# Chapter 7 Exploratory Data Analysis

## 7.1 Introduction

In Section 1.2 data exploration, or Exploratory Data Analysis (EDA) was listed as one of the fundamental steps of the data mining process. EDA is a set of procedures aimed at understanding the data and the relationships among variables. It is usually performed interactively through the calculation of many measures and summary tables, as well as by using graphical representation of the data. Most data mining modeling packages have extensive EDA capabilities. In addition, there are many software packages dedicated to performing some EDA functions, such as SAS/Insight.

This chapter examines how EDA and procedures of data preparation influence each other. Specifically, in order to perform EDA effectively, the data need to be prepared for that purpose. Conversely, EDA results will suggest that certain transformations should be performed on the data to obtain better models and to reveal hidden structures. The methods presented in this chapter follow these two lines of thought.

It should be emphasized that the procedures presented here are supplementary to the graphical and interactive methods of EDA. Interactive graphical analysis of the data is probably the most important aspect of EDA. We only cover the procedures involving calculations that could be automated using SAS macros. Be warned, then, that even implementing *all* the macros presented or discussed in this chapter will not be enough to render a *complete* EDA.

## 7.2 Common EDA Procedures

EDA was proposed, and so named, by Tukey (1977). Current practices in EDA involve the following procedures.

- Descriptive statistics—This involves the examination of the univariate statistics of each variable.

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-11-08 15:22:15.

- Examination of the distribution of each variable (for all variable types) and testing normality assumptions (for continuous variables only).
- Calculation of the coefficients of correlation between the variables (for ordinal and continuous variables only).
- Detection of outliers.
- Exploration of nonlinear relationships among different variables using scatter plots.
- Counting missing values for each variable and examining the effect of the bias introduced by simple substitution schemes, such as substituting the mean or the median, or by deleting these observations.
- Testing the assumptions that different samples were made from the same population.
- Extracting and examining cross-tabulated frequencies of variables.
- Calculation of different statistics on cross-tabulation tables.
- Using multivariate methods, such as factor analysis and principal component analysis, to detect linear structures within the data.

Many of these steps can be performed by dedicated EDA or data mining modeling packages. However, performing them as part of the data preparation process could save some time and automate the process to reduce the interactive effort performed by the analyst.

The interaction of data preparation procedures and the results of EDA is evident from the preceding list. For example, the option of using a simple substitution scheme to treat missing values, and the consequent bias in the data, will have a direct impact on preparing the mining view and the scoring view. Similarly, examining the distribution of each variable (e.g., through histograms) will reveal the necessity of altering the distribution of some of the variables using data transformations.

We will start with macros that implement some of these operations to automate some of the EDA steps.

## 7.3 Univariate Statistics

The calculations of the univariate statistics were introduced in Section 6.5 through the macro `VarUnivar()`. The purpose there was to check these measures and compare them to those of the population using the macro `CVLimits()`. In the case of EDA, the univariate measures are calculated for their own sake. We are interested in calculating these measures to assess the different variables. The macro `VarUnivar()` implemented a wrapper on `PROC UNIVARIATE`. `PROC UNIVARIATE` offers another important feature that is useful in EDA, that is, the identification of extreme observations.

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-11-08 15:22:15.

However, PROC UNIVARIATE does not provide an easy way to store these values in a dataset to facilitate further processing on them. The following code generates a dataset with 32 numbers between 0 and 4.0, in addition to four high values between 22 and 34. The number of extreme values, at either the lower limit or the upper limit, printed by PROC UNIVARIATE is determined by the parameter NEXTROBS, which takes a default value of 5.

```
data Extremes;
 input x @@;
 datalines;
 1.45  0.73  2.43  3.89
 3.86  3.96  2.41  2.29
 2.23  2.19  0.37  2.71
 0.77  0.83  3.61  1.71
 1.06  3.23  0.68  3.15
 1.83  3.37  1.60  1.17
 3.87  2.36  1.84  1.64
 3.97  2.23  2.21  1.93
22.00 33.00 23.00 34.00
;
run;
proc univariate data=extremes nextrobs=6;
 var x;
run;
```

The following is the part of the output where the extreme values are printed.

```
       The UNIVARIATE Procedure
             Variable:  x

         Extreme Observations

----Lowest----        ----Highest----

Value      Obs         Value      Obs

 0.37       11          3.96        6
 0.68       19          3.97       29
 0.73        2         22.00       33
 0.77       13         23.00       35
 0.83       14         33.00       34
 1.06       17         34.00       36
```

Note that because we have chosen to print six extreme values on either side of the variable distribution, some the values identified by PROC MEANS are not really of interest because they do not represent the inserted high values.

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

This shows that using `PROC UNIVARIATE` in this simplified manner would not properly identify *interesting* extreme values, that is, *outliers*, with respect to this particular variable. The reason is that `PROC UNIVARIATE` simply sorts the values and finds the `NEXTROBS` lowest and highest values. However, it does not test these observations to confirm that they are outliers. We discuss outliers in more details in Section 7.5.

## 7.4 Variable Distribution

In addition to examination of the univariate statistics, the distribution of each variable usually reveals any interesting characteristics or potential problems. Continuous variables are usually represented using histograms, and nominal and ordinal variables are represented using pie or bar charts.

Plotting histograms is often done by special software or by preparing the data such that the histogram data is based on the binned values. The binning of continuous variables is discussed in full detail in Section 10.3. Examination of the histogram of continuous variables could reveal that it is skewed to either end of the scale. In this case, the techniques described in Section 9.6 could be used to change the shape of the variable distribution.

In the case of nominal and ordinal variables, pie charts often show two potential issues. The first is that when one of the categories is dominating the variable distribution, with all the other categories having small frequencies. In this case, it may be worthwhile to group the *other* categories into a new group. The second situation is that there are a large number of categories in a nominal variable, high cardinality. The techniques described in Section 10.2 could be employed to remedy this problem.

## 7.5 Detection of Outliers

A basic assumption in data mining is that the data used for modeling, and later for scoring, is obtained from (or generated by) a process that has a specific (but unknown) mechanism (a functional form). Therefore, observations that seem not to follow this mechanism are defined to be outliers. Outliers can be a result of having errors in the data or having observations that do not comply with the hidden (or assumed) mechanism that generated the data. Since the mechanism that generated the data is unknown, the definition of outliers is subjective and speculative at best. Therefore, excluding observations that have been labeled as outliers from the mining view or scoring view should be done after careful examination.

There are three main approaches to identifying outliers.

- The first and simplest one is the specification of an *acceptable* range of values for each variable. The upper and lower limits of this range are determined using some simple statistical justification. Common implementations include specifying $\pm 3$ standard deviations from the mean value for continuous variables. In the case

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

of nominal and ordinal variables, a minimum percentage of the frequency count of the category of 1% is commonly used. The SAS implementation of these methods is straighforward and will be presented shortly. These methods have the advantage of being simple to implement and interpret. However, they do not take into account the multivariate nature of the data; that is, they rely on the examination of each variable independently. Thus the interaction between the variables is ignored. Furthermore, applying some transformations on continuous variables could reshape the distribution, and observations that were deemed outliers could become normal again. Therefore, this technique should be used only after attempts to reshape the variable distribution and after considering the interaction among variables.

- The second category of methods relies on using a dependent variable and a set of assumed independent variables to fit a candidate model such as a least-squares or regression. Observations that show large deviation from the fitted model are then deemed outliers. These methods remedy some of the disadvantages of the previous range-based methods—namely, taking into account the multivariate nature of the data. However, they are based on assuming a certain form of the model that generated the data. In typical data mining applications, the analyst experiments with different modeling techniques and model forms. Therefore, using one of these models to reject some observations because they do not fit well into that model could be misleading. For example, when one uses linear regression to find the outliers, and then uses a decision tree to develop the mining model, the removed outliers may have been useful in identifying a small node of importance in the decision tree.

- The third category of methods uses clustering algorithms to group the data into smaller subsets (clusters). Clusters containing a very small number of observations, ideally one observation, are identified as outliers. This is perhaps the best method that combines the best of the two methods. This is justified as follows. Clustering algorithms group *similar* observations in the same cluster based on the *distance* between them. This is similar to the idea of specifying an acceptable range, but in this case the range is limited to within the cluster or to a smaller subset of the data.

  The second feature is that clusters with very few observations are deemed outliers. This idea is conceptually equivalent to rejecting the observations that do not fit the model. The model here is defined as the set of clusters with a relatively large number of observations, that is, normal values. Finally, because the clusters are defined on the basis of *all* the variables, the issue of the multivariate nature of the data has been taken care of, without assuming a specific form or a structure of the model to be used.

Most of the methods used to test outliers assume that we are dealing with continuous variables. However, most data in data mining applications contains a large number of nominal and ordinal variables. Unfortunately, there are no equivalent methods for the identification of outliers for nominal or ordinal variables, other

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-11-08 15:22:15.

than simply specifying a minimum acceptable percentage of the frequency of each category. This is easily implemented using PROC FREQ and labeling the categories with low frequency as outliers.

On the other hand, accounting for the effect of the multivariate nature of the interaction between continuous and nominal variables, while using the methods used for continuous variables, is achieved by mapping nominal variables into indicator, or dummy, variables. The details of mapping methods are provided in Section 9.4.

The following subsections provide the SAS implementation of the three approaches just described.

## 7.5.1 Identification of Outliers Using Ranges

When a continuous variable follows a normal distribution, 99.7% of the observations fall within ±3 standard deviations from the mean. Based on this observation, we may attempt to identify the outliers by looking for the observations that fall outside this range. This is implemented in the following macro. The macro uses PROC UNIVARIATE to calculate the mean and the standard deviation and simply find the observation numbers that are outside this range.

Table 7.1 Parameters of macro `Extremes1()`.

| ***Header*** | `Extremes1(DSin, VarX, NSigmas, DSout);` |
|---|---|
| *Parameter* | *Description* |
| `DSin` | Input dataset |
| `VarX` | Continuous variable with outliers |
| `NSigmas` | Number of standard deviations used on each side of the mean to identify the outliers |
| `DSout` | Output dataset with observation numbers of the outliers and their values |

***Step 1***

We begin by extracting the variable to a temporary dataset to speed up the manipulation. We also generate a variable to hold the observation number. Obviously, the variable itself is allowed to be anything other than ObsNo.

```
data temp;
 set &DSin;
 ObsNo=_N_;
 keep &VarX ObsNo;
run;
```

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-11-08 15:22:15.

*Step 2*

We use PROC UNIVARIATE to calculate the mean and standard deviation of the variable. Then we extract these values to macro variables and calculate the upper and lower limits of the "normal" range. Values of the variable XVAR outside this range will be labeled as outliers.

```
/* calculate the mean and STD using proc univariate */
 proc univariate data=temp noprint;
 var &VarX;
 output out=temp_u   STD=VSTD   Mean=VMean;
run;

/* Extract the upper and lower limits
   into macro variables */
data _null_;
 set temp_u;
 call symput('STD', VSTD);
 call symput('Mean', VMean);
run;

%let ULimit=%sysevalf(&Mean + &NSigmas * &STD);
%let LLimit=%sysevalf(&Mean - &NSigmas * &STD);
```

*Step 3*

We generate the output dataset DSOUT by reading only the observations outside the normal range.

```
/* extract exteme observations outside these limits */
data &DSout;
 set temp;
 if &VarX < &Llimit or &VarX > &ULimit;
run;
```

*Step 4*

Finally, we clean the workspace and finish the macro.

```
/* clean workspace and finish the macro */
proc datasets library=work nodetails;
 delete temp temp_u;
quit;
%mend;
```

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

To illustrate the use of this macro, we use it with the dataset `Extremes` of Section 7.3, as follows:

```
%let DSin=extremes;
%let varX=x;
%let NSigmas=3;
%let DSout=ExtObs;

%Extremes1(&DSin, &VarX, &NSigmas, &DSout);
```

Executing the macro, using three standard deviations on each side of the mean, will result in an output dataset `ExtObs` with two observations for the values of 33 and 34. This is disappointing because we expected this approach to lead to the discovery of the outliers, which we inserted in this simple dataset.

The reason for the failure is that the mean and standard deviations calculated by `PROC UNIVARIATE` were calculated using *all* the nonmissing observations; that is, it included the outliers. Therefore, the mean as well as the standard deviation are biased because of these outliers. A simple remedy for this situation is to use *unbiased* measures for central tendency and dispersion for the variable in question.

It can be shown that the *median* is more robust to outliers than the mean. We can also use the *interquartile range* in place of the standard deviation as a measure for dispersion. The interquartile range is the difference between the values `Q3` and `Q1` of the variable, that is, the variable range that contains 50% of the values.

It is easy to modify macro `Extremes1()` to use the median and the interquartile range, as shown in the following macro, `Extremes2()`. The macro is identical to `Extremes1()` with the exception of the method of calculating the outliers range.

```
%macro Extremes2(DSin, VarX, NQRange, DSout);
/* Calculation of extreme values for a continuous variable
    outside the range of NQrange * QRange
    from the median. We use the median in place of the mean
    as a more robust estimate of the central tendency */

/* First, extract XVar to a temp dataset, and the
   observation number of the original dataset */
data temp;
 set &DSin;
 ObsNo=_N_;
 keep &VarX ObsNo;
run;

/* Calculate the median and QRange using proc univariate */
 proc univariate data=temp noprint;
 var &VarX;
 output out=temp_u   QRANGE=VQr   mode=Vmode;
run;
```

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-11-08 15:22:15.

```
/* Extract the upper and lower limits into macro variables */
data _null_;
 set temp_u;
 call symput('QR', VQr);
 call symput('Mode', Vmode);
run;
%let ULimit=%sysevalf(&Mode + &NQrange * &QR);
%let LLimit=%sysevalf(&Mode - &NQRange * &QR);

/* Extract extreme observations outside these limits */
data &DSout;
 set temp;
 if &VarX < &Llimit or &VarX > &ULimit;
run;

/* Clean workspace and finish the macro */
proc datasets library=work nodetails;
delete temp temp_u;
quit;
%mend;
```

Macro `Extremes2()` can be invoked on the same dataset and print the resulting dataset, as follows:

```
%let DSin=extremes;
%let varX=x;
%let NQRange=3;
%let DSout=ExtObs;

%Extremes2(&DSin, &VarX, &NQRange, &DSout);

proc print data=extObs;
run;
```

Macro `Extremes2()` correctly identifies the outliers in this case as the four inserted values: 22, 33, 23, and 34.

The number of standard deviations to use in `Extremes1()`, or the interquartile ranges in `Extremes2()`, is subjective and is determined by experience with the particular data of the problem. Commonly, the value of 3 is taken in both cases as a starting point. We reiterate that graphical examination of the variable distribution (e.g., using a histogram) after and before the identification of the outliers will reveal the success or failure of using this method.

### 7.5.2 Identification of Outliers Using Model Fitting

`PROC ROBUSTREG` of SAS/STAT performs what is known as *robust regression.* Robust regression is a modified form of the usual least-squares linear regression. It relies on

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-11-08 15:22:15.

alternative forms of the model fitting criterion such that the effect of outliers is minimized. There are three possible situations in which outliers could be present in the data pertaining to a linear model:

- The outliers could be present in the dependent (response) variable only.
- The outliers could be present in the independent variables only. In this case, observations containing the outliers are called *leverage points.*
- The outliers could be in both the response and independent variables. In this case, they are called simply outliers.

Outliers in linear models lead to large residuals. The residuals are defined as the difference between the predicted and actual values. Therefore, the modification of the objective function to be minimized during the model fitting is the key to robust regression. Different schemes of modifying the objective function and the associated solution algorithm lead to the development of several robust regression formulations. PROC ROBUSTREG implements four of these models. They are known as (1) M estimation, (2) LTS estimation, (3) S estimation, and (4) MM estimation. The default method is M estimation. In addition to fitting the model, while minimizing the effect of the outliers, these methods also offer the identification of outliers after building the model by the calculation of some distance measure between the model and each point.

Without getting into the mathematical details of these methods, the MM estimation method offers a good compromise between statistical efficiency and the ability to identify outliers. When the percentage of the outliers in the dataset is small (i.e., less than 5%), all methods, more or less, give close results using the default parameters.

Since our attention is focused on the identification of outliers, and not on building the best linear regression model, we will use the minimum capabilities of PROC ROBUSTREG to *point* to candidate outliers. We should not forget that linear regression is only one of the techniques usually used by data mining models. Finally, the identification of a set of points as *candidate* outliers should always be confirmed by more than one method before taking a drastic action such as removing them from the mining or scoring view.

The general syntax of PROC ROBUSTREG is very similar to PROC REG, which is used for the usual linear regression. The following macro wraps the procedure to generate an output dataset with two variables indicating the location of outliers and leverage points.

```
%macro RobRegOL(DSin, DV, IVList, DSout);
 /* Extraction of "Candidate" outliers using robust regression */
 proc robustreg data=&DSin method=mm (inith=502 k0=1.8); ;
   model &DV = &IVList /leverage;
   output out=&DSout outlier=_Outlier_ Leverage=_Leverage_;
 run;
 %mend;
```

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Table 7.2 Parameters of macro `RobRegOL()`.

| *Header* | `RobRegOL(DSin, DV, IVList, DSout);` |
|---|---|
| *Parameter* | *Description* |
| `DSin` | Input dataset |
| `DV` | Dependent variable |
| `IVList` | List of independent variables |
| `DSout` | Output dataset with outliers identified |

To demonstrate the use of the macro `RobRegOL()`, we generate an artificial dataset with 5% outliers, together with a linear model.

```
data TestOL ;
 do ObsNo=1 to 1000;
   /* generate three random IVs and an error term */
   x1=rannor(0); x2=rannor(0);
   x3=rannor(0); e=rannor(0);
/* In the first 50 observations, bias the x1, x3 variables */
   if ObsNo <= 50 then do;
    x1=100 * rannor(0);
    x3=100 * rannor(0);
                     end;
/* and in the last 5% of the dataset, bias the DV */
  if ObsNo > 950 then y=100 + e;
/* otherwise, the use the model equation */
  else y= 5 + 6*x1 + 7*x2 + + 8*x3 +  .4 * e;
 output;
 end;
run;
```

This code will generate 50 leverage points at the beginning of the dataset and another 50 outliers at the end. We invoke macro `RobRegOL()` as follows:

```
%let DSin=TestOL;
%let DV=y;
%let IVList=x1 x2 x3;
%let DSout=Test_Out;
%RobRegOL(&DSin, &DV, &IVList, &DSout);
```

The dataset `Test_Out` contains the two new variables: `_Outliers_` and `_Leverage_`, which indicate the location of *candidate* outliers and leverage points, respectively.

You have probably noticed by now that I insist on using the adjective *candidate* each time I refer to the results. This is to help you remember that there is no bullet-proof guarantee that these methods will discover the genuine outliers.

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

### 7.5.3 Identification of Outliers Using Clustering

Section 3.5 introduced briefly the main methods of cluster analysis. One of these methods is *k*-clustering, with one of its most common algorithms: *K*-means. *K*-means clustering is sensitive to outliers. In the presence of outliers that are very far from other observations, it often results in a cluster with one observation in it. From the point of view of clustering, this is considered a weakness. However, it is considered a blessing for finding outliers. The *K*-means algorithm is implemented in `PROC FASTCLUS` using a variety of distance measures. It is suitable for clustering large datasets and identifying outliers. This is usually achieved by specifying a large number of clusters. Clusters that have very few or, ideally, one observation are considered outliers.

In our implementation macro, we use `PROC FASTCLUS` to create 50 clusters, find the clusters with low frequency, and flag them as outliers. The number of clusters is controlled using the parameter `MAXC` in the `PROC FASTCLUS` statement. The value 50 is usually suitable for most datasets with more than few thousand records. `PROC FASTCLUS` is very fast, even with large datasets. Perhaps the only performance issue with this macro implementation is that we sort the observations of the dataset once to match the cluster indexes.

Table 7.3 Parameters of macro `ClustOL()`.

| ***Header*** | `ClustOL(DSin, VarList, Pmin, DSout);` |
|---|---|
| *Parameter* | *Description* |
| `DSin` | Input dataset |
| `VarList` | List of variables used in clustering |
| `Pmin` | Size of cluster, as percentage of dataset, to label its observations as outliers |
| `DSout` | Output dataset with outliers identified |

***Step 1***

Invoke `PROC FASTCLUS` to group the observations into 50 clusters. We then use `PROC FREQ` to identify the frequency of each cluster and store the results in the dataset `Temp_freqs`.

```
/* Build a cluster model with a default of
   50 clusters to identify outliers */

proc fastclus data=&DSin MaxC=50 maxiter=100
     cluster=_ClusterIndex_   out=Temp_clust noprint;
  var &VarList;
run;
```

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

```
/* Analyze temp_clust  */
proc freq data=temp_clust noprint;
  tables _ClusterIndex_ / out=temp_freqs;
run;
```

*Step 2*

Isolate the clusters with a size less than `Pmin` percent of the dataset size.

```
data temp_low;
 set temp_freqs;
 if PERCENT <= &Pmin;
 _Outlier_=1;
 keep _ClusterIndex_ _Outlier_;
run;
```

*Step 3*

Match-merge clusters with low numbers of observations with the clustering results using the cluster index variable `_ClusterIndex_`, which requires some sorting. Also change the outliers indicator from missing to zero where applicable.

```
proc sort data=temp_clust;
 by _ClusterIndex_;
run;
proc sort data=temp_low;
 by _ClusterIndex_;
run;

data &DSout;
 merge temp_clust temp_Low;
 by _ClusterIndex_;
 drop _ClusterIndex_ DISTANCE;
 if _outlier_ = . then _Outlier_=0;
run;
```

*Step 4*

Finally, clean up the workspace and finish the macro.

```
proc datasets library=work;
 delete temp_clust temp_freqs temp_low;
quit;
%mend;
```

Note that `PROC FASTCLUS` can output a dataset with the summary of the number of observations in each cluster directly through the option `MEAN`. However, we have elected to find the frequency in each cluster explicitly using `PROC FREQ` to make the implmentation more clear.

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

### 7.5.4 Notes on Outliers

As mentioned, the definition of outliers is subjective. Therefore, when we label some observations in the mining or scoring view as outliers, it does not mean that they should be deleted. In many cases, it might mean only that they follow a different distribution that might have a valid interpretation. Very often these outliers surface in a decision tree model as a persistent node with a small number of records. A persistent node in a decision tree is a node that keeps appearing even when the order of the splits is overridden manually.

On the other hand, when the number of observations labeled as outliers represents a large percentage of the data (i.e., more than 5%), this is a definite sign that there is more than one mechanism governing the data at hand. In this case, it is worth investigating developing a separate model for the outliers.

Finally, sometimes outliers are *the* interesting observation. This is particularly the case when the *interesting* event is rare in the population. Typical applications are fraud detection, searches for criminal activities, and investigating money laundering.

## 7.6 Testing Normality

Many statistical tests are based on assuming that the variable in question is normally distributed. `PROC UNIVARIATE` provides the capability to test the distribution of continuous variables against a variety of statistical distributions, including the normal distribution. The normality test is invoked by adding the option `NORMAL` in the `PROC UNIVARIATE` statement. For example, using the dataset `TestOL` (Section 7.5.2), we calculate the normality test statistics as follows.

```
proc univariate data=TestOL normal ;
 var x1;
run;
```

The following is the test of normality results portion of the output.

```
              Tests for Normality

Test                  --Statistic---  -----p Value------

Shapiro-Wilk          W     0.238644  Pr < W     <0.0001
Kolmogorov-Smirnov    D     0.425662  Pr > D     <0.0100
Cramer-von Mises      W-Sq  62.67828  Pr > W-Sq  <0.0050
Anderson-Darling      A-Sq  293.5579  Pr > A-Sq  <0.0050
```

The output provides four measures of normality. The most useful are the Shapiro-Wilk $W$ and the Kolmogorov-Smirnov $D$. The Shapiro-Wilk $W$ is calculated only when the number of observations is less than 2000. It takes values between 0 and 1,

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

with large values indicating a close fit to the normal distribution. The Kolmogorov-Smirnov *D* also varies between 0 and 1, but with smaller values indicating a close fit to normality. The extraction of the values of the normality statistics is straightforward using an `OUTPUT` statement in `PROC UNIVARIATE`.

In addition to the normality tests, `PROC UNIVARIATE` provides many graphical options to plot the variable distribution and the normality plots. We refer you to the online SAS help to learn more about these features.

## 7.7 Cross-tabulation

Cross-tabulation is concerned with the construction of the tables of frequency counts of two or more variables. `PROC TABULATE` offers many options and methods of calculating and presenting these tables. It is one of the most versatile procedures of the BASE/SAS product. `PROC FREQ` also offers the calculation of frequency counts for one or more variables. A detailed exploration of the implementation of these two procedures in data exploration is beyond the scope and focus of this book. Refer to the SAS documentation and many of the good books available for basic data reporting using SAS. Please check the SAS website for a comprehensive list of these references.

## 7.8 Investigating Data Structures

One of the objectives of data exploration that has a significant effect on the procedures of data preparation is the discovery of some structure within the data. Of particular interest are linear structures. Linear structures, when they exist, are typically investigated using Principal Component Analysis (PCA), and Factor Analysis (FA). Using these two techniques, we can achieve two objectives: (1) find possible interesting relationships between the variables and (2) reduce the number of variables.

In terms of data preparation procedures, the subject of this book, we focus more on the second objective. This is the subject of Chapters 15 and 16, respectively. PCA and FA as models that describe the structure of the data can be considered data mining models on their own. Refer to Johnson and Wichern (2001) for a detailed exposition of these two interesting subjects.

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-11-08 15:22:15.

This Page Intentionally Left Blank

Copyright © 2006. Elsevier Science & Technology. All rights reserved.

Refaat, Mamdouh. Data Preparation for Data Mining Using SAS, Elsevier Science & Technology, 2006. ProQuest Ebook Central,
http://ebookcentral.proquest.com/lib/tus/detail.action?docID=294574.
Created from tus on 2022-11-08 15:22:15.