Radio advertising and sales followed by newspaper advertising has the weak relationship.

Step 3: Optional - Save Correlation Matrix to a Dataset

If you want to save the correlation matrix for further analysis, you can modify the code as follows:

```
/* Save the correlation matrix to a dataset */
proc corr data=work.ad_data outp=work.corr_matrix;
   var TV_Ad Radio_Ad Newspaper_Ad Sales;
run;
```

- **`outp=work.corr_matrix;`**: This saves the correlation matrix to a dataset named `corr_matrix` in the `work` library.

Example Problem: Correlation Between Hours Studied and Exam Scores

Problem Statement: A professor wants to understand the relationship between the number of hours students spend studying and their scores on a final exam. The professor collected data from 10 students, recording the number of hours they studied and their exam scores. The goal is to determine if there is a significant correlation between study hours and exam scores.

Dataset: The dataset contains the following variables:

- `Student`: Identifier for each student.
- `Hours_Studied`: Number of hours the student spent studying.
- `Exam_Score`: The score the student received on the final exam (out of 100).

Here is the sample data:

Student	Hours_Studied	Exam_Score
1	10	78
2	15	85
3	9	72
4	8	68
5	20	90
6	7	65
7	14	88
8	12	82
9	5	58
10	18	92

Step 1: Input the Data

First, input the data into a SAS dataset:

```
/* Create a dataset named 'study_data' */
data work.study_data;
   input Student Hours_Studied Exam_Score;
   datalines;
1 10 78
2 15 85
3 9 72
4 8 68
5 20 90
6 7 65
7 14 88
8 12 82
9 5 58
10 18 92
;
run;
```

Step 2: Perform Correlation Analysis

Use the `PROC CORR` procedure to calculate the correlation coefficient between `Hours_Studied` and `Exam_Score`:

```
/* Calculate correlation between hours studied and exam score */
proc corr data=work.study_data;
   var Hours_Studied Exam_Score;run;
```

Output Interpretation:

The output will include the Pearson correlation coefficient between `Hours_Studied` and `Exam_Score`. It will also provide a p-value to test the significance of this correlation.

- **Positive Correlation**: A positive correlation coefficient (close to +1) would suggest that as the number of hours studied increases, the exam score tends to increase.
- **Negative Correlation**: A negative correlation coefficient (close to -1) would suggest that as the number of hours studied increases, the exam score tends to decrease.
- **No Correlation**: A correlation coefficient close to 0 would suggest that there is no linear relationship between the number of hours studied and the exam score.

Hypothetical Output Interpretation:

Assume the output shows a correlation coefficient of **0.95** between `Hours_Studied` and `Exam_Score`, with a p-value of less than 0.05.

- **Interpretation**: A correlation coefficient of 0.95 indicates a strong positive correlation, meaning that students who study more tend to score higher on the exam. The p-value being less than 0.05 indicates that this correlation is statistically significant.

Example Problem: Correlation Between Physical Activity and Health Metrics

Problem Statement: A health researcher wants to investigate the relationship between the amount of time individuals spend on physical activity each week and their health metrics, such as body mass index (BMI) and resting heart rate. The researcher collected data from 10 participants. The goal is to determine if there is a significant correlation between the weekly physical activity (in hours) and the health metrics.

Dataset: The dataset contains the following variables:

- `Participant`: Identifier for each participant.
- `Activity_Hours`: Number of hours spent on physical activity per week.
- `BMI`: Body Mass Index (a measure of body fat based on height and weight).
- `Heart_Rate`: Resting heart rate (in beats per minute).

Here is the sample data:

Participant	Activity_Hours	BMI	Heart_Rate
1	5	24.5	70
2	8	22.0	68
3	3	27.8	75
4	6	23.5	72
5	2	29.0	80

Participant	Activity_Hours	BMI	Heart_Rate
6	7	21.5	66
7	4	26.2	74
8	9	20.0	64
9	1	30.1	82
10	10	19.5	62

Step 1: Input the Data

First, input the data into a SAS dataset in SAS Cloud:

```
/* Create a dataset named 'health_data' */
data work.health_data;
   input Participant Activity_Hours BMI Heart_Rate;
   datalines;
1 5 24.5 70
2 8 22.0 68
3 3 27.8 75
4 6 23.5 72
5 2 29.0 80
6 7 21.5 66
7 4 26.2 74
8 9 20.0 64
9 1 30.1 82
10 10 19.5 62
;
run;
```

Step 2: Perform Correlation Analysis

Use the `PROC CORR` procedure to calculate the correlation coefficients between `Activity_Hours`, `BMI`, and `Heart_Rate`:

```
/* Calculate correlations between activity hours, BMI, and heart rate */
proc corr data=work.health_data;
   var Activity_Hours BMI Heart_Rate;
run;
```

Hypothetical Output Interpretation:

Assume the output shows the following correlations:

- **Activity_Hours and BMI**: -0.99 (strong negative correlation)
- **Activity_Hours and Heart_Rate**: -0.98 (strong negative correlation)
- **BMI and Heart_Rate**: 0.98 (strong positive correlation)

These results suggest that increased physical activity is strongly associated with lower BMI and a lower resting heart rate, while higher BMI is associated with a higher resting heart rate.

Step 3: Optional - Save Correlation Results to a Dataset

If you want to save the correlation results for further analysis, you can modify the code as follows:

```
/* Save the correlation matrix to a dataset */
proc corr data=work.health_data outp=work.corr_results;
   var Activity_Hours BMI Heart_Rate;
run;
```

- **`outp=work.corr_results;`**: This saves the correlation results to a dataset named `corr_results` in the `work` library.

Linear Regression analysis

To write SAS code for performing regression analysis, you can use the `PROC REG` procedure for linear regression.

Example Problem: Linear Regression Analysis

Problem Statement: You want to examine the relationship between `Hours_Studied` and `Exam_Score` to predict exam scores based on the number of hours studied. We have collected data from 10 students.

Dataset: The dataset contains the following variables:

- `Student_ID`: Identifier for each student.
- `Hours_Studied`: Number of hours spent studying.
- `Exam_Score`: Score obtained in the exam.

Sample Data:

Student_ID	**Hours_Studied**	**Exam_Score**
1	1	50
2	2	55
3	3	60
4	4	65
5	5	70
6	6	75
7	7	80
8	8	85
9	9	90

Student_ID	Hours_Studied	Exam_Score
10	10	95

SAS Code for Regression Analysis

Step 1: Input the Data

```
/* Create a dataset named 'study_data' */
data work.study_data;
   input Student_ID Hours_Studied Exam_Score;
   datalines;
1 1 50
2 2 55
3 3 60
4 4 65
5 5 70
6 6 75
7 7 80
8 8 85
9 9 90
10 10 95
;
run;
```

Step 2: Perform Linear Regression

To analyze the relationship between `Hours_Studied` (independent variable) and `Exam_Score` (dependent variable), use `PROC REG`:

```
/* Perform linear regression analysis */
proc reg data=work.study_data;
   model Exam_Score = Hours_Studied;
run;
```

Output Interpretation

The output will include:

1. **Parameter Estimates**:
    - Coefficients for the intercept and `Hours_Studied` will be shown.
    - We will get the regression equation, which can be used to predict `Exam_Score` based on `Hours_Studied`.
2. **Model Statistics**:
    - **R-Squared**: Indicates how well the model explains the variability in the dependent variable.
    - **p-Value**: Tests the significance of the independent variable's coefficient.
3. **Residuals**:
    - Analysis of residuals helps assess the fit of the model.

Example Output (Hypothetical):

```
                    The REG Procedure
                    Model: MODEL1

                Number of Observations Read        10
                Number of Observations Used        10

Parameter Estimates

Variable         DF    Estimate    Standard Error    t Value    Pr > |t|
Intercept         1    45.0000       2.5000          18.00       <.0001
Hours_Studied     1    5.0000        0.5000          10.00       <.0001

Model Fit Statistics
R-Square: 0.95
Root MSE: 2.500
```

In this hypothetical output:

- The intercept is 45, and the coefficient for `Hours_Studied` is 5. This means the regression equation is `Exam_Score = 45 + 5 * Hours_Studied`.
- **R-Squared** of 0.95 indicates that 95% of the variability in `Exam_Score` is explained by `Hours_Studied`.

Example Problem: Predicting House Prices

Problem Statement: We have a dataset with information about house prices and several features, and we want to build a linear regression model to predict house prices based on these features. The features include `Size` (in square feet), `Bedrooms`, and `Age` (of the house in years).

Dataset: The dataset contains the following columns:

- `House_ID`: Identifier for each house.
- `Size`: Size of the house in square feet.
- `Bedrooms`: Number of bedrooms.
- `Age`: Age of the house in years.
- `House_Price`: Price of the house in thousands of dollars.

Sample Data:

House_ID	Size	Bedrooms	Age	House_Price
1	2000	3	10	300
2	1500	2	15	250

House_ID	Size	Bedrooms	Age	House_Price
3	1800	3	5	280
4	2200	4	20	320
5	1700	3	12	270

SAS Code for Regression Analysis

Step 1: Input the Data

First, you need to create the dataset in SAS.

```
/* Create a dataset named 'house_data' */
data work.house_data;
   input House_ID Size Bedrooms Age House_Price;
   datalines;
1 2000 3 10 300
2 1500 2 15 250
3 1800 3 5 280
4 2200 4 20 320
5 1700 3 12 270
;
run;
```

Step 2: Perform Linear Regression

Now, use the `PROC REG` procedure to perform the regression analysis.

```
/* Perform linear regression analysis */
proc reg data=work.house_data;
   model House_Price = Size Bedrooms Age;
   /* Optionally, you can include diagnostics like plots */
   /* plots(only)=diagnostics;
   */
run;
```

Output Interpretation

The output from `PROC REG` will include:

1. **Parameter Estimates**:
   - Coefficients for the intercept and each predictor variable will be shown.
   - Example output might look like:

```
                 The REG Procedure
                 Model: MODEL1

             Number of Observations Read        5
             Number of Observations Used        5
```

2. Parameter Estimates

Variable	DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	100.0000	0000	infinity	0.0001
Size	1	0.1000	0	infinity	0.0001
Bedrooms	1	-0.00000000	0	-infinity	0.0001
Age	1	000	0	infinity	0.0001

3. Model Fit Statistics R-Square: 0.95 Root MSE: 15.0
4. `Copy code`
5. **Model Statistics**:
   - **R-Squared**: Measures how well the model explains the variability in `House_Price`.
   - **p-Values**: Indicate the significance of each predictor.
6. **Diagnostic Plots** (if requested):
   - Plots like residuals vs. fitted values to assess the fit of the model.

Example Problem: Predicting Salary

Problem Statement: We have a dataset with information about employees' salaries and their years of experience. We want to build a linear regression model to predict `Salary` based on `Years_Experience`.

Dataset: The dataset contains the following columns:

- `Employee_ID`: Identifier for each employee.
- `Years_Experience`: Number of years of experience the employee has.
- `Salary`: Annual salary of the employee in thousands of dollars.

Sample Data:

Employee_ID	Years_Experience	Salary
1	1	50
2	2	55
3	3	60
4	4	65
5	5	70
6	6	75
7	7	80
8	8	85
9	9	90

Employee_ID	Years_Experience	Salary
10	10	95

SAS Code for Regression Analysis

Step 1: Input the Data

First, create the dataset in SAS using the `DATA` step.

```
/* Create a dataset named 'employee_data' */
data work.employee_data;
   input Employee_ID Years_Experience Salary;
   datalines;
1 1 50
2 2 55
3 3 60
4 4 65
5 5 70
6 6 75
7 7 80
8 8 85
9 9 90
10 10 95
;
run;
```

Step 2: Perform Linear Regression

Next, use the `PROC REG` procedure to perform linear regression.

```
/* Perform linear regression analysis */
proc reg data=work.employee_data;
   model Salary = Years_Experience;
   /* Optionally, include diagnostics */
   /* plots(only)=diagnostics;
   */
run;
```

Output Interpretation

The output from `PROC REG` will include:

1. **Parameter Estimates**:
   - Coefficients for the intercept and the `Years_Experience` variable will be shown.
   - Example output might look like:

```
                     The REG Procedure
                     Model: MODEL1

                 Number of Observations Read        10
                 Number of Observations Used        10
```

2. Parameter Estimates
3. Variable DF Estimate Standard Error t Value Pr > |t|

Intercept	1	45.0000	2.5000	18.00	<.0001
Years_Experience	1	5.0000	0.5000	10.00	<.0001

4. Model Fit Statistics R-Square: 0.95 Root MSE: 2.500
5. Copy code
6. **Model Statistics**:
   - **R-Squared**: Indicates the proportion of variance in `Salary` explained by `Years_Experience`.
   - **p-Value**: Tests the significance of the `Years_Experience` coefficient.
7. **Diagnostic Plots** (if requested):
   - Plots such as residuals vs. fitted values to assess the fit of the model.

Example Problem: Predicting House Prices

Problem Statement: We have a dataset with information about house prices and various features. We want to build a linear regression model to predict `House_Price` based on `Size` (in square feet), `Bedrooms`, and `Age` (in years).

Dataset: The dataset contains the following columns:

- `House_ID`: Identifier for each house.
- `Size`: Size of the house in square feet.
- `Bedrooms`: Number of bedrooms.
- `Age`: Age of the house in years.
- `House_Price`: Price of the house in thousands of dollars.

Sample Data:

House_ID	Size	Bedrooms	Age	House_Price
1	2000	3	10	300
2	1500	2	15	250
3	1800	3	5	280
4	2200	4	20	320
5	1700	3	12	270

SAS Code for Regression Analysis

Step 1: Input the Data

Use the `DATA` step to create a dataset in SAS.

```
/* Create a dataset named 'house_data' */
data work.house_data;
   input House_ID Size Bedrooms Age House_Price;
   datalines;
```

```
1 2000 3 10 300
2 1500 2 15 250
3 1800 3 5 280
4 2200 4 20 320
5 1700 3 12 270
;
run;
```

Step 2: Perform Linear Regression

Use `PROC REG` to perform the regression analysis.

```
/* Perform linear regression analysis */
proc reg data=work.house_data;
   model House_Price = Size Bedrooms Age;
   /* Optionally, include diagnostics */
   /* plots(only)=diagnostics;
   */
run;
```

Output Interpretation

The output will include several sections:

1. **Parameter Estimates**:
    - Estimates for the intercept and coefficients for `Size`, `Bedrooms`, and `Age`.
    - Example output might look like:

```
                    The REG Procedure
                    Model: MODEL1

               Number of Observations Read       5
               Number of Observations Used        5
```

2. Parameter Estimates
3. Variable DF Estimate Standard Error t Value Pr > |t|
4. Model Fit Statistics R-Square: 0.95 Root MSE: 15.0
5. Copy code
6. **Model Statistics**:
    - **R-Squared**: Indicates how well the model explains the variability in `House_Price`.
    - **p-Values**: Assess the significance of each predictor variable.
7. **Diagnostic Plots** (if requested):
    - Plots to assess model fit and residuals.

Example Problem: Chi-Square Test for Independence

Problem Statement: If we want to test if there is a significant association between `Gender` and `Smoking_Status` (i.e., whether smoking status is independent of gender). We have a dataset containing these two categorical variables.

Dataset:

Person_ID	Gender	Smoking_Status
1	Male	Smoker
2	Female	Non-Smoker
3	Male	Smoker
4	Female	Smoker
5	Female	Non-Smoker
6	Male	Non-Smoker
7	Female	Smoker
8	Male	Smoker
9	Male	Non-Smoker
10	Female	Non-Smoker

SAS Code for Chi-Square Test

Step 1: Input the Data

First, create the dataset using the `DATA` step.

```
/* Create a dataset named 'survey_data' */
data work.survey_data;
   input Person_ID Gender $ Smoking_Status $;
   datalines;
1 Male Smoker
2 Female Non-Smoker
3 Male Smoker
```

```
4 Female Smoker
5 Female Non-Smoker
6 Male Non-Smoker
7 Female Smoker
8 Male Smoker
9 Male Non-Smoker
10 Female Non-Smoker
;
run;
```

Step 2: Perform Chi-Square Test

Use `PROC FREQ` to perform the chi-square test for independence.

```
/* Perform chi-square test for independence */
proc freq data=work.survey_data;
   tables Gender*Smoking_Status / chisq;
run;
```

Output Interpretation

The output from `PROC FREQ` will include:

1. **Contingency Table**:
   - Shows the frequency distribution of the two categorical variables.
   - Example table might look like:

     ```
     plaintext
     Copy code
           Smoking_Status
        Gender  Non-Smoker  Smoker  Total
        Male        2          3       5
        Female      3          2       5
        Total       5          5      10
     ```

2. **Chi-Square Test Results**:
   - **Chi-Square Statistic**: Measures the difference between observed and expected frequencies.
   - **Degrees of Freedom**: Calculated based on the number of categories.
   - **p-Value**: Indicates the probability of observing the data if the null hypothesis is true.

   Example output might look like:

   ```
           Chi-Square Test Results

    Chi-Square Statistic  = 0.5271
    DF (Degrees of Freedom) = 1
    Pr > Chi-Square        = 1.000
   ```

   - If the **p-value** is greater than a significance level (e.g., 0.05), we accept the null hypothesis, suggesting a significant association between the variables.

Problem Statement: If we want to determine if there is an association between a person's education level (`Education`) and whether they own a car (`Car_Ownership`). we have a dataset with two categorical variables: `Education` (which can be "High School", "Bachelor's", or "Master's") and `Car_Ownership` (which can be "Yes" or "No").

Sample Data:

Person_ID	Education	Car_Ownership
1	High School	Yes
2	Bachelor's	No
3	Master's	Yes
4	High School	No
5	Bachelor's	Yes
6	Master's	Yes
7	High School	No
8	Bachelor's	Yes
9	Master's	Yes
10	Bachelor's	No

SAS Code for Chi-Square Test

Step 1: Input the Data

First, create the dataset in SAS.

```
/* Create a dataset named 'education_data' */
data work.education_data;
   input Person_ID Education $ Car_Ownership $;
   datalines;
1 "High School" Yes
2 "Bachelor's" No
3 "Master's" Yes
4 "High School" No
5 "Bachelor's" Yes
6 "Master's" Yes
7 "High School" No
8 "Bachelor's" Yes
```

```
9 "Master's" Yes
10 "Bachelor's" No
;
run;
```

Step 2: Perform the Chi-Square Test

Use `PROC FREQ` to perform the chi-square test for independence.

```
/* Perform chi-square test for independence */
proc freq data=work.education_data;
   tables Education*Car_Ownership / chisq;
run;
```

Output Interpretation

The output from `PROC FREQ` will include:

1. **Contingency Table**:
   - Shows the frequency distribution of `Education` and `Car_Ownership`.
   - Example table might look like:

```
        Car_Ownership
Education      No        Yes     Total
High School     3         0         3
Bachelor's      2         2         4
Master's        0         3         3
Total           5         5        10
```

2. **Chi-Square Test Results**:
   - **Chi-Square Statistic**: Measures the difference between observed and expected frequencies.
   - **Degrees of Freedom**: Calculated based on the number of categories.
   - **p-Value**: Indicates the probability of observing the data if the null hypothesis is true.

   Example output might look like:

```
        Chi-Square Test Results

Chi-Square Statistic   = 0.0113
DF (Degrees of Freedom) = 2
Pr > Chi-Square         = 0.189
```

   - If the **p-value** is less than a significance level (e.g., 0.05), you would reject the null hypothesis, indicating a significant association between `Education` and `Car_Ownership`. In this case, with a p-value of 0.189, you would not reject the null hypothesis, suggesting no significant association.

Example Problem: Association Between Gender and Voting Preference

Problem Statement: We want to determine if there is an association between `Gender` and `Voting_Preference` in a recent election. The dataset contains information on whether people prefer Candidate A or Candidate B, categorized by gender.

Sample Data:

Person_ID	Gender	Voting_Preference
1	Male	A
2	Female	B
3	Female	A
4	Male	B
5	Female	B
6	Male	A
7	Male	A
8	Female	A
9	Male	B
10	Female	A

SAS Code for Chi-Square Test

Step 1: Input the Data

First, create the dataset in SAS.

```
/* Create a dataset named 'voting_data' */
data work.voting_data;
   input Person_ID Gender $ Voting_Preference $;
   datalines;
1 Male A
2 Female B
3 Female A
4 Male B
5 Female B
6 Male A
```

```
7 Male A
8 Female A
9 Male B
10 Female A
;
run;
```

Step 2: Perform the Chi-Square Test

Use `PROC FREQ` to perform the chi-square test for independence.

```
/* Perform chi-square test for independence */
proc freq data=work.voting_data;
   tables Gender*Voting_Preference / chisq;
run;
```

Expected Output

The output from `PROC FREQ` will include:

1. **Contingency Table**:
   - Shows the frequency distribution of `Gender` and `Voting_Preference`.
   - Example table might look like:

```
                    Voting_Preference

Gender          A        B     Total
Male            3        2         5
Female          3        2         5
Total           6        4        10
```

2. **Chi-Square Test Results**:
   - **Chi-Square Statistic**: Measures the difference between observed and expected frequencies.
   - **Degrees of Freedom**: Calculated based on the number of categories.
   - **p-Value**: Indicates the probability of observing the data if the null hypothesis is true.

   Example output might look like:

```
plaintext
Copy code
        Chi-Square Test Results

 Chi-Square Statistic   = 0.000
 DF (Degrees of Freedom) = 1
 Pr > Chi-Square         = 1.000
```

   - If the **p-value** is less than a significance level (e.g., 0.05), you would reject the null hypothesis, indicating a significant association between `Gender` and `Voting_Preference`. In this case, with a p-value of 1.000, you would not reject the null hypothesis, suggesting no significant association.

Example Problem: Association Between Age Group and Product Preference

Problem Statement: A retail store wants to determine if there is an association between customers' age groups and their preference for two different products, Product X and Product Y. The store has collected data on customers' age groups and the product they prefer.

Sample Data:

Customer_ID	Age_Group	Product_Preference
1	Under 30	X
2	30-50	Y
3	Over 50	X
4	Under 30	Y
5	30-50	X
6	Over 50	Y
7	Under 30	X
8	30-50	Y
9	Over 50	X
10	30-50	X

SAS Code for Chi-Square Test

Step 1: Input the Data

First, you need to create the dataset in SAS.

```
/* Create a dataset named 'product_data' */
data work.product_data;
   input Customer_ID Age_Group $ Product_Preference $;
   datalines;
1 "Under 30" X
2 "30-50" Y
3 "Over 50" X
4 "Under 30" Y
5 "30-50" X
6 "Over 50" Y
```

```
7 "Under 30" X
8 "30-50" Y
9 "Over 50" X
10 "30-50" X
;
run;
```

Step 2: Perform the Chi-Square Test

Use `PROC FREQ` to perform the chi-square test for independence between `Age_Group` and `Product_Preference`.

```
/* Perform chi-square test for independence */
proc freq data=work.product_data;
   tables Age_Group*Product_Preference / chisq;
run;
```

Expected Output

The output will include:

1. **Contingency Table**:
   - Displays the frequency distribution of `Age_Group` and `Product_Preference`.
   - Example table:

```
                Product_Preference
Age_Group           X        Y       Total
Under 30            3        0         3
30-50               2        2         4
Over 50             3        0         3
Total               8        2        10
```

2. **Chi-Square Test Results**:
   - **Chi-Square Statistic**: Measures the association between `Age_Group` and `Product_Preference`.
   - **Degrees of Freedom**: Based on the number of categories in the variables.
   - **p-Value**: The probability of observing the data if the null hypothesis (no association) is true.

   Example output might look like:

```
       Chi-Square Test Results

 Chi-Square Statistic   = 0.0028
 DF (Degrees of Freedom) = 6
 Pr > Chi-Square         = 1.000
```

   - If the **p-value** is less than a significance level (e.g., 0.05), you would reject the null hypothesis, indicating a significant association between `Age_Group` and

`Product_Preference`. In this example, with a p-value of 1.000, you would not reject the null hypothesis, suggesting no significant association.

Example Problem: Association Between Education Level and Job Satisfaction

Problem Statement: A company wants to determine if there is an association between the education level of employees and their job satisfaction. The dataset includes the education level (`Education_Level`) and job satisfaction (`Job_Satisfaction`) of employees.

Sample Data:

Employee_ID	Education_Level	Job_Satisfaction
1	High School	Satisfied
2	Bachelor's	Dissatisfied
3	Master's	Satisfied
4	High School	Dissatisfied
5	Bachelor's	Satisfied
6	Master's	Satisfied
7	High School	Dissatisfied
8	Bachelor's	Satisfied
9	Master's	Dissatisfied
10	Bachelor's	Satisfied

SAS Code for Chi-Square Test

Step 1: Input the Data

First, input the data into SAS using the `DATA` step.

```
/* Create a dataset named 'job_satisfaction_data' */
data work.job_satisfaction_data;
   input Employee_ID Education_Level $ Job_Satisfaction $;
   datalines;
1 "High School" Satisfied
2 "Bachelor's" Dissatisfied
3 "Master's" Satisfied
```

```
4 "High School" Dissatisfied
5 "Bachelor's" Satisfied
6 "Master's" Satisfied
7 "High School" Dissatisfied
8 "Bachelor's" Satisfied
9 "Master's" Dissatisfied
10 "Bachelor's" Satisfied
;
run;
```

Step 2: Perform the Chi-Square Test

Use `PROC FREQ` to perform the chi-square test for independence between `Education_Level` and `Job_Satisfaction`.

```
/* Perform chi-square test for independence */
proc freq data=work.job_satisfaction_data;
   tables Education_Level*Job_Satisfaction / chisq;
run;
```

Expected Output

The output from `PROC FREQ` will include:

1. **Contingency Table**:
   - Displays the frequency distribution of `Education_Level` and `Job_Satisfaction`.
   - Example table:

```
            Job_Satisfaction
Education_Level Satisfied Dissatisfied Total
High School          1          2          3
Bachelor's           4          1          5
Master's             2          1          3
Total                7          4         10
```

2. **Chi-Square Test Results**:
   - **Chi-Square Statistic**: Measures the difference between observed and expected frequencies.
   - **Degrees of Freedom**: Calculated based on the number of categories in the variables.
   - **p-Value**: Indicates the probability of observing the data if the null hypothesis (no association) is true.

   Example output might look like:

```
       Chi-Square Test Results

Chi-Square Statistic  = 2.667
DF (Degrees of Freedom) = 2
Pr > Chi-Square        = 0.264
```

- If the **p-value** is less than a significance level (e.g., 0.05), you would reject the null hypothesis, indicating a significant association between `Education_Level` and `Job_Satisfaction`. In this example, with a p-value of 0.264, you would not reject the null hypothesis, suggesting no significant association.

Example Problem: Comparing Average Salaries Between Two Departments

Problem Statement: A company wants to compare the average salaries of employees in two departments: `Department A` and `Department B`. The dataset contains employee salaries along with their department information.

Sample Data:

Employee_ID	Department	Salary
1	A	55000
2	A	60000
3	B	58000
4	A	62000
5	B	59000
6	A	61000
7	B	57000
8	B	60000
9	A	64000
10	B	56000

SAS Code for t-Test

Step 1: Input the Data

First, input the data into SAS.

```
/* Create a dataset named 'salary_data' */
data work.salary_data;
   input Employee_ID Department $ Salary;
   datalines;
```

```
1 A 55000
2 A 60000
3 B 58000
4 A 62000
5 B 59000
6 A 61000
7 B 57000
8 B 60000
9 A 64000
10 B 56000
;
run;
```

Step 2: Perform the t-Test

Use `PROC TTEST` to perform the t-test, comparing the average salaries between `Department A` and `Department B`.

```
/* Perform t-test to compare average salaries between departments */
proc ttest data=work.salary_data;
   class Department;
   var Salary;
run;
```

Expected Output

The output from `PROC TTEST` will include:

1. **Descriptive Statistics**:
   - Mean, standard deviation, and other summary statistics for each department.
2. **t-Test Results**:
   - **t-Statistic**: Measures the difference between the two group means relative to the variability in the data.
   - **Degrees of Freedom**: Reflects the sample size.
   - **p-Value**: Indicates whether the difference in means is statistically significant.

   Example output might look like:

```
              TTEST PROCEDURE
                     Statistics
Department     N       Mean      Std Dev
A              5       60400     3500
B              5       58000     1520

t Value = 1.656
DF (Degrees of Freedom) = 8
Pr > |t| = 0.142
```

   - If the **p-value** is less than a significance level (e.g., 0.05), you would reject the null hypothesis, indicating a significant difference in average salaries between the two departments. In this example, with a p-value of 0.142, you would not reject the null hypothesis, suggesting no significant difference.

This example demonstrates how to perform and interpret a t-test in SAS to compare the means of two groups.

Example: Comparing Test Scores Between Two Classes

Problem Statement: A school wants to compare the average test scores of students from two different classes, `Class A` and `Class B`. The dataset contains the test scores of students from both classes.

Sample Data:

Student_ID	Class	Test_Score
1	A	85
2	A	88
3	A	90
4	A	92
5	A	87
6	B	78
7	B	82
8	B	80
9	B	85
10	B	83

SAS Code for t-Test

Step 1: Input the Data

First, input the data into SAS.

```
/* Create a dataset named 'test_scores' */
data work.test_scores;
   input Student_ID Class $ Test_Score;
   datalines;
1 A 85
2 A 88
3 A 90
4 A 92
5 A 87
6 B 78
7 B 82
8 B 80
9 B 85
10 B 83
;
run;
```

Step 2: Perform the t-Test

Use `PROC TTEST` to perform the t-test, comparing the average test scores between `Class A` and `Class B`.

```
/* Perform t-test to compare average test scores between classes */
proc ttest data=work.test_scores;
   class Class;
   var Test_Score;
run;
```

Expected Output

The output from `PROC TTEST` will include:

1. **Descriptive Statistics**:
   - Mean, standard deviation, and other summary statistics for each class.
2. **t-Test Results**:
   - **t-Statistic**: Measures the difference between the two group means relative to the variability in the data.
   - **Degrees of Freedom (DF)**: Reflects the sample size.
   - **p-Value**: Indicates whether the difference in means is statistically significant.

   Example output might look like:

```
plaintext
Copy code
              TTEST PROCEDURE
                     Statistics
    Class        N       Mean      Std Dev
    A            5       88.4      2.59
    B            5       81.6      2.86

    t Value = 4.054
    DF (Degrees of Freedom) = 8
```

```
Pr > |t| = 0.004
```

- If the **p-value** is less than a significance level (e.g., 0.05), you would reject the null hypothesis, indicating a significant difference in average test scores between the two classes. In this example, with a p-value of 0.004, you would conclude that there is a significant difference in average test scores between Class A and Class B.

Example: Comparing Mean Heights Between Two Groups

Problem Statement: A researcher wants to compare the average heights of two groups of people, `Group 1` and `Group 2`. The dataset contains the height measurements for individuals in both groups.

Sample Data:

Person_ID	Group	Height
1	1	170
2	1	172
3	1	168
4	1	174
5	1	169
6	2	165
7	2	163
8	2	167
9	2	166
10	2	162

SAS Code for t-Test

Step 1: Input the Data

You first need to input the data into SAS.

```
/* Create a dataset named 'height_data' */
data work.height_data;
```

```
   input Person_ID Group $ Height;
   datalines;
1 1 170
2 1 172
3 1 168
4 1 174
5 1 169
6 2 165
7 2 163
8 2 167
9 2 166
10 2 162
;
run;
```

Step 2: Perform the t-Test

Use `PROC TTEST` to perform the t-test, comparing the average heights between `Group 1` and `Group 2`.

```
/* Perform t-test to compare average heights between groups */
proc ttest data=work.height_data;
   class Group;
   var Height;
run;
```

Expected Output

The output from `PROC TTEST` will include:

1. **Descriptive Statistics**:
   - Mean, standard deviation, and other summary statistics for each group.
2. **t-Test Results**:
   - **t-Statistic**: Measures the difference between the two group means relative to the variability in the data.
   - **Degrees of Freedom (DF)**: Reflects the sample size.
   - **p-Value**: Indicates whether the difference in means is statistically significant.

   Example output might look like:

   ```
   plaintext
   Copy code
                 TTEST PROCEDURE
                        Statistics
       Group        N       Mean      Std Dev
       1            5       170.6     2.41
       2            5       164.6     2.33

       t Value = 4.472
       DF (Degrees of Freedom) = 8
       Pr > |t| = 0.002
   ```

- If the **p-value** is less than a significance level (e.g., 0.05), you would reject the null hypothesis, indicating a significant difference in average heights between the two groups. In this example, with a p-value of 0.002, you would conclude that there is a significant difference in average heights between Group 1 and Group 2.

Example: Comparing Mean Weights Between Two Groups

Problem Statement: A researcher wants to compare the average weights of two different diets, `Diet A` and `Diet B`. The dataset contains the weight measurements of individuals on both diets.

Sample Data:

Subject_ID	Diet	Weight
1	A	68
2	A	70
3	A	65
4	A	72
5	A	66
6	B	75
7	B	78
8	B	76
9	B	74
10	B	77

SAS Code for t-Test

Step 1: Input the Data

First, input the data into SAS.

```
/* Create a dataset named 'weight_data' */
data work.weight_data;
   input Subject_ID Diet $ Weight;
   datalines;
1 A 68
```

```
2 A 70
3 A 65
4 A 72
5 A 66
6 B 75
7 B 78
8 B 76
9 B 74
10 B 77
;
run;
```

Step 2: Perform the t-Test

Use `PROC TTEST` to compare the average weights between `Diet A` and `Diet B`.

```
sas
Copy code
/* Perform t-test to compare average weights between diets */
proc ttest data=work.weight_data;
   class Diet;
   var Weight;
run;
```

Expected Output

When you run this code in SAS, the output from `PROC TTEST` will include:

1. **Descriptive Statistics**:
   - Mean, standard deviation, and other summary statistics for each diet.
2. **t-Test Results**:
   - **t-Statistic**: A measure of the difference between the two group means relative to the variability in the data.
   - **Degrees of Freedom (DF)**: Reflects the sample size.
   - **p-Value**: Indicates whether the difference in means is statistically significant.

   Example output might look like this:

   ```
   plaintext
   Copy code
                  TTEST PROCEDURE
                         Statistics
       Diet         N      Mean     Std Dev
       A            5      68.2     2.59
       B            5      76.0     1.58

       t Value = -5.776
       DF (Degrees of Freedom) = 8
       Pr > |t| < 0.001
   ```

   - If the **p-value** is less than the significance level (commonly 0.05), you reject the null hypothesis, indicating a significant difference in average weights between the two diets.

In this example, with a p-value less than 0.001, you would conclude that there is a significant difference between Diet A and Diet B.

Example: Comparing Average Exam Scores Between Two Classes

Problem Statement: A teacher wants to compare the average exam scores of students in two different classes, `Class 1` and `Class 2`. The dataset contains the exam scores of students from both classes.

Sample Data:

Student_ID	Class	Exam_Score
1	1	85
2	1	88
3	1	90
4	1	87
5	1	86
6	2	78
7	2	82
8	2	80
9	2	83
10	2	79

SAS Code for t-Test

Step 1: Input the Data

First, input the data into SAS.

```
/* Create a dataset named 'exam_scores' */
```

```
data work.exam_scores;
   input Student_ID Class $ Exam_Score;
   datalines;
1 1 85
2 1 88
3 1 90
4 1 87
5 1 86
6 2 78
7 2 82
8 2 80
9 2 83
10 2 79
;
run;
```

Step 2: Perform the t-Test

Use `PROC TTEST` to perform the t-test, comparing the average exam scores between `Class 1` and `Class 2`.

```
/* Perform t-test to compare average exam scores between classes */
proc ttest data=work.exam_scores;
   class Class;
   var Exam_Score;
run;
```

Expected Output

Running the code will provide you with:

1. **Descriptive Statistics**:
   - Mean, standard deviation, and other summary statistics for each class.
2. **t-Test Results**:
   - **t-Statistic**: A measure of the difference between the two group means relative to the variability in the data.
   - **Degrees of Freedom (DF)**: Reflects the sample size.
   - **p-Value**: Indicates whether the difference in means is statistically significant.

For example:

```
           TTEST PROCEDURE
                  Statistics
Class        N      Mean     Std Dev
1            5      87.2     2.17
2            5      80.4     2.07

t Value = 5.776
DF (Degrees of Freedom) = 8
Pr > |t| = 0.0004
```

- If the **p-value** is less than the significance level (e.g., 0.05), you reject the null hypothesis, suggesting a significant difference in average exam scores between the two classes.

This code performs a t-test in SAS, enabling you to compare the means of two groups effectively.

To write SAS code for conducting a Z-test, you can use the `PROC IML` (Interactive Matrix Language) procedure, `DATA` step, or `PROC TTEST` with some customization. Below is a basic example of how to perform a one-sample Z-test.

Example: One-Sample Z-Test in SAS

Assume you have a dataset where you want to test whether the mean of a sample is equal to a known population mean.

Steps:

1. **Input the data**: Enter the sample data into a dataset.
2. **Calculate the Z-test statistic**.
3. **Compute the p-value** based on the Z statistic.
4. **Draw conclusions** based on the p-value.

SAS Code Example

```
sas
Copy code
/* Step 1: Input the data */
data sample_data;
    input value @@;
    datalines;
    50.1 49.8 50.3 50.5 50.7 49.6 50.2 50.0 50.4 49.9
    ;
run;

/* Step 2: Calculate the Z statistic */
proc means data=sample_data mean stddev noprint;
    var value;
    output out=stats mean=mean std=stddev n=n;
run;

data z_test;
    set stats;
    /* Known population mean */
    population_mean = 50;
    /* Calculate the Z statistic */
```

```
    z_stat = (mean - population_mean) / (stddev / sqrt(n));
    /* Compute the p-value */
    p_value = 2 * (1 - probnorm(abs(z_stat)));
run;

/* Step 3: Display the results */
proc print data=z_test;
    var mean stddev n z_stat p_value;
run;
```

Explanation:

- **DATA Step**: You input your sample data. Each value represents a data point in the sample.
- **PROC MEANS**: This procedure computes the sample mean (`mean`), standard deviation (`stddev`), and sample size (`n`).
- **Z-Test Calculation**: The Z-statistic is calculated using the formula:

$$Z = \frac{\text{sample mean} - \text{population mean}}{\frac{\text{standard deviation}}{\sqrt{n}}}$$

  The `probnorm` function computes the p-value associated with the calculated Z-statistic.

- **PROC PRINT**: Displays the mean, standard deviation, sample size, Z statistic, and p-value.

This is a basic example for a one-sample Z-test. You can modify the code for other types of Z-tests, such as two-sample Z-tests, by adjusting the formula and data input.

Suppose a manufacturer claims that the average weight of a product is 500 grams. A quality control manager takes a random sample of 30 products and finds the following weights (in grams):

```
498, 505, 499, 502, 497, 501, 503, 499, 500, 504, 506, 498, 501, 499, 502,
500, 498, 504, 497, 503, 501, 502, 498, 505, 499, 501, 504, 503, 497, 500
```

We want to test whether the average weight is significantly different from 500 grams using a Z-test at a 5% significance level.

```
/* Step 1: Input the data */
data weights;
    input weight @@;
    datalines;
    498 505 499 502 497 501 503 499 500 504
    506 498 501 499 502 500 498 504 497 503
    501 502 498 505 499 501 504 503 497 500
    ;
```

```
run;

/* Step 2: Calculate the sample mean, standard deviation, and Z statistic */
proc means data=weights mean stddev noprint;
    var weight;
    output out=stats mean=mean std=stddev n=n;
run;

data z_test;
    set stats;
    /* Known population mean */
    population_mean = 500;
    /* Calculate the Z statistic */
    z_stat = (mean - population_mean) / (stddev / sqrt(n));
    /* Compute the p-value */
    p_value = 2 * (1 - probnorm(abs(z_stat)));
run;

/* Step 3: Display the results */
proc print data=z_test;
    var mean stddev n z_stat p_value;
    title "Z-Test Results for Product Weight";
run;
```

Z-Test Results for Product Weight

Obs	mean	stddev	n	z_stat	p_value
1	500.867	2.63574	30	1.80099	0.071705

Interpretation:

- **Z-Statistic**: A high absolute value of the Z-statistic indicates that the difference in blood pressure before and after the diet is significant.
- **P-Value**: If the p-value is less than the significance level (0.05), you reject the null hypothesis that there is no difference in blood pressure before and after the diet.

Imagine a company wants to test whether a new training program improves employee productivity. The productivity (in units produced per hour) of 10 employees is measured before and after the training. The data is as follows:

Employee	Productivity Before	Productivity After
1	50	55
2	52	53
3	48	49
4	47	50

Employee	Productivity Before	Productivity After
5	46	48
6	49	51
7	45	47
8	53	54
9	50	52
10	51	53

We want to test whether there is a significant increase in productivity after the training using a paired Z-test.

SAS Code:

```
/* Step 1: Input the data */
data productivity;
    input employee before after;
    diff = before - after;
    datalines;
    1 50 55
    2 52 53
    3 48 49
    4 47 50
    5 46 48
    6 49 51
    7 45 47
    8 53 54
    9 50 52
    10 51 53
    ;
run;

/* Step 2: Calculate the mean and standard deviation of the differences */
proc means data=productivity mean stddev noprint;
    var diff;
    output out=stats mean=mean_diff std=stddev_diff n=n;
run;

data z_test;
    set stats;
    /* Hypothesized difference (usually 0) */
    population_mean_diff = 0;
    /* Calculate the Z statistic */
    z_stat = (mean_diff - population_mean_diff) / (stddev_diff / sqrt(n));
    /* Compute the p-value */
    p_value = 2 * (1 - probnorm(abs(z_stat)));
run;

/* Step 3: Display the results */
proc print data=z_test;
    var mean_diff stddev_diff n z_stat p_value;
    title "Paired Z-Test Results for Productivity Improvement";
run;
```

Interpretation:

- **Z-Statistic**: If the Z-statistic has a high absolute value, it suggests that the difference in productivity before and after the training is significant.
- **P-Value**: If the p-value is less than the significance level (e.g., 0.05), you reject the null hypothesis that there is no difference in productivity before and after the training.

One Way ANOVA Classification

Example Problem:

Suppose a researcher is interested in studying the effect of different diets on weight loss. The researcher randomly assigns 15 participants to one of three diets (A, B, or C). After 8 weeks, the weight loss (in kg) is recorded for each participant. The data is as follows:

Participant	Diet	Weight Loss (kg)
1	A	3.1
2	A	2.8
3	A	3.3
4	A	3.0
5	A	3.2
6	B	2.7
7	B	2.5
8	B	2.9
9	B	2.8
10	B	3.0
11	C	2.0
12	C	2.2
13	C	2.3
14	C	2.5
15	C	2.1

The goal is to determine if there is a significant difference in the mean weight loss among the three diets.

SAS Code:

```
/* Step 1: Input the data */
data diet_study;
    input participant diet $ weight_loss;
    datalines;
```

```
    1 A 3.1
    2 A 2.8
    3 A 3.3
    4 A 3.0
    5 A 3.2
    6 B 2.7
    7 B 2.5
    8 B 2.9
    9 B 2.8
    10 B 3.0
    11 C 2.0
    12 C 2.2
    13 C 2.3
    14 C 2.5
    15 C 2.1
    ;
run;

/* Step 2: Perform the One-Way ANOVA */
proc anova data=diet_study;
    class diet;
    model weight_loss = diet;
    means diet / tukey;
    title "One-Way ANOVA for Diet Effect on Weight Loss";
run;

/* Step 3: Display ANOVA Table */
proc print data=diet_study;
run;
```

Interpretation:

- **F-Statistic**: The F-statistic tests whether the group means are equal. A higher value suggests significant differences between group means.
- **P-Value**: If the p-value is less than the significance level (e.g., 0.05), you reject the null hypothesis that all group means are equal.

The output will allow you to determine if there is a statistically significant difference in weight loss between the different diets. If the ANOVA is significant, the Tukey's HSD test will indicate which specific diets differ from each other.

A company wants to determine if three different marketing strategies (A, B, and C) have different effects on sales. They implement each strategy in three different regions, and the sales (in thousands of dollars) are recorded for each region.

The data is as follows:

Region	**Strategy**	**Sales (in $1000)**
1	A	50
2	A	55
3	A	53

Region	Strategy	Sales (in $1000)
4	B	57
5	B	59
6	B	56
7	C	48
8	C	52
9	C	50

The goal is to determine if there is a significant difference in mean sales among the three marketing strategies.

SAS Code:

```
/* Step 1: Input the data */
data marketing_study;
    input region strategy $ sales;
    datalines;
    1 A 50
    2 A 55
    3 A 53
    4 B 57
    5 B 59
    6 B 56
    7 C 48
    8 C 52
    9 C 50
    ;
run;

/* Step 2: Perform the One-Way ANOVA */
proc anova data=marketing_study;
    class strategy;
    model sales = strategy;
    means strategy / tukey;
    title "One-Way ANOVA for Marketing Strategy Effect on Sales";
run;

/* Step 3: Display the Dataset (optional) */
proc print data=marketing_study;
    title "Marketing Study Data";
run;
```

Interpretation:

- **F-Statistic**: The F-statistic tests whether the group means (sales for each strategy) are equal. A high F-statistic suggests significant differences between group means.

- **P-Value**: If the p-value is less than the significance level (e.g., 0.05), you reject the null hypothesis that all group means are equal, indicating that at least one marketing strategy is significantly different in its effect on sales.
- **Tukey's HSD Test**: If the ANOVA is significant, the Tukey's test will indicate which specific strategies differ from each other.