Pharmaceutical Production Case Study - SAS

Avani Joshi

June 25th, 2017

A. Document Information

Document Name	Pharmaceutical Production Case Study - SAS
Document Author	Avani Joshi
Document Objective	
Document Version	Version 2
Release Date	25th June, 2017

B. Document History

Version No	Date	Section No	Description of Change	Author
V1	26th May 2017		First Version	
V2	25th June 2017		Second Version	

1. BACKGROUND

1.1 Introduction

Drug discovery teams are often faced with data for which the samples have been categorized into two or more groups. For example, early in the drug discovery process, high throughput screening is used to identify compounds' activity status against a specific biological target. At a subsequent stage of discovery, screens are used to measure compounds' solubility, permeability, and toxicity status. In other areas of drug discovery, information from animal models on disease classification, survival status, and occurrence of adverse events is obtained and scrutinized. Based on these categorizations, teams must decide which compounds to pursue for further development.

In addition to the categorical response, discovery data often contain variables that describe features of the samples. For example, many computational chemistry software packages have the ability to generate structural and physical-property descriptors for any defined set of compounds. In genomics and proteomics, expression profiles can be measured on tissue samples.

1.2 Problem Statement

Generally speaking, permeability is the ability of a molecule to cross a membrane. In the body, key membranes exist in the intestine and brain, and are composed of layers of molecules and proteins organized in a way to prevent harmful substances from crossing while allowing essential substances to pass through. Because a compound's permeability status is critically important to its success, pharmaceutical companies would like to identify poorly permeable compounds as early as possible in the discovery process.

We have to perform the identification of the same on the "permy" SAS data set with 354 variables, and classify our compounds on the basis of permeability.

Training Data Results

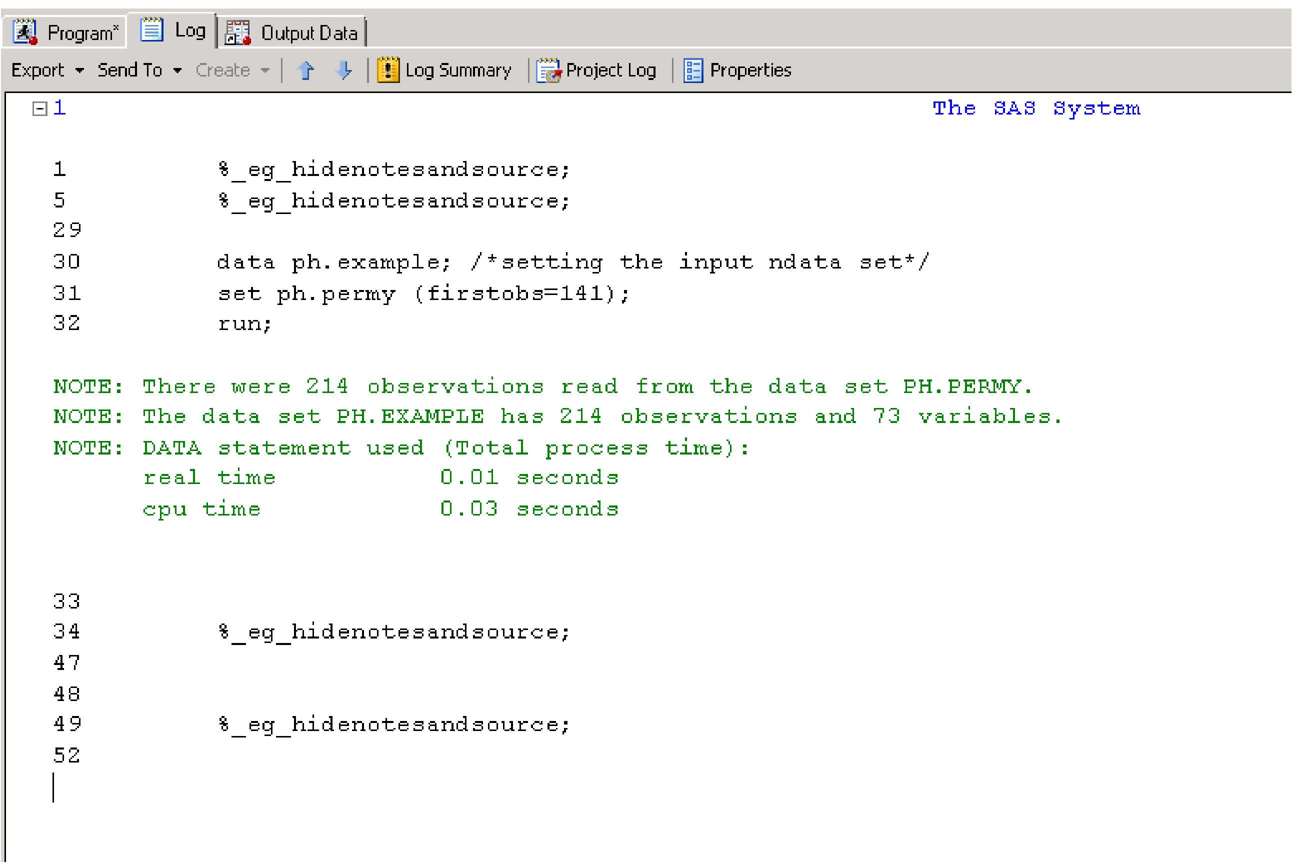
Step 1- Importing the data set into enterprise guide and printing the dataset

We start with importing our dataset into our local SAS server and print the dataset to have a thorough look at our data. We use only the training data set here (observations starting from 141 till 354).

Code used –

```
libname ph 'C:\SAS';
data ph.example; /*setting the input ndata set*/
set ph.permy (firstobs=141);
run;
proc print data=ph.example;
run;
```

Output-



Step 2 – Using The Means procedure to calculate descriptive statistics for the dataset

In the next step we are using the MEANS procedure to calculate all the descriptive statistics like mean, median, interquartile ranges etc. This helps us to analyze our dataset better and know the distribution of our data.

Code used –

```
proc means data=ph.example printalltypes n mean median std min max q1
q3;
class y;
var x1-x10;
run;
```

Output-

The MEANS Procedure

N Obs	Variable	N	Mean	Median	Std Dev	Minimum	Maximum	Lower Quartile	Upper Quartile
214	x1	214	884.2757009	883.1250000	164.4834417	477.0000000	1258.63	773.7500000	989.8750000
	x2	214	599.3289673	601.3310000	101.6303037	332.9160000	837.4430000	531.9980000	665.6470000
	x3	214	1.4717196	1.4670000	0.0416471	1.3710000	1.6340000	1.4460000	1.4890000
	x4	214	1.5576262	1.5645000	0.1179401	1.1970000	1.8310000	1.4840000	1.6370000
	x5	214	1553.50	1591.44	239.0947752	738.1250000	2121.75	1388.00	1727.25
	x6	214	1012.19	1022.00	214.7105889	309.5000000	1504.38	866.0000000	1164.50
	x7	214	584.8750000	595.3750000	155.2068073	138.3750000	974.2500000	472.6250000	688.3750000
	x8	214	246.4339953	249.6250000	78.6629285	57.8750000	464.5000000	199.3750000	290.7500000
	x9	214	126.8539720	129.2500000	45.0731636	27.5000000	260.6250000	95.8750000	153.2500000
	x10	214	66.4509346	65.7500000	26.0457049	12.0000000	145.7500000	48.7500000	80.5000000

y	N Obs	Variable	N	Mean	Median	Std Dev	Minimum	Maximum	Lower Quartile	Upper Quartile
0	107	x1	107	924.0642523	916.2500000	144.1685665	628.3750000	1258.63	826.2500000	1017.00
		x2	107	626.2027850	620.1900000	91.0968870	429.6280000	837.4430000	558.5800000	676.5430000
		x3	107	1.4738037	1.4680000	0.0341750	1.4000000	1.5750000	1.4500000	1.4880000
		x4	107	1.5922150	1.5940000	0.1081788	1.3330000	1.8310000	1.5180000	1.6690000
		x5	107	1648.38	1684.75	211.8024358	984.2500000	2121.75	1507.50	1812.38
		x6	107	1096.48	1122.88	200.3220305	451.2500000	1504.38	955.6250000	1244.00
		x7	107	642.3235981	667.6250000	152.4106005	166.5000000	974.2500000	554.1250000	750.0000000
		x8	107	274.1214953	275.5000000	82.4566736	61.5000000	464.5000000	220.2500000	323.0000000
		x9	107	140.8726636	143.5000000	48.4019262	27.5000000	260.6250000	110.3750000	162.5000000
		x10	107	73.9474299	71.3750000	28.7390260	12.0000000	145.7500000	55.1250000	89.2500000
1	107	x1	107	844.4871495	836.0000000	174.3104575	477.0000000	1250.50	741.2500000	939.7500000
		x2	107	572.4551495	570.9670000	104.8723251	332.9160000	801.3790000	499.7960000	644.7330000
		x3	107	1.4696355	1.4640000	0.0480482	1.3710000	1.6340000	1.4380000	1.4900000
		x4	107	1.5230374	1.5310000	0.1176136	1.1970000	1.7940000	1.4370000	1.6060000
		x5	107	1458.62	1487.88	227.6789917	738.1250000	1923.75	1291.13	1636.25
		x6	107	927.9007009	948.2500000	195.3548188	309.5000000	1279.50	795.3750000	1072.75
		x7	107	527.4264019	535.8750000	136.0647552	138.3750000	819.3750000	443.8750000	635.1250000
		x8	107	218.7464953	222.7500000	63.9322000	57.8750000	383.2500000	177.1250000	263.0000000
		x9	107	112.8352804	115.7500000	36.6448389	31.6250000	198.3750000	85.2500000	139.7500000
		x10	107	58.9544393	60.5000000	20.5856857	16.7500000	103.8750000	43.3750000	75.1250000

Page Break

Note: the output above is just for first 10 variables, but the results are performed for all 71 in the same manner.

Step 3 – Finding the number of missing values and imputing them with 0

Before performing any operations on our data, it's important to normalize and cleanse our data. Determining missing values and resolving them is a part of the same process.

Code Used-

```
proc format; /* create a format to group missing and nonmissing */
     value $missfmt ' '='Missing' other='Not Missing';
     value  missfmt  . ='Missing' other='Not Missing';
      value zmissfmt 0='Missing' other = 'Not Missing';
run;
proc freq data=ph.example;
     format _CHAR_ $missfmt.;
     tables _CHAR_ / missing missprint nocum nopercent;
     format _NUMERIC_ missfmt.;;
     tables _NUMERIC_ / missing missprint nocum nopercent;
     format _NUMERIC_ zmissfmt.;;
     tables _NUMERIC_ / missing missprint nocum nopercent;
run;

data ph.example;

   set ph.example;
   array change _numeric_;
        do over change;
            if change=. then change=0;
        end;
 run ;
```

Results obtained –

The results show that few variables had some missing values. We impute the missing values found by 0, since that is the most suitable and generalized way for handling missing numeric values.

The following table shows the variables that were found to have missing values, and the count of missing observations. It also shows whether the variable is normal or not:

Table 1: Summary of variables present

Variable Number	**No. of Missing Values**	**Percent of Missing values**
X1	0	0
X2	0	0
X3	0	0
X4	0	0
X5	0	0
X6	0	0
X7	0	0
X8	0	0
X9	0	0
X10	0	0
X11	0	0
X12	1	0.47
X13	0	0
X14	0	0
X15	0	0
X16	0	0
X17	0	0
X18	0	0
X19	0	0
X20	1	0.47
X21	0	0
X22	0	0
X23	0	0
X24	0	0
X25	0	0
X26	0	0
X27	0	0
X28	1	0.47
X29	0	0
X30	0	0
X31	0	0
X32	0	0
X33	0	0
X34	0	0

X35	0	0
X36	0	0
X37	0	0
X38	0	0
X39	0	0
X40	0	0
X41	0	0
X42	0	0
X43	0	0
X44	0	0
X45	0	0
X46	0	0
X47	0	0
X48	0	0
X49	0	0
X50	0	0
X51	0	0
X52	0	0
X53	0	0
X54	0	0
X55	0	0
X56	0	0
X57	0	0
X58	9	4.2
X59	11	5.14
X60	14	6.54
X61	0	0
X62	0	0
X63	0	0
X64	0	0
X65	0	0
X66	0	0
X67	0	0
X68	0	0
X69	0	0
X70	1	0.47
X71	0	0
TOTAL:	***38***	***0.25***

Step 4 – Assessing normality of data using PROC UNIVARIATE (using skewness, kurtosis, histogram and probability plots)

After imputing the missing values, we will now analyze the normality of our data. We'll produce more descriptive statics for this and also use Probability plots and histograms.

Code used –

```
proc univariate data=ph.example;
   var x1-x71;

   histogram x1-x71 / normal(mu=est sigma=est);
   inset skewness kurtosis / position=ne;
   probplot x1-x71 / normal(mu=est sigma=est);
   inset skewness kurtosis;
   title 'Descriptive Statistics Using PROC UNIVARIATE';
run;
```

Example plots-

1. **Statistics and plots for a variable depicting normality**

Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure
Variable: x1

Moments			
N	214	Sum Weights	214
Mean	884.275701	Sum Observations	189235
Std Deviation	164.483442	Variance	27054.8026
Skewness	0.10681944	Kurtosis	-0.294342
Uncorrected SS	173098585	Corrected SS	5762672.95
Coeff Variation	18.6009229	Std Error Mean	11.2438566

Basic Statistical Measures			
Location		Variability	
Mean	884.2757	Std Deviation	164.48344
Median	883.1250	Variance	27055
Mode	790.2500	Range	781.62500
		Interquartile Range	216.12500

Note: The mode displayed is the smallest of 5 modes with a count of 2.

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	78.64523	Pr > |t|	<.0001
Sign	M	107	Pr >= |M|	<.0001
Signed Rank	S	11502.5	Pr >= |S|	<.0001

Quantiles (Definition 5)	
Level	Quantile
100% Max	1258.625
99%	1250.500
95%	1166.750
90%	1117.125
75% Q3	989.875
50% Median	883.125
25% Q1	773.750
10%	666.750
5%	617.125
1%	524.750
0% Min	477.000

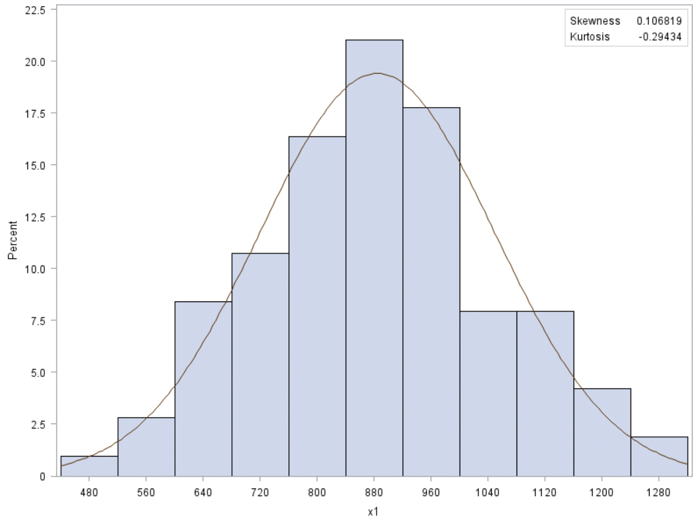
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
477.000	77	1237.50	138
491.625	206	1240.88	199
524.750	203	1250.50	184
546.000	194	1253.25	45
560.750	117	1258.63	2

Page Break

Page Break

Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure

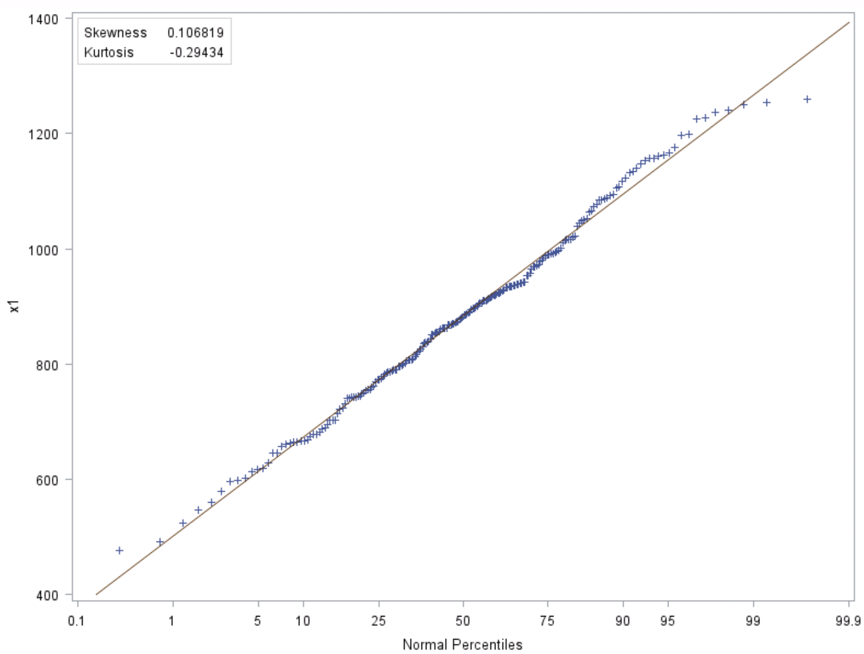


Page Break

Page Break

Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure



Page Break

Descriptive Statistics Using PROC UNIVARIATE

2. Statistics and plots for a variable not depicting normality

Page Break

Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure
Variable: x32

Moments			
N	214	**Sum Weights**	214
Mean	3.11236916	**Sum Observations**	666.047
Std Deviation	3.40100231	**Variance**	11.5668167
Skewness	1.63699396	**Kurtosis**	2.03056079
Uncorrected SS	4536.7161	**Corrected SS**	2463.73196
Coeff Variation	109.273744	**Std Error Mean**	0.23248773

Basic Statistical Measures			
Location		**Variability**	
Mean	3.112369	**Std Deviation**	3.40100
Median	1.225000	**Variance**	11.56682
Mode	0.707000	**Range**	14.34300
		Interquartile Range	3.77100

Tests for Location: Mu0=0				
Test	**Statistic**		**p Value**	
Student's t	**t**	13.38724	**Pr > |t|**	<.0001
Sign	**M**	107	**Pr >= |M|**	<.0001
Signed Rank	**S**	11502.5	**Pr >= |S|**	<.0001

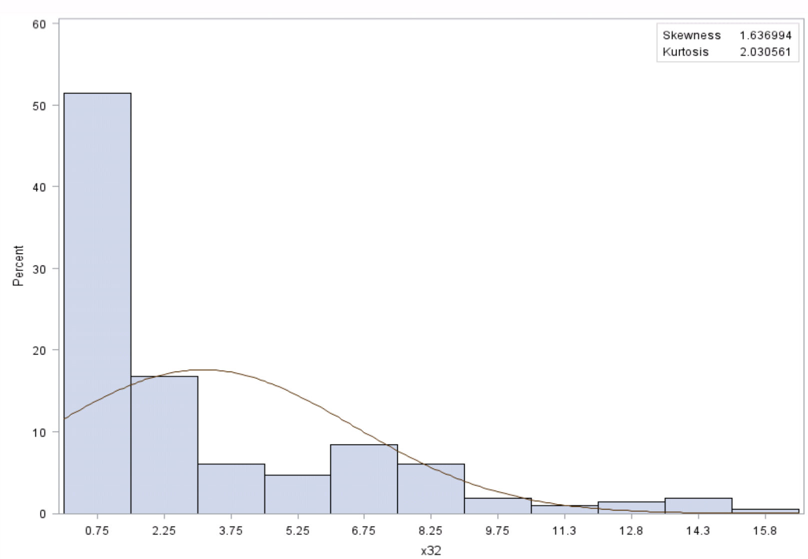
Quantiles (Definition 5)	
Level	**Quantile**
100% Max	15.050
99%	14.169
95%	10.271
90%	8.170
75% Q3	4.637
50% Median	1.225
25% Q1	0.866
10%	0.707
5%	0.707
1%	0.707
0% Min	0.707

Extreme Observations			
Lowest		**Highest**	
Value	**Obs**	**Value**	**Obs**
0.707	214	14.124	74
0.707	213	14.169	107
0.707	209	14.169	204
0.707	207	14.637	125
0.707	203	15.050	131

Page Break

Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure



Page Break

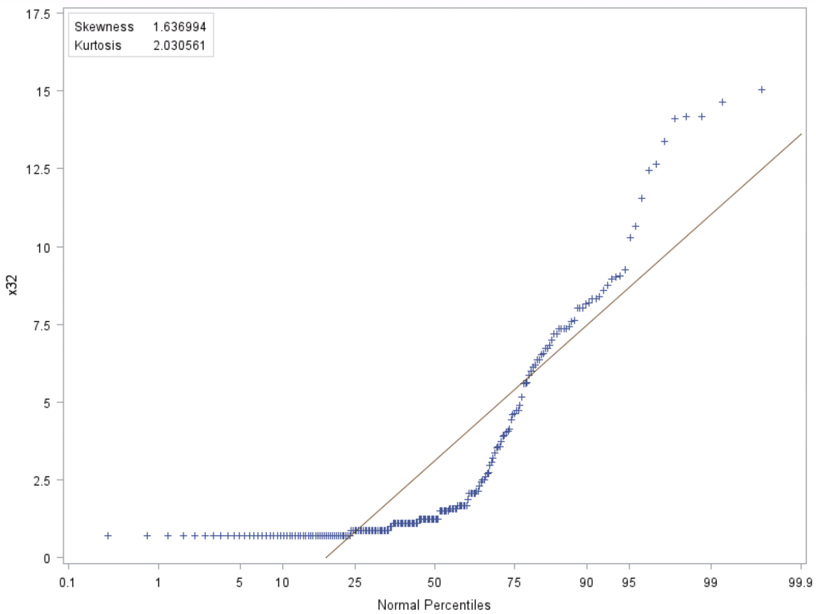
Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure
Fitted Normal Distribution for x32

Page Break

Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure



Page Break

Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure
Variable: x33

<u>Output can be assessed as follows –</u>

Variables with skewness and kurtosis values greater than zero will affect the normality of the data. We have assessed the data using Skewness and Kurtosis for normal data as follows:

1. **For skewness:**
   a. Normal Data = +1 to -1
   b. Right Skewed = value > +1
   c. Left Skewed = value < -1
2. **For Kurtosis:**
   a. Normal Data = exact 3
   b. Leptokurtic = value > 3
   c. Platykurtic = value < 3

We have assessed the plots by manual visualization. Results show the different variations in normality of each variable according to the parameters used by us.

The following table sums up the normality assessment results. The values marked with red show very high values for the properties (exceeding the normal limits):

Table2: Normality Assessment results

Variable	Normality assessment according to Skewness Value	Is the variable normal based on Skewness	Normality assessment according to Kurtosis Value	Is the variable normal based on Kurtosis	Normal Plots
X1	0.1	Normal	-0.29	Not Normal - Platykurtic	Yes
X2	-0.02	Normal	-0.33	Not Normal - Platykurtic	Yes
X3	1.02	Right Skewed	2	Not Normal - Platykurtic	Yes
X	-0.26	Normal	-0.15	Not Normal - Platykurtic	Yes
X5	-0.4	Normal	-0.11	Not Normal - Platykurtic	Yes
X6	-0.32	Normal	-0.08	Not Normal - Platykurtic	Yes
X7	-0.14	Normal	-0.16	Not Normal - Platykurtic	Yes
X8	0.23	Normal	0.16	Not Normal - Platykurtic	Yes
X9	0.35	Normal	0.44	Not Normal - Platykurtic	Yes
X10	0.53	Normal	0.59	Not Normal - Platykurtic	Yes
X11	0.87	Normal	0.38	Not Normal - Platykurtic	Yes
X12	1.13	Right Skewed	0.8	Not Normal - Platykurtic	Yes
X13	1.13	Right Skewed	0.94	Not Normal - Platykurtic	Yes
X14	1.21	Right Skewed	1.21	Not Normal - Platykurtic	Yes
X15	1.2	Right Skewed	1.1	Not Normal - Platykurtic	Yes
X16	0.84	Normal	0.14	Not Normal - Platykurtic	Yes
X17	0.7	Normal	-0.1	Not Normal - Platykurtic	Yes
X18	0.59	Normal	-0.27	Not Normal - Platykurtic	Yes
X19	0.58	Normal	-0.27	Not Normal - Platykurtic	Yes

X20	0.36	Normal	-0.2	Not Normal - Platykurtic	Yes
X21	0.02	Normal	-0.59	Not Normal - Platykurtic	Yes
X22	0.01	Normal	-0.44	Not Normal - Platykurtic	Yes
X23	0	Normal	-0.4	Not Normal - Platykurtic	Yes
X24	0.29	Normal	-0.1	Not Normal - Platykurtic	Yes
X25	0.42	Normal	-0.01	Not Normal - Platykurtic	Yes
X26	0.6	Normal	0.07	Not Normal - Platykurtic	Yes
X27	0.91	Normal	0.04	Not Normal - Platykurtic	Yes
X28	1.15	Right Skewed	0.37	Not Normal - Platykurtic	Yes
X29	1.54	Right Skewed	2.5	Not Normal - Platykurtic	No
X30	1.37	Right Skewed	1.63	Not Normal - Platykurtic	Yes
X31	1.18	Right Skewed	2.97	Not Normal - Platykurtic	No
X32	1.63	Right Skewed	2.03	Not Normal - Platykurtic	Yes
X33	1.1	Right Skewed	0.53	Not Normal - Platykurtic	Yes
X34	1.14	Right Skewed	0.44	Not Normal - Platykurtic	Yes
X35	0.14	Normal	-0.02	Not Normal - Platykurtic	Yes
X36	0.16	Normal	-0.05	Not Normal - Platykurtic	Yes
X37	0.39	Normal	0.13	Not Normal - Platykurtic	Yes
X38	0.78	Normal	1.04	Not Normal - Platykurtic	Yes
X39	1.06	Right Skewed	1.76	Not Normal - Platykurtic	Yes
X40	1.08	Right Skewed	1.63	Not Normal - Platykurtic	Yes
X41	0.88	Normal	1.05	Not Normal - Platykurtic	Yes
X42	0.77	Normal	0.85	Not Normal - Platykurtic	Yes
X43	0.7	Normal	0.36	Not Normal - Platykurtic	Yes
X44	0.93	Normal	1.39	Not Normal - Platykurtic	Yes

X45	1.1	Right Skewed	2.49	Not Normal - Platykurtic	No
X46	1.43	Right Skewed	4.28	Not Normal - Leptokurtic	No
X47	1.51	Right Skewed	4.77	Not Normal - Leptokurtic	No
X48	1.55	Right Skewed	4.81	Not Normal - Leptokurtic	No
X49	1.55	Right Skewed	4.9	Not Normal - Leptokurtic	No
X50	1.4	Right Skewed	4.7	Not Normal - Leptokurtic	No
X51	1.34	Right Skewed	2.63	Not Normal - Platykurtic	No
X52	1.7	Right Skewed	4.7	Not Normal - Leptokurtic	No
X43	0.15	Normal	-0.44	Not Normal - Platykurtic	Yes
X54	2.65	Right Skewed	8.23	Not Normal - Leptokurtic	No
X55	4.21	Right Skewed	23	Not Normal - Leptokurtic	No
X56	1.22	Right Skewed	2.72	Not Normal - Platykurtic	No
X57	1.26	Right Skewed	2.83	Not Normal - Platykurtic	No
X58	1.92	Right Skewed	6.71	Not Normal - Leptokurtic	No
X59	2.7	Right Skewed	13.6	Not Normal - Leptokurtic	No
X60	4.53	Right Skewed	36.28	Not Normal - Leptokurtic	No
X61	-0.37	Normal	0.09	Not Normal - Platykurtic	Yes
X62	0.14	Normal	-0.01	Not Normal - Platykurtic	Yes
X63	-4.68	Left Skewed	25.36	Not Normal - Leptokurtic	No
X64	-0.39	Normal	0.82	Not Normal - Platykurtic	Yes
X65	0.25	Normal	0.03	Not Normal - Platykurtic	Yes
X66	0.45	Normal	0.44	Not Normal - Platykurtic	Yes
X67	0.47	Normal	0.53	Not Normal - Platykurtic	Yes
X68	0.62	Normal	0.78	Not Normal - Platykurtic	Yes
X69	0.88	Normal	0.52	Not Normal - Platykurtic	Yes

X70	1.05	Right Skewed	0.34	Not Normal - Platykurtic	Yes
X71	-0.13	Normal	-0.15	Not Normal - Platykurtic	Yes

Step 5 – Performing Stepwise Variable Selection for the model using different approaches

In this step we use three different procedures to perform variable selection for our **train data**. The procedures used are Proc REG, Proc GLMSELECT and Proc LOGISTIC. We perform the stepwise variable selection using these procedures with two different variation –

a. using default SL values

b. Using manually selected SL values

The following table summarizes the whole analysis if stepwise variable selection methods:

Table3: Stepwise Variable Selection Summary

Stepwise Selection Result (proc REG) Variable included in Final Model		**Stepwise Selection Result (proc GLMSELECT) Variable included in Final Model**		**Stepwise Selection Result (proc LOGISTIC) Variables included in Final Model**
According to Default SL values (SLS & SLE=0.15)	**According to manually selected SL values (SLS & SLE=0.05)**	**According to Default SL values (SLS & SLE = 0.15)**	**According to manually selected SL values (SLS & SLE=0.05)**	**According to SLE & SLS = 0.05 (here this is also the default value)**
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
Yes	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	Yes
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No

Yes	No	Yes	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
Yes	Yes	Yes	Yes	Yes
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No
Yes	Yes	Yes	Yes	Yes
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
Yes	Yes	Yes	Yes	Yes

Yes	No	Yes	No	No
Yes	Yes	Yes	Yes	Yes
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No
No	No	Yes	No	No
No	No	No	No	No
No	No	No	No	No
Yes	Yes	Yes	Yes	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No

Step 6 – Performing Linear Discriminant Analysis using Proc DISCRIM

No after the variable selection, we will apply the LDA using PROC DISCRIM on all different model selections that we got, using different SL values in Proc REG, Proc GLMSELECT and Proc LOGISTIC.

We'll see all these one by one:

1. **LDA on Variables selected using PROC REG – Stepwise selection (SL=0.15)**

Code Used:

```
proc discrim data = ph.example outstat=ph.ldamodel
method=normal pool=yes;
class y;
var x5 x10 x12 x20 x30 x41 x43 x54 x55 x56 x61 x66;
      run;
```

Output:

Page Break

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLE
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y			
From y	0	1	Total
0	83	24	107
	77.57	22.43	100.00
1	22	85	107
	20.56	79.44	100.00
Total	105	109	214
	49.07	50.93	100.00
Priors	0.5	0.5	

Error Count Estimates for y			
	0	1	Total
Rate	0.2243	0.2056	0.2150
Priors	0.5000	0.5000	

The classification table above shows that out of 107 non-permeable observation (y=0) only 83 were correctly classified as non-permeable i.e. **77.57%.** While out of 107 permeable observations (y=1) 85 were correctly classified as permeable i.e. **79.44%.**
So this shows that it is comparatively easier to predict the permeable observations.

2. **LDA on Variables selected using PROC REG – Stepwise selection (SL=0.05)**

Code Used:

```
proc discrim data = ph.example outstat=ph.ldamodel
method=normal pool=yes;
class y;
var x20 x43 x54 x56 x66;
     run;
```

Output:

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLE
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y			
From y	0	1	Total
0	76	31	107
	71.03	28.97	100.00
1	32	75	107
	29.91	70.09	100.00
Total	108	106	214
	50.47	49.53	100.00
Priors	0.5	0.5	

Error Count Estimates for y			
	0	1	Total
Rate	0.2897	0.2991	0.2944
Priors	0.5000	0.5000	

The classification table above shows that out of 107 non-permeable observation (y=0) only 76 were correctly classified as non-permeable i.e. **71.03%**. While out of 107 permeable observations (y=1) 75 were correctly classified as permeable i.e. **70.09%.**
So this shows that it is comparatively easier to predict the non-permeable observations.

3. LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.15)

Code Used:

```
proc discrim data = ph.example outstat=ph.ldamodel
method=normal pool=yes;
class y;
var x10 x12 x20 x30 x41 x43 x54 x55 x56 x61 x63 x66;
     run;
```

Output:

Page Break

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLE
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y

From y	0	1	Total
0	83	24	107
	77.57	22.43	100.00
1	22	85	107
	20.56	79.44	100.00
Total	105	109	214
	49.07	50.93	100.00
Priors	0.5	0.5	

Error Count Estimates for y

	0	1	Total
Rate	0.2243	0.2056	0.2150
Priors	0.5000	0.5000	

The classification table above shows that out of 107 non-permeable observation (y=0) only 83 were correctly classified as non-permeable i.e. **77.57%.** While out of 107 permeable observations (y=1) 85 were correctly classified as permeable i.e. **79.44%.**
So this shows that it is comparatively easier to predict the permeable observations.

4. **<u>LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.05)</u>**

<u>Code Used:</u>

```
proc discrim data = ph.example outstat=ph.ldamodel
method=normal pool=yes;
class y;
var x20 x43 x54 x56 x66;
      run;
```

<u>Output:</u>

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLE
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y			
From y	0	1	Total
0	76	31	107
	71.03	28.97	100.00
1	32	75	107
	29.91	70.09	100.00
Total	108	106	214
	50.47	49.53	100.00
Priors	0.5	0.5	

Error Count Estimates for y			
	0	1	Total
Rate	0.2897	0.2991	0.2944
Priors	0.5000	0.5000	

The classification table above shows that out of 107 non-permeable observation (y=0) only 76 were correctly classified as non-permeable i.e. **71.03%**. While out of 107 permeable observations (y=1) 75 were correctly classified as permeable i.e. **70.09%.** So this shows that it is comparatively easier to predict the non-permeable observations.

5. **<u>LDA on Variables selected using PROC LOGISTIC – Stepwise selection (SL=0.05)</u>**

<u>Code Used:</u>

```
proc discrim data = ph.example outstat=ph.ldamodel
method=normal pool=yes;
class y;
var x8 x20 x43 x54 x56 ;
      run;
```

Output:

Page Break

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLE
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y

From y	0	1	Total
0	74	33	107
	69.16	30.84	100.00
1	30	77	107
	28.04	71.96	100.00
Total	104	110	214
	48.60	51.40	100.00
Priors	0.5	0.5	

Error Count Estimates for y

	0	1	Total
Rate	0.3084	0.2804	0.2944
Priors	0.5000	0.5000	

Page Break

The classification table above shows that out of 107 non-permeable observation (y=0) only 74 were correctly classified as non-permeable i.e. **69.16%**. While out of 107 permeable observations (y=1) 77 were correctly classified as permeable i.e. **71.96%.** So this shows that it is comparatively easier to predict the permeable observations.

Summary of LDA on Training Data:

The following table summarizes the results obtained by performing Linear Discriminant Analysis (using proc discrim):

Table4: Summary of LDA

	Correctly Predicted Permeable observations		**Correctly Predicted Non-Permeable observations**	
	Count	**Percentage**	**Count**	**Percentage**
LDA on Variables selected using PROC REG – Stepwise selection (SL=0.15)	**85**	**79.44%**	**83**	**77.57%**
LDA on Variables selected using PROC REG – Stepwise selection (SL=0.05)	75	70.09%	76	71.03%
LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.15)	**85**	**79.44%**	**83**	**77.57%**
LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.05)	75	70.09%	76	71.03%
LDA on Variables selected using PROC LOGISTIC – Stepwise selection (SL=0.05)	77	71.96%	74	69.16%

Test Data Results

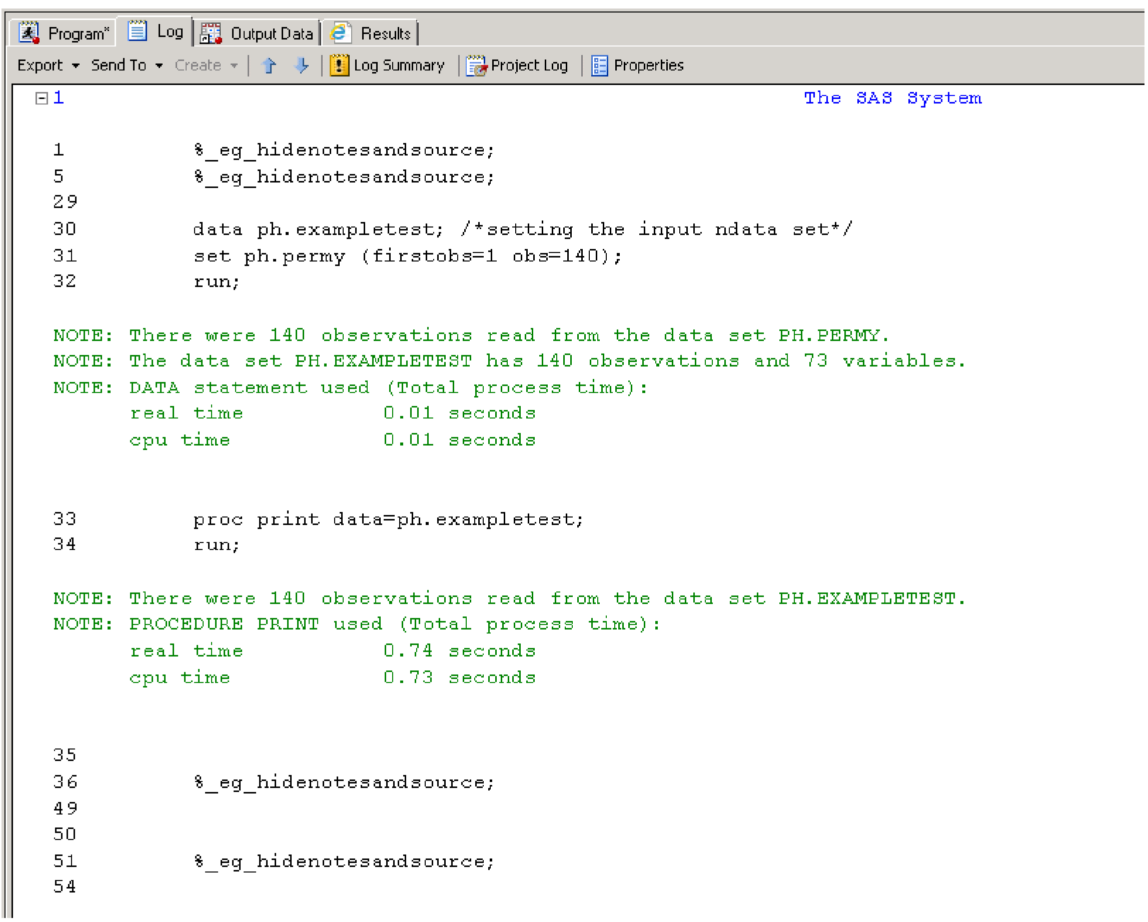
Step 1- Importing the data set into enterprise guide and printing the dataset

We start with importing our dataset into our local SAS server and print the dataset to have a thorough look at our data. We use only the training data set here (observations starting from 141 till 354).

Code used –

```
libname ph 'C:\SAS';
data ph.exampletest; /*setting the input ndata set*/
set ph.permy (firstobs=1 obs=140);
run;
proc print data=ph.exampletest;
run;
```

Output-




```
1                                                          The SAS System

1          %_eg_hidenotesandsource;
5          %_eg_hidenotesandsource;
29
30         data ph.exampletest; /*setting the input ndata set*/
31         set ph.permy (firstobs=1 obs=140);
32         run;

NOTE: There were 140 observations read from the data set PH.PERMY.
NOTE: The data set PH.EXAMPLETEST has 140 observations and 73 variables.
NOTE: DATA statement used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds


33         proc print data=ph.exampletest;
34         run;

NOTE: There were 140 observations read from the data set PH.EXAMPLETEST.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.74 seconds
      cpu time            0.73 seconds


35
36         %_eg_hidenotesandsource;
49
50
51         %_eg_hidenotesandsource;
54
```

Step 2 – Using The Means procedure to calculate descriptive statistics for the dataset

In the next step we are using the MEANS procedure to calculate all the descriptive statistics like mean, median, interquartile ranges etc. This helps us to analyze our dataset better and know the distribution of our data.

Code used –

```
proc means data=ph.exampletest printalltypes n mean median std min max
q1 q3;
class y;
var x1-x10;
run;
```

Output-

The MEANS Procedure

N Obs	Variable	N	Mean	Median	Std Dev	Minimum	Maximum	Lower Quartile	Upper Quartile
140	x1	140	900.2767857	895.4375000	161.2828830	491.8750000	1246.00	799.0625000	991.2500000
	x2	140	610.5786143	605.3705000	100.3121190	354.3120000	849.6270000	550.0545000	665.5465000
	x3	140	1.4711714	1.4670000	0.0413093	1.3770000	1.6250000	1.4465000	1.4885000
	x4	140	1.5733571	1.5730000	0.1168610	1.2530000	1.8920000	1.5090000	1.6580000
	x5	140	1591.96	1583.06	223.8627588	992.7500000	2284.63	1448.56	1753.50
	x6	140	1056.31	1042.50	195.7568223	521.0000000	1593.25	906.1250000	1196.44
	x7	140	618.9910714	623.2500000	145.7353483	220.5000000	976.2500000	513.0625000	715.0000000
	x8	140	262.7285714	264.6875000	73.7260172	53.1250000	438.5000000	218.0625000	305.4375000
	x9	140	136.3205357	139.5625000	43.1237168	17.5000000	240.0000000	103.7500000	161.4375000
	x10	140	71.9383929	72.3750000	25.3170520	10.0000000	143.7500000	54.3750000	85.8125000

y	N Obs	Variable	N	Mean	Median	Std Dev	Minimum	Maximum	Lower Quartile	Upper Quartile
0	70	x1	70	936.5196429	916.0625000	151.2451335	491.8750000	1240.38	849.6250000	1033.13
		x2	70	636.7631143	626.6825000	94.7065757	354.3120000	849.6270000	581.8600000	700.7240000
		x3	70	1.4681143	1.4625000	0.0363691	1.3880000	1.6040000	1.4460000	1.4820000
		x4	70	1.6111143	1.6000000	0.1088954	1.2750000	1.8920000	1.5410000	1.6860000
		x5	70	1678.15	1683.88	209.4302621	1068.25	2284.63	1535.63	1815.88
		x6	70	1146.37	1154.00	182.8423537	729.1250000	1593.25	1025.25	1272.00
		x7	70	690.2142857	683.3750000	131.1503489	360.3750000	976.2500000	616.2500000	788.7500000
		x8	70	299.4428571	292.1875000	67.5448713	116.0000000	438.5000000	258.5000000	345.2500000
		x9	70	156.2375000	151.5625000	40.5891126	46.5000000	240.0000000	134.8750000	176.5000000
		x10	70	82.5678571	83.3125000	24.9289542	23.6250000	143.7500000	65.1250000	97.6250000
1	70	x1	70	864.0339286	850.1875000	163.8934748	554.2500000	1246.00	771.1250000	962.6250000
		x2	70	584.3941143	576.6930000	99.5510485	394.8380000	824.1350000	535.0140000	646.7780000
		x3	70	1.4742286	1.4705000	0.0457818	1.3770000	1.6250000	1.4470000	1.4990000
		x4	70	1.5356000	1.5375000	0.1129609	1.2530000	1.8300000	1.4820000	1.5950000
		x5	70	1505.77	1508.00	204.9899707	992.7500000	1910.63	1385.75	1641.50
		x6	70	966.2553571	945.2500000	165.2549612	521.0000000	1366.25	876.1250000	1085.25
		x7	70	547.7678571	546.4375000	123.6626435	220.5000000	802.7500000	475.3750000	635.2500000
		x8	70	226.0142857	235.6250000	60.4364880	53.1250000	366.6250000	177.8750000	269.2500000
		x9	70	116.4035714	122.5000000	35.9710480	17.5000000	194.8750000	87.1250000	142.3750000
		x10	70	61.3089286	62.4375000	20.9880133	10.0000000	105.6250000	43.3750000	78.2500000

Note: the output above is just for first 10 variables, but the results are performed for all 71 in the same manner.

Step 3 – Finding the number of missing values and imputing them with 0

Before performing any operations on our data, it's important to normalize and cleanse our data. Determining missing values and resolving them is a part of the same process.

Code Used-

```
proc format; /* create a format to group missing and nonmissing */
     value $amissfmt ' '='Missing' other='Not Missing';
     value  bmissfmt  . ='Missing' other='Not Missing';
 value cmissfmt 0='Missing' other = 'Not Missing';
run;

proc freq data=ph.exampletest;
format _CHAR_ $amissfmt.;
tables _CHAR_ / missing missprint nocum nopercent;
format _NUMERIC_ bmissfmt.;;
tables _NUMERIC_ / missing missprint nocum nopercent;
format _NUMERIC_ cmissfmt.;;
tables _NUMERIC_ / missing missprint nocum nopercent;
run;

data ph.exampletest;
   set ph.exampletest;
   array change _numeric_;
        do over change;
            if change=. then change=0;
        end;
 run;
```

Results obtained –

The results show that few variables had some missing values. We impute the missing values found by 0, since that is the most suitable and generalized way for handling missing numeric values.

The following table shows the variables that were found to have missing values, and the count of missing observations. It also shows whether the variable is normal or not:

Table 5: Summary of variables present

Variable Number	No. of Missing Values	Percent of Missing values
X1	0	0
X2	0	0
X3	0	0
X4	0	0
X5	0	0
X6	0	0
X7	0	0
X8	0	0
X9	0	0
X10	0	0
X11	0	0
X12	1	0.71
X13	0	0
X14	0	0
X15	0	0
X16	0	0
X17	0	0
X18	0	0
X19	0	0
X20	1	0.71
X21	0	0
X22	0	0
X23	0	0
X24	0	0
X25	0	0
X26	0	0
X27	0	0
X28	1	0.71
X29	0	0
X30	0	0
X31	0	0
X32	0	0
X33	0	0
X34	0	0

X35	0	0
X36	0	0
X37	0	0
X38	0	0
X39	0	0
X40	0	0
X41	0	0
X42	0	0
X43	0	0
X44	0	0
X45	0	0
X46	0	0
X47	0	0
X48	0	0
X49	0	0
X50	0	0
X51	0	0
X52	0	0
X53	0	0
X54	0	0
X55	0	0
X56	0	0
X57	0	0
X58	6	4.3
X59	9	6.42
X60	9	6.42
X61	0	0
X62	0	0
X63	0	0
X64	0	0
X65	0	0
X66	0	0
X67	0	0
X68	0	0
X69	0	0
X70	1	0.71
X71	0	0
TOTAL:	***28***	***0.28%***

Step 4 – Assessing normality of data using PROC UNIVARIATE (using skewness, kurtosis, histogram and probability plots)

After imputing the missing values, we will now analyze the normality of our data. We'll produce more descriptive statics for this and also use Probability plots and histograms.

Code used –

```
proc univariate data=ph.exampletest;
   var x1-x71;

   histogram x1-x71 / normal(mu=est sigma=est);
   inset skewness kurtosis / position=ne;
   probplot x1-x71 / normal(mu=est sigma=est);
   inset skewness kurtosis;
   title 'Descriptive Statistics Using PROC UNIVARIATE';
run;
```

Example plots-

3. **Statistics and plots for a variable depicting normality**

Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure
Variable: x53

Moments			
N	140	Sum Weights	140
Mean	19.07595	Sum Observations	2670.633
Std Deviation	2.03676743	Variance	4.14842154
Skewness	0.01267187	Kurtosis	-0.1605304
Uncorrected SS	51521.4922	Corrected SS	576.630595
Coeff Variation	10.6771481	Std Error Mean	0.17213827

Basic Statistical Measures			
Location		Variability	
Mean	19.07595	Std Deviation	2.03677
Median	19.32000	Variance	4.14842
Mode	17.62000	Range	10.06800
		Interquartile Range	2.61400

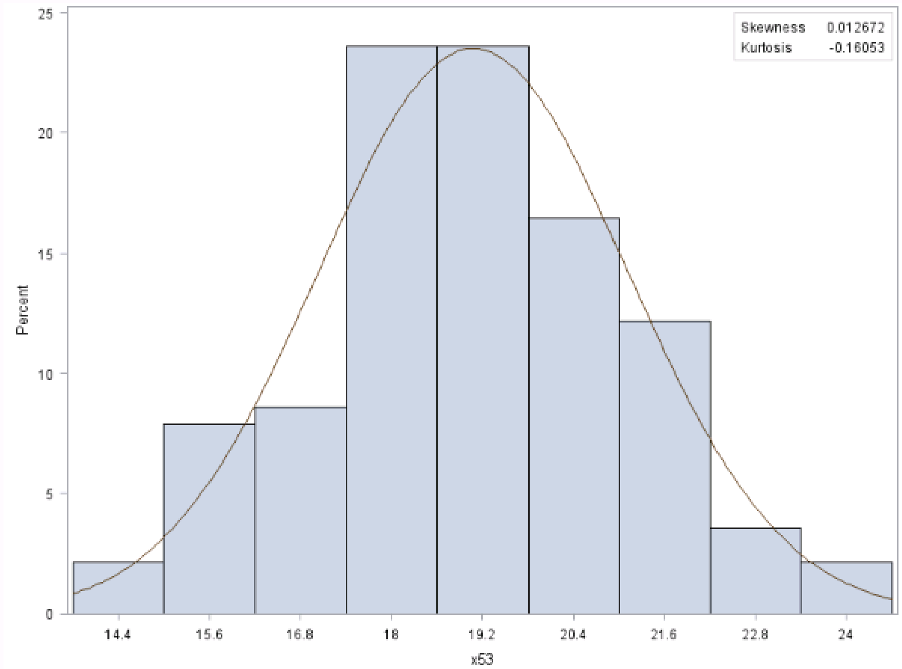
Note: The mode displayed is the smallest of 3 modes with a count of 2.

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	110.8176	Pr > |t|	<.0001
Sign	M	70	Pr >= |M|	<.0001
Signed Rank	S	4935	Pr >= |S|	<.0001

Quantiles (Definition 5)	
Level	Quantile
100% Max	24.1770
99%	24.0260
95%	22.3575
90%	21.8145
75% Q3	20.3040
50% Median	19.3200
25% Q1	17.6900
10%	16.2195
5%	15.5545
1%	14.4410
0% Min	14.1090

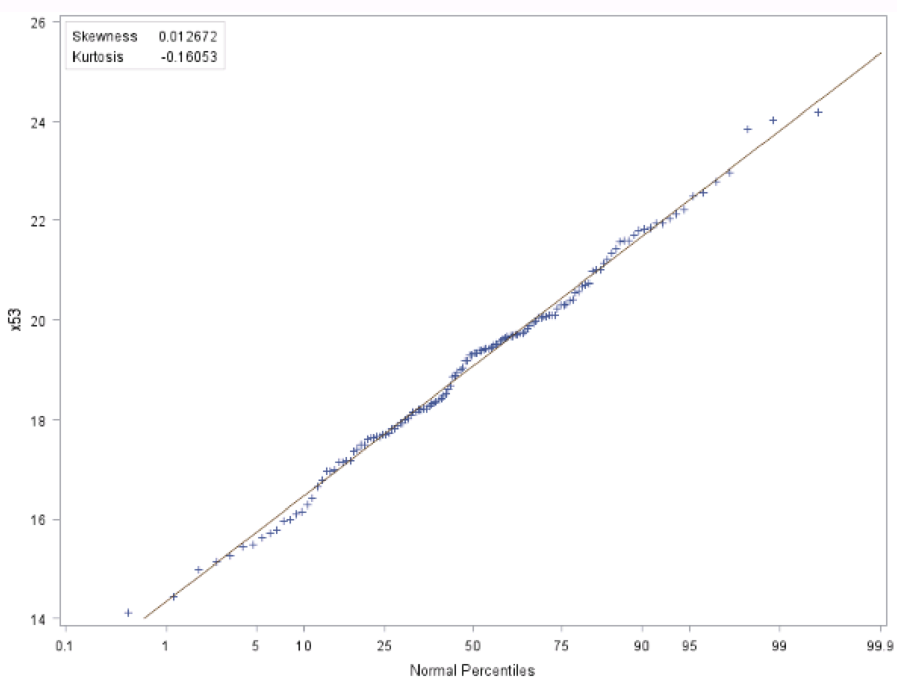
Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure



Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure



4. Statistics and plots for a variable not depicting normality

Descriptive Statistics Using PROC UNIVARIATE

The UNIVARIATE Procedure
Variable: x63

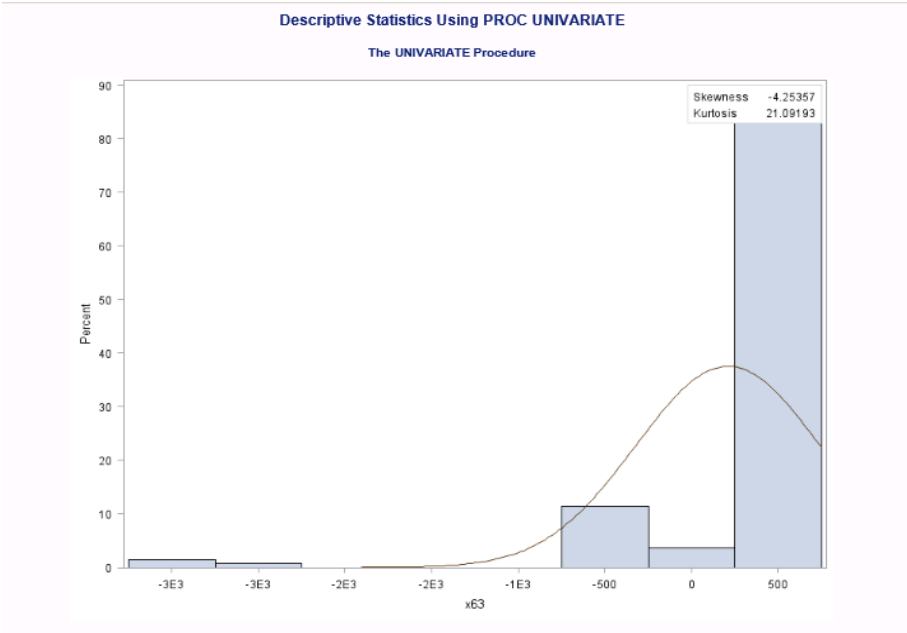
Moments			
N	140	Sum Weights	140
Mean	210.617857	Sum Observations	29486.5
Std Deviation	531.207944	Variance	282181.88
Skewness	-4.2535696	Kurtosis	21.0919261
Uncorrected SS	45433664.7	Corrected SS	39223281.3
Coeff Variation	252.214105	Std Error Mean	44.8952654

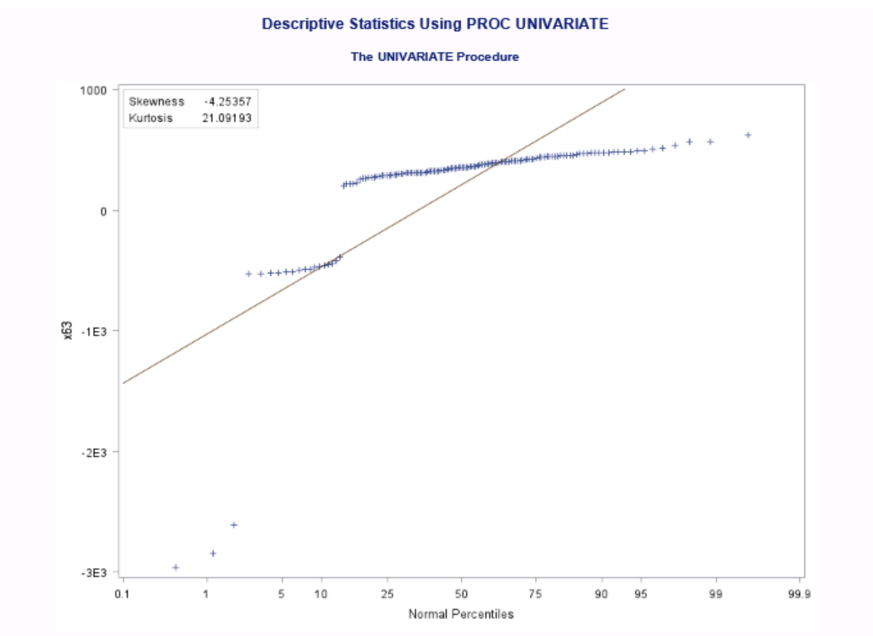
Basic Statistical Measures			
Location		Variability	
Mean	210.6179	Std Deviation	531.20794
Median	351.6875	Variance	282182
Mode	311.0000	Range	3583
		Interquartile Range	140.12500

Note: The mode displayed is the smallest of 3 modes with a count of 2.

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	4.691316	Pr > |t|	<.0001
Sign	M	51	Pr >= |M|	<.0001
Signed Rank	S	2691	Pr >= |S|	<.0001

Quantiles (Definition 5)	
Level	Quantile
100% Max	619.125
99%	564.750
95%	490.125
90%	475.250
75% Q3	426.875
50% Median	351.688
25% Q1	286.750
10%	-461.688
5%	-513.625
1%	-2844.500
0% Min	-2963.875





Output can be assessed as follows –

Variables with skewness and kurtosis values greater than zero will affect the normality of the data. We have assessed the data using Skewness and Kurtosis for normal data as follows:

3. **For skewness:**
    d. Normal Data = +1 to -1
    e. Right Skewed = value > +1
    f. Left Skewed = value < -1

4. **For Kurtosis:**
   d. Normal Data = exact 3
   e. Leptokurtic = value > 3
   f. Platykurtic = value < 3

We have assessed the plots by manual visualization. Results show the different variations in normality of each variable according to the parameters used by us.

The following table sums up the normality assessment results. The values marked with red show very high values for the properties (exceeding the normal limits):

Table6: Normality Assessment results

Variable Number	Normality assessment according to Skewness Value	Is the variable normal based on Skewness	Normality assessment acccoriding to Kurtosis Value	Is the variable normal based on Kurtosis	Normal Plots
X1	0.07	Normal	-0.07	Not Normal - Platykurtic	Yes
X2	0	Normal	-0.04	Not Normal - Platykurtic	Yes
X3	0.85	Normal	1.7	Not Normal - Platykurtic	Yes
X4	-0.18	Normal	0.31	Not Normal - Platykurtic	Yes
X5	-0.07	Normal	0.25	Not Normal - Platykurtic	Yes
X6	0.1	Normal	-0.3	Not Normal - Platykurtic	Yes
X7	-0.06	Normal	-0.3	Not Normal - Platykurtic	Yes
X8	0.02	Normal	0.002	Not Normal - Platykurtic	Yes
X9	0.82	Normal	0.1	Not Normal - Platykurtic	Yes
X10	0.32	Normal	0.26	Not Normal - Platykurtic	Yes
X11	0.79	Normal	0.72	Not Normal - Platykurtic	Yes
X12	1.4	Right Skewed	2.3	Not Normal - Platykurtic	No
X13	1.03	Right Skewed	0.5	Not Normal - Platykurtic	Yes
X14	1.06	Right Skewed	0.73	Not Normal - Platykurtic	Yes

X15	1.13	Right Skewed	0.91	Not Normal - Platykurtic	No
X16	0.8	Normal	0.05	Not Normal - Platykurtic	Yes
X17	0.75	Normal	-0.15	Not Normal - Platykurtic	Yes
X18	0.69	Normal	-0.41	Not Normal - Platykurtic	Yes
X19	0.7	Normal	-0.14	Not Normal - Platykurtic	Yes
X20	0.9	Normal	0.9	Not Normal - Platykurtic	Yes
X21	0.08	Normal	-0.59	Not Normal - Platykurtic	Yes
X22	0.11	Normal	-0.48	Not Normal - Platykurtic	Yes
X23	-0.07	Normal	-0.45	Not Normal - Platykurtic	Yes
X24	-0.15	Normal	-0.15	Not Normal - Platykurtic	Yes
X25	-0.15	Normal	-0.07	Not Normal - Platykurtic	Yes
X26	0.27	Normal	-0.13	Not Normal - Platykurtic	Yes
X27	0.7	Normal	-0.16	Not Normal - Platykurtic	Yes
X28	1.08	Right Skewed	0.5	Not Normal - Platykurtic	No
X29	1.7	Right Skewed	3.11	Not Normal - Leptokurtic	Yes
X30	1.67	Right Skewed	1.44	Not Normal - Platykurtic	Yes
X31	1.45	Right Skewed	3.42	Not Normal - Leptokurtic	Yes
X32	1.48	Right Skewed	1.32	Not Normal - Platykurtic	No
X33	1.22	Right Skewed	0.74	Not Normal - Platykurtic	No
X34	1.5	Right Skewed	1.57	Not Normal - Platykurtic	No
X35	0.35	Normal	-0.32	Not Normal - Platykurtic	Yes
X36	0.27	Normal	-0.48	Not Normal - Platykurtic	Yes
X37	0.23	Normal	-0.35	Not Normal - Platykurtic	Yes
X38	0.39	Normal	0.19	Not Normal - Platykurtic	Yes
X39	0.53	Normal	0.27	Not Normal - Platykurtic	Yes

X40	0.6	Normal	0.5	Not Normal - Platykurtic	Yes
X41	0.37	Normal	-0.21	Not Normal - Platykurtic	Yes
X42	0.33	Normal	-0.43	Not Normal - Platykurtic	Yes
X43	0.6	Normal	-0.41	Not Normal - Platykurtic	Yes
X44	0.78	Normal	0.09	Not Normal - Platykurtic	Yes
X45	1.05	Right Skewed	1.01	Not Normal - Platykurtic	Yes
X46	1.22	Right Skewed	1.49	Not Normal - Platykurtic	Yes
X47	1.3	Right Skewed	1.77	Not Normal - Platykurtic	Yes
X48	1.27	Right Skewed	1.53	Not Normal - Platykurtic	Yes
X49	1.21	Right Skewed	1.55	Not Normal - Platykurtic	Yes
X50	1.17	Right Skewed	1.49	Not Normal - Platykurtic	Yes
X51	0.8	Normal	0.5	Not Normal - Platykurtic	Yes
X52	1.14	Right Skewed	1.24	Not Normal - Platykurtic	Yes
X53	0.01	Normal	-0.16	Not Normal - Platykurtic	Yes
X54	2.72	Right Skewed	10.72	Not Normal - Leptokurtic	No
X55	4	Right Skewed	20.1	Not Normal - Leptokurtic	No
X56	1.35	Right Skewed	2.75	Not Normal - Platykurtic	Yes
X57	0.98	Normal	1.14	Not Normal - Platykurtic	Yes
X58	0.88	Normal	0.43	Not Normal - Platykurtic	Yes
X59	0.94	Normal	0.48	Not Normal - Platykurtic	Yes
X60	1.6	Right Skewed	4.03	Not Normal - Leptokurtic	Yes
X61	0.28	Normal	0.87	Not Normal - Platykurtic	Yes
X62	0.11	Normal	0.32	Not Normal - Platykurtic	Yes
X63	-4.2	Left Skewed	21.09	Not Normal - Leptokurtic	No
X64	-0.38	Normal	0.09	Not Normal - Platykurtic	Yes

X65	0.08	Normal	-0.72	Not Normal - Platykurtic	Yes
X66	0	Normal	-0.22	Not Normal - Platykurtic	Yes
X67	0.09	Normal	0.05	Not Normal - Platykurtic	Yes
X68	0.38	Normal	0.31	Not Normal - Platykurtic	Yes
X69	0.85	Normal	0.84	Not Normal - Platykurtic	Yes
X70	1.35	Right Skewed	2.21	Not Normal - Platykurtic	Yes
X71	0.18	Normal	0.5	Not Normal - Platykurtic	Yes

Step 5 – Performing Stepwise Variable Selection for the model using different approaches

In this step we use three different procedures to perform variable selection for our **test data**. The procedures used are Proc REG, Proc GLMSELECT and Proc LOGISTIC. We perform the stepwise variable selection using these procedures with two different variation –

a. using default SL values

b. Using manually selected SL values

The following table summarizes the whole analysis if stepwise variable selection methods:

Table7: Stepwise Variable Selection Summary

Stepwise Selection Result (proc REG) Variable included in Final Model		**Stepwise Selection Result (proc GLMSELECT) Variable included in Final Model**		**Stepwise Selection Result (proc LOG,.ISTIC) Variables included in Final Model**
According to Default SL values (SLS & SLE=0.15)	**According to manually selected SL values (SLS & SLE=0.05)**	**According to Default SL values (SLS & SLE = 0.15)**	**According to manually selected SL values (SLS & SLE=0.05)**	**According to SLE & SLS = 0.05 (here this is default value itself)**
No	No	No	No	No
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No

No	No	No	No	No
Yes	Yes	Yes	Yes	Yes
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No

No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No
No	No	No	No	No

Step 6 – Performing Linear Discriminant Analysis using Proc DISCRIM

No after the variable selection, we will apply the LDA using PROC DISCRIM on all different model selections that we got, using different SL values in Proc REG, Proc GLMSELECT and Proc LOGISTIC.

We'll see all these one by one:

1. **LDA on Variables selected using PROC REG – Stepwise selection (SL=0.15)**

   Code Used:

```
proc discrim data = ph.exampletest outstat=ph.ldamodeltest
method=normal pool=yes;
class y;
var x3 x8 x20 x31 x34 x62;
      run;
```

Output:

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLETEST
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y

From y	0	1	Total
0	53 75.71	17 24.29	70 100.00
1	17 24.29	53 75.71	70 100.00
Total	70 50.00	70 50.00	140 100.00
Priors	0.5	0.5	

Error Count Estimates for y

	0	1	Total
Rate	0.2429	0.2429	0.2429
Priors	0.5000	0.5000	

The classification table above shows that out of 70 non-permeable observation (y=0) 53 were correctly classified as non-permeable i.e. **75.71%.** Similarly, out of 70 permeable observations (y=1) 53 were correctly classified as permeable i.e. **75.71%.**
So this shows that by this method the probability for predicting the permeable and non-permeable observations is equal.

2. LDA on Variables selected using PROC REG – Stepwise selection (SL=0.05)

Code Used:

```
proc discrim data = ph.exampletest outstat=ph.ldamodeltest
method=normal pool=yes;
class y;
var x8;
      run;
```

Output:

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLETEST
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y

From y	0	1	Total
0	51 72.86	19 27.14	70 100.00
1	21 30.00	49 70.00	70 100.00
Total	72 51.43	68 48.57	140 100.00
Priors	0.5	0.5	

Error Count Estimates for y

	0	1	Total
Rate	0.2714	0.3000	0.2857
Priors	0.5000	0.5000	

The classification table above shows that out of 70 non-permeable observation (y=0) 51 were correctly classified as non-permeable i.e. **72.85%**. While out of 70 permeable observations (y=1) 49 were correctly classified as permeable i.e. **70%.**
So this shows that it is comparatively easier to predict the non-permeable observations.

3. LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.15)

Code Used:

```
proc discrim data = ph.exampletest outstat=ph.ldamodeltest
method=normal pool=yes;
class y;
var x3 x8 x20 x31 x34 x62;
      run;
```

Output:

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLETEST
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y

From y	0	1	Total
0	53 75.71	17 24.29	70 100.00
1	17 24.29	53 75.71	70 100.00
Total	70 50.00	70 50.00	140 100.00
Priors	0.5	0.5	

Error Count Estimates for y

	0	1	Total
Rate	0.2429	0.2429	0.2429
Priors	0.5000	0.5000	

The classification table above shows that out of 70 non-permeable observation (y=0) 53 were correctly classified as non-permeable i.e. **75.71%.** Similarly, out of 70 permeable observations (y=1) 53 were correctly classified as permeable i.e. **75.71%.**
So this shows that by this method the probability for predicting the permeable and non-permeable observations is equal.

4. LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.05)

Code Used:

```
proc discrim data = ph.exampletest outstat=ph.ldamodeltest
method=normal pool=yes;
class y;
      var x8;
      run;
```

Output:

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLETEST
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y

From y	0	1	Total
0	51 72.86	19 27.14	70 100.00
1	21 30.00	49 70.00	70 100.00
Total	72 51.43	68 48.57	140 100.00
Priors	0.5	0.5	

Error Count Estimates for y

	0	1	Total
Rate	0.2714	0.3000	0.2857
Priors	0.5000	0.5000	

The classification table above shows that out of 70 non-permeable observation (y=0) 51 were correctly classified as non-permeable i.e. **72.85%**. While out of 70 permeable observations (y=1) 49 were correctly classified as permeable i.e. **70%.**

So this shows that it is comparatively easier to predict the non-permeable observations.

5. LDA on Variables selected using PROC LOGISTIC – Stepwise selection (SL=0.05)

Code Used:

```
proc discrim data = ph.exampletest outstat=ph.ldamodeltest
method=normal pool=yes;
class y;
      var x8;
      run;
```

Output:

Descriptive Statistics Using PROC UNIVARIATE

The DISCRIM Procedure
Classification Summary for Calibration Data: PH.EXAMPLETEST
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into y			
From y	0	1	Total
0	51 72.86	19 27.14	70 100.00
1	21 30.00	49 70.00	70 100.00
Total	72 51.43	68 48.57	140 100.00
Priors	0.5	0.5	

Error Count Estimates for y			
	0	1	Total
Rate	0.2714	0.3000	0.2857
Priors	0.5000	0.5000	

The classification table above shows that out of 70 non-permeable observation (y=0) 51 were correctly classified as non-permeable i.e. **72.85%**. While out of 70 permeable observations (y=1) 49 were correctly classified as permeable i.e. **70%.**
So this shows that it is comparatively easier to predict the non-permeable observations.

Summary of LDA on Testing Data:

The following table summarizes the results obtained by performing Linear Discriminant Analysis (using proc discrim):

Table8: Summary of LDA

	Correctly Predicted Permeable observations		**Correctly Predicted Non-Permeable observations**	
	Count	**Percentage**	**Count**	**Percentage**
LDA on Variables selected using PROC REG – Stepwise selection (SL=0.15)	**53**	**75.71%**	**53**	**75.71%**
LDA on Variables selected using PROC REG – Stepwise selection (SL=0.05)	49	70.00%	51	72.85%
LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.15)	**53**	**75.71%**	**53**	**75.71%**
LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.05)	49	70.00%	51	72.85%
LDA on Variables selected using PROC LOGISTIC – Stepwise selection (SL=0.05)	49	70.00%	51	72.85%

Comparison of Results of LDA on Test and Train Data:

Method	**Train data Efficiency**		**Test Data Efficiency**	
	Correctly Predicted Permeable	**Correctly Predicted Non-Permeable**	**Correctly Predicted Permeable**	**Correctly Predicted Non-Permeable**
LDA on Variables selected using PROC REG – Stepwise selection (SL=0.15)	79.44%	77.57%	75.71%	75.71%

LDA on Variables selected using PROC REG – Stepwise selection (SL=0.05)	70.09%	71.03%	70.00%	72.85%
LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.15)	79.44%	77.57%	75.71%	75.71%
LDA on Variables selected using PROC GLMSELECT – Stepwise selection (SL=0.05)	70.09%	71.03%	70.00%	72.85%
LDA on Variables selected using PROC LOGISTIC – Stepwise selection (SL=0.05)	71.96%	69.16%	70.00%	72.85%

On an average, the highest efficiency obtained on Train data using LDA for permeable observation is **79.44%** and for correctly predicted non permeable observation is **77.57%.**

While on test data, the highest efficiency obtained for both correctly predicted permeable and non-permeable observation using LDA is **75.71%.**

The following bar graph shows these results graphically-