138-2009

A Little Stats Won't Hurt You

Nathaniel Derby, Statis Pro Data Analytics, Seattle, WA

ABSTRACT

This paper gives an introduction to some basic but critically important concepts of statistics and data analysis for the SAS programmer who pulls or manipulates data, but who might not understand what goes into a proper data analysis. We first introduce some basic ideas of descriptive statistics for one-variable data, and then expand those ideas into many variables. We then introduce the idea of statistical significance, and then conclude with how all these ideas can be used to answer questions about the data. Examples and SAS® code are provided.

Keywords: Descriptive statistics, Statistical significance, SAS.

All SAS code used in this paper is downloadable from `http://nderby.org/docs/SGF138-2009.sas`.

INTRODUCTION: WHAT CAN STATISTICAL METHODS TELL US?

Statistical methods can be used to describe data and then extract information from them. They can be used to test hypotheses from the data ("Is X correlated with Y?").[1] They can be used to extrapolate trends for forecasts, or to look into the past and quantify what happened ("What is the effect of X on Y?). But before any of these more complex methods can be used, the first step is always to *look at the data* in many different ways. The idea is to effectively describe (or *summarize*) the data, and then look for "interesting" features. Note, however, that the precise meaning of that word depends on both the data and the business need in question; one person's interesting feature is another person's nuisance. For example,

- A *general business analyst* often wants to forecast some variable based on some kind of trend: "What can we expect the future values of the data to be?"
- A *risk analyst* cares a lot less about the expected future value of the variable and is much more interested in how much that variable fluctuates: "How volatile will the data be?"

Moreover, we want our data to make sense within our given context. A basic strategy is to look for *data irregularities*, which could be errors or something interesting. A common maxim in statistics is "if it looks interesting, it's probably wrong." However, if a data irregularity isn't wrong, it may be an interesting feature that might not have been found had a statistical method not been done. A good strategy is to think about what caused such an irregularity and to investigate it.

In this paper, we will illustrate these ideas with a few examples. These methods comprise *exploratory data analysis* (EDA), which involves looking at the data in a few different ways to let us see what the data are telling us, and which questions we should consider asking. Though many of these methods may seem simple, they are prerequisites for more advanced statistical methods we may have heard of, such as ANOVA (`PROC ANOVA`, `PROC GLM`), linear regression (`PROC REG`), logistic regression (`PROC LOGISTIC`), and ARIMA (`PROC ARIMA`). As such, *exploratory data analysis is essential*, and must be done before using more advanced statistical techniques.[2] These more complex statistical techniques all quantify the results we uncover with an EDA. For example, if an EDA shows us some evidence that X has some kind of an effect on Y, we can use a more complex model like linear regression (`PROC REG`) to give us our best estimate of that effect. If an EDA weren't performed first, we wouldn't know which two variables X and Y to look at.

For simplicity, this paper will focus on *univariate* methods, where we look at one variable at a time. We will compare two data sets by comparing their univariate characteristics, but we will not look at methods involving interactions between two variables (e.g., two-dimensional scatterplots).

[1] Note that correlation alone does not imply causality. As an example, cities with larger police forces tend to also have more crime. Does it follow that police presence causes crime?

[2] The major exception to this rule is for many techniques in data mining, for which complex statistical methods are applied to data without first looking at the data. This is done out of necessity; the data sets are huge, and there are simply too many variables to allow for doing an EDA.

EXPLORATORY DATA ANALYSIS (EDA)

The main idea of *exploratory data analysis* is, as its name implies, to explore the data. For the univariate case (looking at one variable at a time), this means looking at data in ways designed to easily discern the *distribution* of the data (i.e., how it is *distributed*, or spread out). This mainly involves two methods:

- *Data Visualization*: Graphical techniques which allow us to quickly and easily see general trends in the data.
- *Descriptive Statistics*: Statistical measures which summarize various characteristics of the data distribution.

Throughout this paper, we will illustrate examples of both classes of the above methods with a few data sets of annual percentage rates (APRs) of promissory notes, each having 50 observations. For example, this is our first data set:

6.99%	7.27%	6.77%	7.54%	7.50%	6.95%	7.55%	7.26%	7.03%	7.12%
6.85%	6.70%	7.11%	7.20%	7.12%	7.39%	7.09%	7.44%	7.32%	7.29%
7.37%	7.88%	7.15%	6.89%	6.70%	7.18%	7.69%	6.87%	6.98%	7.09%
7.41%	6.50%	7.93%	6.85%	7.22%	7.43%	7.05%	7.04%	7.32%	7.06%
7.20%	6.54%	7.06%	6.81%	7.54%	7.38%	6.95%	7.13%	7.24%	7.39%

These data sets will simply be numbered `data1` through `data6`, and will have the variables `apr` (`percent8.2` format, shown above) and `group` (`2.` format, equal to a number between 1 and 6).

Looking at these raw data points above makes it difficult to discern any characteristics about their distribution. Furthermore, the problem would be much worse if we had 500 or 5000 data points. There is simply too much information here. However, the two classes of methods above will alleviate this problem.

ONE-DIMENSIONAL SCATTERPLOTS

An easy and logical first step at looking at the data is to simple make a one-dimensional *scatterplot*, as shown in Figures 1(a)-(b).[3] We do this with the SAS/GRAPH® package as such:[4]

```
PROC GPPLOT data=data1;
  PLOT group*apr;
RUN;
```

The above code produces Figure 1(a), which looks very plain. We can make it look a little more readable with some customizations, producing Figure 1(b):

```
goptions ftitle='Times/bold' ftext='Times';                    * Use Times New Roman font;
symbol1 c=red;                                                 * Plot the data points in red;
axis1 label=( '' ) order=( .064 to .08 by .002 ) minor=( number=3 )
  value=( height=1.2 );                                        * Make the x-axis more readable;
axis2 label=( '' ) value=( height=1.2 );                       * Take off the label on the y-axis;
title 'Note Rates';                                            * Give this graph a title;

PROC GPLOT data=data1;
  PLOT group*apr=1 / haxis=axis1 vaxis=axis2;
RUN;
```

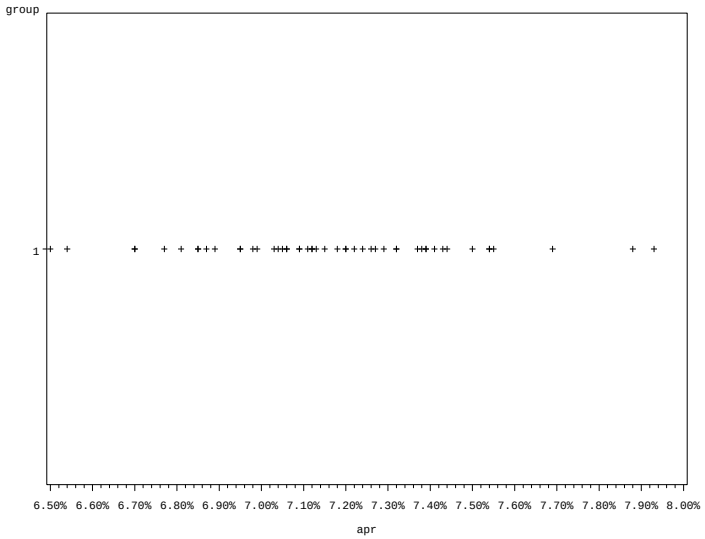
A scatterplot simply gives a first glance of the data, without giving any quantitative information. Here we see that most of the values are between 7.00% and 7.40%. Values below 6.70% or above 7.60% are rare, and as such might be called *outliers*. This term has a number of different mathematical definitions, but in this paper we will use this term loosely to simply mean data points that are outside the range that contains most of the data points.

Beyond finding outliers and a range (or number of different ranges) where most of the data lie, a scatterplot does not have many uses. Still, just for these two uses, a scatterplot is extremely useful, as it shows us information that we might not otherwise see, which could be important.

[3] Most uses of a scatterplot involve two dimensions, where two-dimensional data points (*x*,*y*) are plotted onto an *x*-*y* set of axes. Here we are essentially doing the same, except that we are plotting multiple values of *x* with one value of *y* (equal to 1 in this case).

[4] These results are from ODS PDF on SAS 9.1.3. Because of ODS restructuring on SAS 9.2, these results will look slightly different on that platform. As with any ODS PDF output, these examples can be generated by first using `options papersize="letter" orientation=landscape;` and then placing the example code between the statements `ods pdf file="&outputroot\outputx.pdf";` and `ods pdf close;`. Note that the code for this and other examples can be done without the SAS/GRAPH package simply by removing the `G` (e.g., `PROC PLOT` rather than `PROC GPLOT`).

(a)



(b)

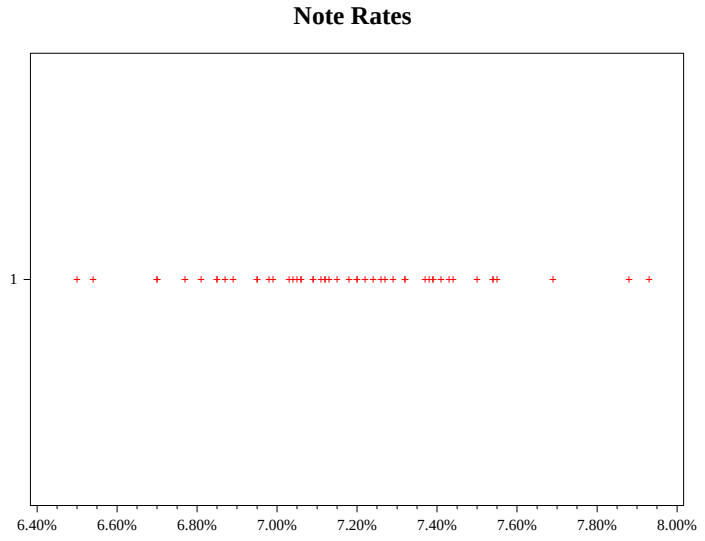


Figure 1: A one-dimensional scatterplot of 50 promissory bank note interest rates using SAS/GRAPH, with the default (a) and custom (b) formatting.

HISTOGRAMS

A *histogram* is simply a chart of frequency or percentage counts for different ranges of the data. We can do this via the following code, producing Figure 2(a):

```
PROC GCHART DATA=data1;
  VBAR apr;
RUN;
```

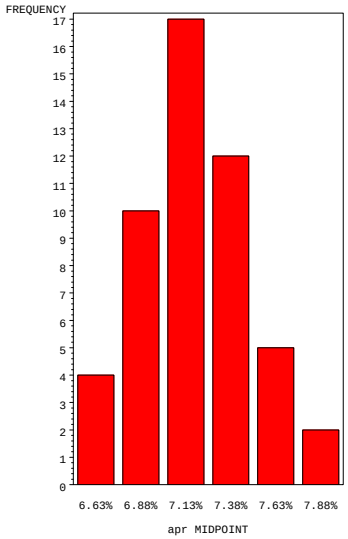
As before, we would like to add some custom formatting to make it look a little more readable, producing Figure 2(b):

```
goptions ftitle='Times/bold' ftext='Times';
axis1 label=( 'Interval Midpoint' height=1.2 ) offset=( 8, 8 ) value=( height=1.2 );
axis2 label=( angle=90 height=1.2 'Frequency' ) order=( 0 to 20 by 5 ) minor=( number=3 ) value=( height=1.2 );
title "Note Rates";

PROC GCHART DATA=data1;
  VBAR apr / maxis=axis1 raxis=axis2 width=4 space=2;
RUN;
```

Above, in the `axis1` statement, the `offset` option places spaces between the left-/rightmost frequency bars and the left/right edges of the graph. Here we see a chart of five bars whose heights are proportional to the frequency counts for their designated

(a)



(b)

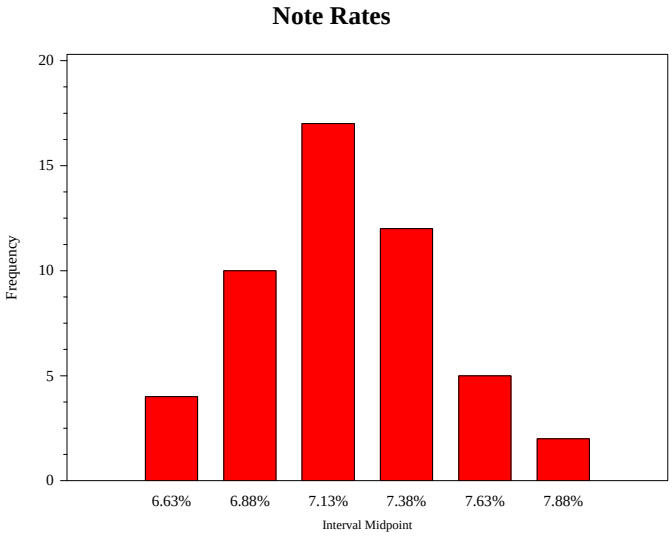


Figure 2: Histogram of 50 promissory bank note interest rates, with the default (a) and custom (b) formatting.

variable intervals. For instance, the first bar tells us that our data set has four data points with an APR in an interval centered at 6.63%. However, this is still not ideal, since we don't know explicitly what this interval is. Mathematically, the upper limit of this interval is halfway between 6.63% and the midpoint of the next interval, 6.88%, which is equal to 6.755%. For a quick glance at this data, this could be fine. However, it might be more useful to show these intervals explicitly. Furthermore, it might be nice to have the dividing line between the intervals to be a number more common than 6.755%. With a little more work with the `axis1` and `midpoints` statements, we can set the ranges ourselves, resulting in Figure 3:

```
axis1 label=( ' ' ) value=( height=1.2 '6.5% - 6.8%' '6.8% - 7.1%' '7.1% - 7.4%' '7.4% - 7.7%'
  '7.7% - 8.0%' ) offset=( 8, 8 );
axis2 label=( angle=90 height=1.2 'Frequency' ) order=( 0 to 20 by 5 ) minor=( number=3 ) value=( height=1.2 );

PROC GCHART DATA=data1;
  VBAR apr / maxis=axis1 raxis=axis2 width=5 space=3 midpoints = 0.0665 to 0.0785 by 0.003;
RUN;
```

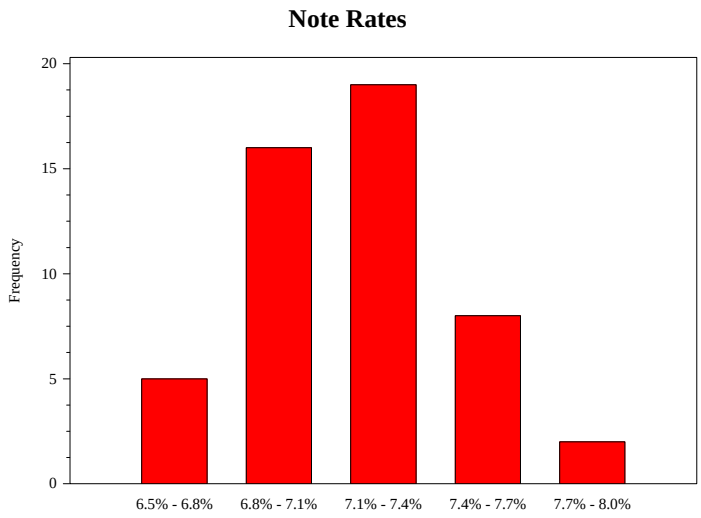


Figure 3: Histogram of 50 promissory bank note interest rates, with customized ranges.

In the above code, we explicitly set the midpoints of our intervals to be

6.65% 6.95% 7.25% 7.55% 7.85%

with the labels

6.5% - 6.8% 6.8% - 7.1% 7.1% - 7.4% 7.4% - 7.7% 7.7% - 8.0%

This makes our output just a little more readable. Note that we have gone from six intervals to five. This is done so that we will have one central interval and the same number of intervals on either side.

The reasoning for this will become clear later.

Looking at Figure 3, we see that the data appear to be centered at around 7.2%, and that the "bulk" of the data appears to be within 0.4% of 7.2%. That is, we are answering two questions of interest about the data:

- *What is the central value of the data?* This is typically the most important question about any data set, which might follow our intuition. If we purchase an investment, we usually want to know the average (expected) return on that investment. If we are manufacturing a device, we usually want to know the average failure rate. While we will quantify the term "average" shortly, the idea is that we would like to know the central value.
- *How spread out are the data?* This is typically the second most important question about a given data set. Indeed, having a central value is almost meaningless if we don't also know how spread out the data are. Indeed, given that the bank notes in our data set are centered around 7.2%, it makes quite a difference if the "bulk" of the data is within 0.4% of 7.2% versus within 4% of 7.2%.

Once again, by using an odd number of intervals in Figure 3, it is easier (at least in this case) to see an estimate of the central value and spread. We can quantify both of these concepts:

- *What is the central value of the data?* We can address this in three ways:
  - The *mean* of a data set is the arithmetic average, which means that we take their sum and divide it by the number of data points.
  - The *median* of a data set is the "middle value", meaning that 50% of the data is below this value. Arithmetically, this means that we order the data points. If we have an odd number of data points, we take the observation at the middle number (e.g., 3 is the middle number of 5). If we have an even number of data points, the median is the arithmetic average of the $\frac{n}{2}^{\text{th}}$ and the $\frac{n}{2} + 1^{\text{th}}$ observation.

    It is often useful to look at both the mean and the median. If they are equal, we say that the data is *symmetric*, meaning that there is just as much data above the mean as there is below it. Otherwise, we say that the data is *skewed*, or *long-tailed*, as shown in Figure 4. If the mean is less than the median, the "bulk" of the data is to the right, and there is a long tail on the left, and vice-versa. The direction of the longer tail is the same as the direction

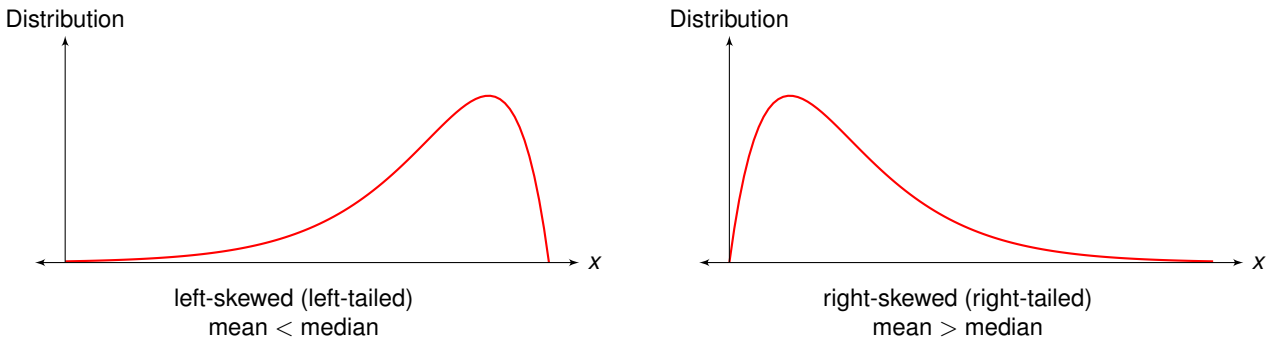


Figure 4: Skewness.

of the skewness. Therefore, a right-skewed distribution is sometimes informally called *right-tailed*. In our histogram in Figure 3, the data appears to be left-skewed, but this might be misleading, as we'll comment on shortly.

- The *mode* of a data set is the value or the category that is a maximum (or "high point") in the distribution or the histogram. In our histogram in Figure 3, the mode is the category "7-1% - 7.4%". Here we have only one mode, but more than one mode is possible. Practically, this means that there is only one value around which the data is clustered.

- *How spread out are the data?* We can address this in several ways:
  - The *standard deviation* gives a rough estimate of the average distance from the mean.[5] In a bell-shaped (or *normal*) distribution, about 65% of the data lie within one standard deviation of the mean.
  - The *minimum* is simply the smallest value in a given data set.
  - The *25th percentile* is a data point for which 25% of the data set lies below this value. This is also called the *1st quartile*.
  - The *75th percentile* is a data point for which 75% of the data set lies below this value. This is also called the *3rd quartile*.
  - The *interquartile range* is the difference between the 75th and the 25th percentiles.
  - The *maximum* is the largest value in a given data set.

  Note that the median is the same as the 50th percentile. Furthermore, as the standard deviation increases, the difference between the 75th and the 25th percentile also increases.

The shape of the distribution is often very important. For example, there is a huge difference between a right-skewed APR distribution (with more loans with smaller interest rates) than a left-skewed one (with more loans with larger interest rates).

While our histogram in Figure 3 appears to give us a useful estimate of the distribution of our data, care must be given to make sure that we don't have too many or too few intervals (also known as *levels* or *classes*). The number of intervals can be set with the `levels` option in `PROC GCHART` as follows:

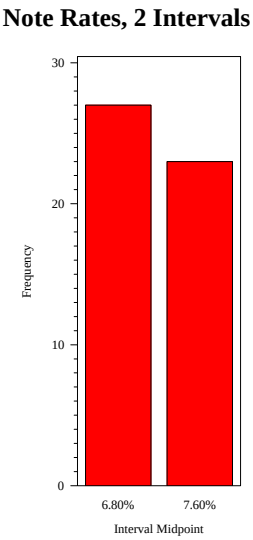
```
axis1 label=( 'Interval Midpoint' height=1.2 ) value=( height=1.2 );
axis2 label=( angle=90 'Frequency' height=1.2 ) value=( height=1.2 );
title 'Note Rates, xx levels';

PROC GCHART DATA=data1;
  VBAR apr / maxis=axis1 raxis=axis2 levels = xx;
RUN;
```

Doing this for 2 and 15 intervals gives us Figures 5(a) and (b), respectively. Here we see that with 2 intervals, we don't have enough intervals to show any useful information about the distribution. On the other hand, with 15 intervals, we get a misleading estimate of our distribution, since with so many intervals (in relation to the number of data points), we have many intervals with very small frequencies. Indeed, in such a situation, a difference of one or two data points will look significant. For example, in Figure 5(b), it looks like our data has four modes (high points): At the 1st, 7th, 10th and 15th intervals. Actually, the data has one mode; it just looks like there are more than one because of the paucity of data in many of these intervals.

[5] More specifically, it is the square root of the mean squared distance from the mean. It is calculated this way so that a negative distance is counted the same as a positive one. More details about this are given on pages 14 and 15.

(a)



(b)

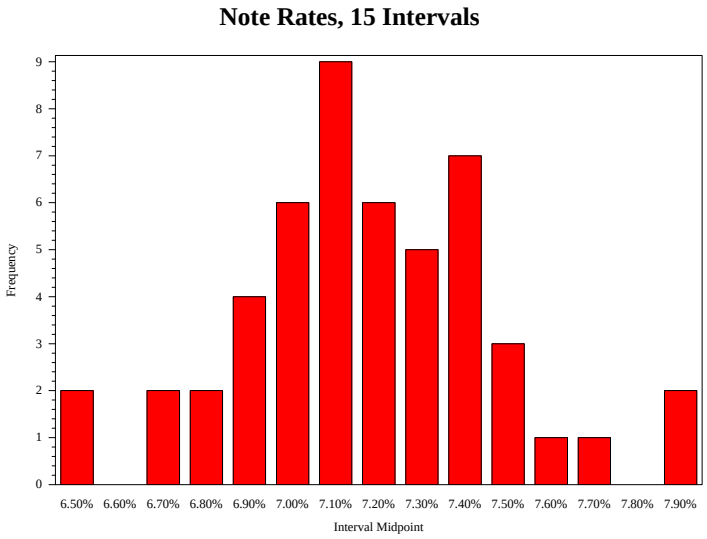


Figure 5: Histogram of 50 promissory bank note interest rates, with 2 intervals (a) and 15 intervals (b).

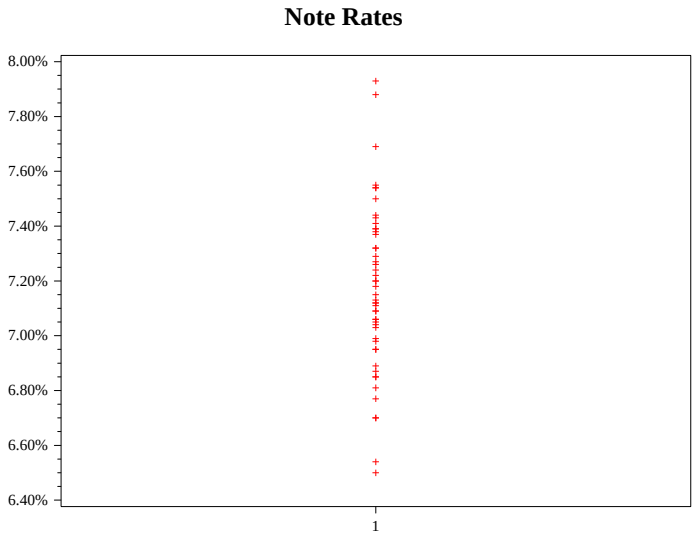
For the number of intervals, we can use the following guide as a rule of thumb:

Number of Data Points	Number of Intervals
Under 50	5 to 7
50 to 100	6 to 10
100 to 250	7 to 12
over 250	10 to 20

For more information about customizing histograms with SAS, see Watts (2008).

Overall, because of variations in resulting histograms because of a different number of intervals (or even just shifting the intervals, which is not illustrated here), inferences we get from histograms might be misleading. For more concrete results, we turn to the box plot.

(a)



(b)

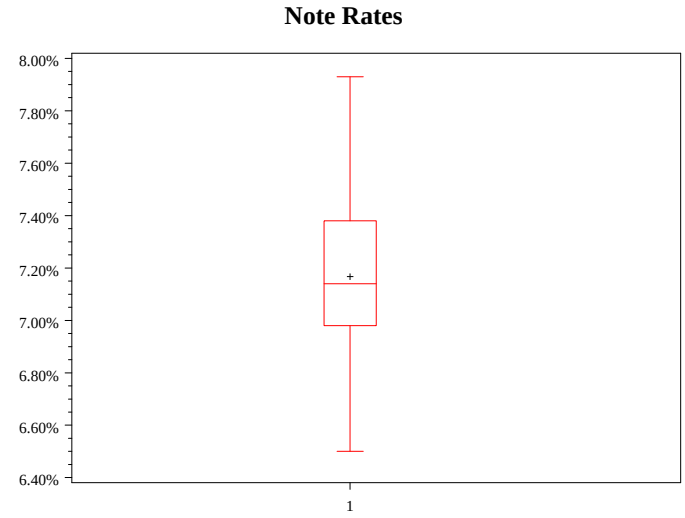


Figure 6: Vertical scatterplot (a) and box plot (b) of of 50 promissory bank note interest rates.

BOX PLOTS

For more precise descriptions of the data distribution, we can create a *box plot*.[6] This is simply a graphical representation of the six most common descriptive statistics of the data, as defined in the previous section:

minimum, 25^{th} percentile, mean (50^{th} percentile), 75^{th} percentile, maximum, mean.

The first five of these numbers, when used together, comprise a *five-number summary* of the data.

The SAS code needed to create a box plot is analogous to the `PROC GPLOT` statement for our scatterplot, but changing the order of the variables `apr` and `group`. This creates a one-dimensional scatterplot that is oriented vertically rather than horizontally, as shown in Figure 6(a):

```
goptions ftitle='Times/bold' ftext='Times';
symbol1 c=red;
axis1 label=( '' ) order=( .064 to .08 by .002 ) minor=( number=3 ) value=( height=1.2 );
axis2 label=( '' ) value=( height=1.2 );
title 'Note Rates';
```

[6] also known as a *box-and-whiskers plot*.

```
PROC GPPLOT data=data1;
  PLOT apr*group=1 / vaxis=axis1 haxis=axis2;
RUN;
```

While we can create a box plot using a minor variation of the above `PROC GPLOT` code, it is easier to use `PROC BOXPLOT` as follows, creating Figure 6(b):[7]

```
symbol1;

PROC BOXPLOT data=data1;
  PLOT apr*group / vaxis=axis1 haxis=axis2;
RUN;
```

In Figure 6(b), the highest and lowest horizonal lines represent the maximum and minimum values, respectively. The "box" has three horizontal lines in it: The highest and lowest ones represent the 75th and 25th percentiles, respectively, while the middle one represents the median. As such, the height of the box represents the interquartile range. The black cross represents the mean value.

Comparing Figures 6(a) and 6(b), we see two different representations of the same data set. They both show that the "bulk" of the data is between 7.00% and 7.40% (as we saw in the scatterplot section), except that with the box plot this is more precisely quantified. That is, in the scatterplot in Figure 6(a), it is difficult to see precisely where the 25th and 75th percentiles might be – whereas this is very clear in the box plot!

When looking at histograms of our data set, we had tentative evidence that the data were right-skewed (or right-tailed). This was tentative because, as we saw with the histograms of 2 and 15 levels, the shape of a histogram can be unduly influenced by the number of intervals, or by shifting the intervals. However, we know from Figure 4 that a right-skewed distribution has its mean greater than its median. With our box plot in Figure 6(b), we see that that is indeed the case, so we can now definitively conclude that our data set is right-skewed.

In summary,

A box plot is more reliable than a histogram in illustrating percentiles or skewness.

This is because a box plot incorporates *summary statistics*: Statistical measures (such as percentiles, or a sample mean) which describe the data. For illustrative purposes, it might be useful to write the five-number summary over the box plot with an annotate data set:[8]

```
PROC UNIVARIATE noprint data=data1;
  VAR apr;
  BY group;
  OUTPUT min=min mean=mean q1=q1 median=med q3=q3 max=max out=stats;
RUN;

DATA anno1;
  SET stats;
  FORMAT function $8. text $50.;
  RETAIN when 'a';
  function = 'label';
  text = '{'||trim( left( put( 100*min, 5.2 ) ) )||', '||trim( left( put( 100*q1, 5.2 ) ) )||', '||
    trim( left( put( 100*med, 5.2 ) ) )||', '||trim( left( put( 100*q3, 5.2 ) ) )||', '||
    trim( left( put( 100*max, 5.2 ) ) )||'}';
  position = '2';
  xsys = '2';
  ysys = '3';
  x = 1;
  y = 85;
  size = 1.1;
  OUTPUT;
RUN;

axis1 label=( '' ) order=( .064 to .082 by .002 ) minor=( number=3 ) value=( height=1.2 );

PROC BOXPLOT data=data1;
  PLOT apr*group / vaxis=axis1 haxis=axis2 annotate=anno1;
RUN;
```

[7] To make *almost* the same output as Figure 6(b) by using `PROC GPLOT`, use the code above used to create Figure 6(a), but change the `symbol1` statement to `symbol1 i=boxt co=red bwidth=10;`. This creates the desired boxplot, but without the black cross designating the mean value. This can be added to the graph via an annotate data set, but that seems overly complicated, given that `PROC BOXPLOT` does it without the annotate data set. This is one of a few differences between `PROC GPLOT` and `PROC BOXPLOT` when making box plots; see Adams (2008) for details.

Careful comparisons of Figures 6(a) and 6(b) show slight differences between their respective dimensions. This is simply due to various default setting differences between `PROC GPLOT` and `PROC BOXPLOT`.

SAS appears to be unable to make horizontal box plots (at least with `PROC GPLOT` or `PROC BOXPLOT`).

[8] An *annotate data set* is a data set with a special structure, used to add a symbol or data to an existing graph. For more information, see the SAS help files.

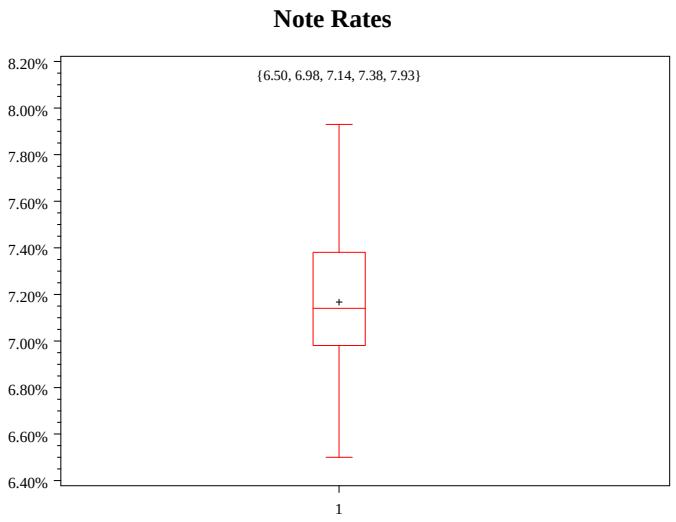


Figure 7: Box plot of 50 promissory bank note interest rates, annotated to include the five-number summary above.

Here we use `PROC UNIVARIATE` to compute these statistics and put them into a data set (`stats`), and then use that data set to create the annotate data set, which is then used in our box plot graph. We also modify the `axis1` statement to give us an interval longer than actually needed for the box plot, to make room for the five-number summary to be printed above it. The result is shown in Figure 7. Here we see that our five-number summary is

$$\{\ 6.50,\ 6.98,\ 7.14,\ 7.38,\ 7.93\ \}.$$

The percentage characters are left out simply to save space.

COMPARING MULTIPLE DATA SETS

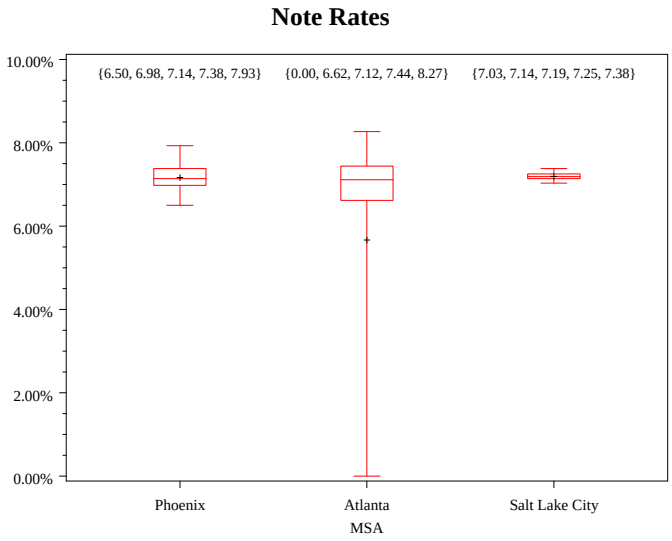
Scatterplots, histograms, box plots and summary statistics are effective for summary purposes, and they can be particularly effective when comparing multiple data sets. Suppose we have data sets of promissory banks notes from three different cities (Phoenix, Atlanta and Salt Lake City), where Phoenix is the data we've been analyzing up to now. We will compare these three data sets with box plots and the five-number summary:

```
DATA data123;
  SET data1 data2a data3;
RUN;

PROC UNIVARIATE data=data123 noprint;
  VAR apr;
  BY group;
  OUTPUT min=min mean=mean q1=q1 median=med q3=q3 max = max out=stats;
RUN;

DATA anno123;
  SET stats;
  FORMAT function $8. text $50.;
  RETAIN when 'a';
  function = 'label';
  text = '{'||trim( left( put( 100*min, 5.2 ) ) )||', '||trim( left( put( 100*q1, 5.2 ) ) )||', '||
    trim( left( put( 100*med, 5.2 ) ) )||', '||trim( left( put( 100*q3, 5.2 ) ) )||', '||
    trim( left( put( 100*max, 5.2 ) ) )||'}';
  position = '2';
  xsys = '2';
  ysys = '3';
  x = 1;
  y = 85;
  size = 1.1;
  OUTPUT;
RUN;
```

(a)



(b)

Figure 8: Box plots and five-number summaries of promissory bank note interest rates from three cities, with (a) and without (b) a data error for Atlanta.

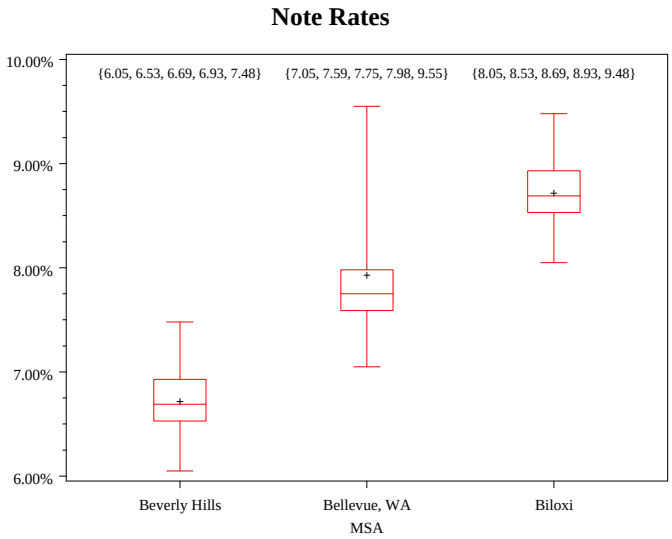
```
axis1 label=( '' ) minor=( number=3 );
axis2 label=( height=1.2 'MSA' ) order=( 1 to 3 by 1 ) value=( height=1.2 'Phoenix' 'Atlanta' 'Salt Lake City' )
  minor=none;

PROC BOXPLOT data=data123;
  PLOT apr*group / vaxis=axis1 haxis=axis2 annotate=anno123;
RUN;
```

The output, shown in Figure 8(a), shows an obvious data irregularity of some kind in Atlanta. Indeed, this is the real value of exploratory data analysis: To quickly and easily find data irregularities, just as we are seeing here. Certainly there is something to comment about with Salt Lake City as well, but Atlanta is the bigger problem. This analysis shows not only that the minimum is equal to zero, but that the data is heavily skewed toward that value as well. That is, the mean is not only less than the median (making it left-skewed), but it is *significantly* less than the median – so much that the mean is even less than the 25th percentile, which is highly unusual!

When finding a data irregularity such as this, a good first step is to ask if this makes sense. Without even looking at the raw data, we know from the minimal value in the five-number summary that we have at least one data point with an interest rate of 0%. Did something happen in Atlanta that would explain this? Assuming that interest rates are not tumbling in that city alone (which might not always be the case!), the answer is clearly no. Looking at the raw data reveals that we have 11 data points (out of 50) with an APR of 0%. For the purposes of this paper, we can assume that this is a data error which was corrected the

(a)



(b)

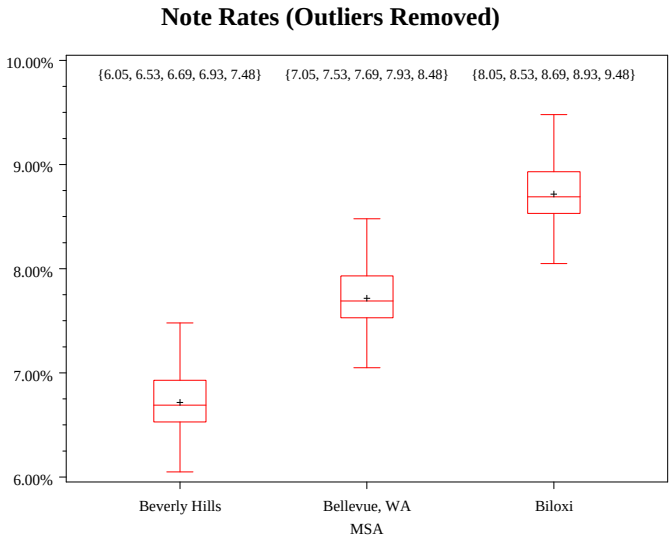


Figure 9: Box plots and five-number summaries of promissory bank note interest rates from three cities, with (a) and without (b) outliers for Bellevue, WA.

next day, thus producing Figure 8(b) when we re-run the code. We can make a couple observations from this revised box plot:

- All three cities appear to have about the same central values, at least compared with their spreads. That is, the median values (7.14%, 7.21% and 7.19%) and mean values (the locations of the black crosses) appear to be about the same.
- These three cities have very different spreads. Both the overall ranges (maximum minus the minimum) and interquartile ranges (75^{th} percentile minus the 25^{th} percentile) are highest for Atlanta and lowest for Salt Lake City.

Again, we can ask ourselves if this makes sense, or if it's a sign of some kind of data irregularity. For the purposes of this paper, we can assume that it does make sense; perhaps there is one monopoly bank in Salt Lake City keeping consistent rates, whereas there are many competitors in Atlanta with a rate war, with Phoenix in between these two extremes. Regardless of the plausibility of this explanation, these general principles of considering whether the data behavior makes sense still stand.

For another example, we now turn to Figure 9(a) to consider data from Beverly Hills, Bellevue (Washington), and Biloxi (Mississippi), using similar SAS code to that of the previous example. Here we see a data irregularity for Bellevue, except that it's oriented upward this time. Unlike for the other two cities (and the three cities in the preceding example), this data is extremely right-skewed. Does this make sense?

Let's assume for the purposes of this paper that this irregularity actually does make sense. The raw data reveals that there are six values (out of 50) above 9.4%. Perhaps in Bellevue there was a short period of extremely high interest rates because

of some local shortage. Although these are valid data points, these are *outliers* in the sense that they are a small number of values that are far outside the range for all the rest of the data points. As such, including them in the data analysis distorts the general tendencies we are looking for.

One general strategy with outliers is to remove them from the analysis, resulting in Figure 9(b).[9] We can make a couple observations from this:

- All three cities appear to have about the same spreads. That is, the overall ranges (maximum minus the minimum) and interquartile ranges (75^{th} percentile minus the 25^{th} percentile) are the same for the three cities.
- These three cities have very different central values, at least compared with their spreads. That is, both the mean and median for Beverly Hills are below the minimum for Bellevue, whose mean and median are below the minimum for Biloxi.

Again, the questions to ask is if these observations make sense. What could explain interest rates being very low for Beverly Hills and very high for Biloxi? That question is left for a business analyst and is outside the realm of this paper.

In this case, we removed the outliers from this data analysis. However, if we do this, **we must be careful to report them in the final analysis**. That is, we cannot simply discard them! In most cases, it is enough to mention it in a sentence or a footnote in the final report. But to either not report them or hide them (e.g., in end notes or the appendix), to say the least. Be sure to always mention something about outliers that have been removed from the analysis.

DATA SUMMARIES: DEMYSTIFYING `PROC UNIVARIATE`

Up to now, we have been deliberately ignoring `PROC UNIVARIATE`. There is a reason for this: **`PROC UNIVARIATE` has much, much more information than we would usually need** for any one data summary. As an illustration, let's look at Figure 10, which is the `PROC UNIVARIATE` output when applied to our first data set (as shown in Figure 1). The purpose of a data summary (like the five-number summary) is to describe a large set of data with just a few numbers. With this goal in mind, note that for this example,

`PROC UNIVARIATE` is using 46 numbers to summarize 50 data points.

As such, it would appear that `PROC UNIVARIATE` is not an effective data summary. Furthermore, this would be true even if we were analyzing a data set with 500 or 5000 values; indeed, a 46-number data summary is not very effective.

However, `PROC UNIVARIATE` was not meant to be a data summary *per se*. Rather, it was designed to give (nearly) all possible statistics for a data summary that a user would ever want. Keep in mind, however, that just because a statistic is listed on `PROC UNIVARIATE` does not mean that we need to use it! Indeed, some of the output such as kurtosis, standard error (of the) mean, or the signed rank test are rather esoteric measures which are of interest only to very few data analysts.

Nonetheless, for completeness, we will detail all parts of the output here, starting from the bottom:

- `Extreme Observations` are simply the five lowest and highest observations, and their observation numbers (i.e., where they are in the data set).
- `Quantiles` are just percentiles (the words are synonyms). Thus, the 75^{th} percentile is 0.0738, meaning that 75% of the rates are below the value of 0.783 (or 7.38%). However, there are some different ways to define a percentile value. The output `(Definition 5)` means that we define the percentiles just as we did the median on page 5. That is, we order the data points and calculate the percentile rank. For instance, for 50 data points, the 25^{th} percentile rank is $0.25 \times 50 = 12.5$.
  - If the percentile rank is a whole number j, then the percentile is the number in the j^{th} position when the values are in sequence. For instance, with 50 data points, 50% of 50 is 25, so the percentile is the 25^{th} number in the ordered sequence of all data points.
  - If the percentile rank is a fractional number $j + g$, where j is a whole number and g is a fractional number, then the percentile is the average of the numbers in the j^{th} and $(j + 1)^{st}$ positions when the values are in sequence. For instance, with 50 data points, 25% of 50 is 12.5, so the percentile is $0.5 \times (x_{12} + x_{13})$, where x_j is the j^{th} number in the ordered sequence of all data points.

  There are other definitions that also make sense. See the SAS help file "The `UNIVARIATE` Procedure: Calculating Percentiles" for more information. For general exploratory data analysis, however, the percentile definition is not of major importance.

[9] The more correct strategy is to designate the outliers as isolated data points over the maximum or under the minimum of the normal range (whichever is the case). `PROC BOXPLOT` has an option to do this. However, for simplicity that is left out of this paper.

Moments

N	50	Sum Weights	50
Mean	0.071668	Sum Observations	3.5834
Std Deviation	0.00305015	Variance	9.30344E-6
Skewness	0.18373955	Kurtosis	0.2700401
Uncorrected SS	0.25727098	Corrected SS	0.00045587
Coeff Variation	4.25595091	Std Error Mean	0.00043136

Basic Statistical Measures

Location		Variability	
Mean	0.071668	Std Deviation	0.00305
Median	0.071400	Variance	9.30344E-6
Mode	0.067000	Range	0.01430
		Interquartile Range	0.00400

NOTE: The mode displayed is the smallest of 10 modes with a count of 2.

Tests for Location: Mu0=0

Test	-Statistic-		-----p Value------	
Student's t	t	166.1454	Pr > |t|	<.0001
Sign	M	25	Pr >= |M|	<.0001
Signed Rank	S	637.5	Pr >= |S|	<.0001

Quantiles (Definition 5)

Quantile	Estimate
100% Max	0.0793
99%	0.0793
95%	0.0769
90%	0.0754
75% Q3	0.0738
50% Median	0.0714
25% Q1	0.0698
10%	0.0679
5%	0.0670
1%	0.0650
0% Min	0.0650

Extreme Observations

-----Lowest-----		-----Highest----	
Value	Obs	Value	Obs
0.0650	32	0.0754	45
0.0654	42	0.0755	7
0.0670	25	0.0769	27
0.0670	12	0.0788	22
0.0677	3	0.0793	33

Figure 10: `PROC UNIVARIATE` output for the APR values of the original data set.

From here, we then go to the top of the output:

- `N` is the total number of data points in our data set.
- `Sum Weights` gives the total sum of all our data point weights. This is only important when we weight some data points more than others; otherwise, it is always equal to the total number of data points, `N`.
- `Mean` is our sample mean, as we have seen before.
- `Sum Observations` is simply the sum of the variable observations in question over all the data points. This is sometimes used for further calculations (such as the mean, which is equal to $\frac{\texttt{Sum Observations}}{\texttt{N}}$) and can be used to quickly check some other calculations from `PROC UNIVARIATE`.

$$\text{Sum of Observations} = \sum_{i=1}^{N} x_i, \qquad x_i = i^{\text{th}} \text{ observed value.}$$

- `Std Deviation` is our sample *standard deviation*, as we briefly mentioned before. This gives us an estimate of our average deviation from the sample mean, and is the square root of the *sample variance*:

$$\text{Standard Deviation} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}, \qquad x_i = i^{\text{th}} \text{ observed value}, \quad \bar{x} = \text{sample mean.}$$

- `Variance` is our sample *variance*, which gives an estimate of our average squared distance from the mean:[10]

$$\text{Variance} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2, \qquad x_i = i^{\text{th}} \text{ observed value}, \quad \bar{x} = \text{sample mean}.$$

  The distance is squared so that we equally count positive and negative distances (i.e., whether a given value is above or below the mean).

- `Skewness` gives us a numerical estimate of how *skewed* our data is. As shown in Figure 4, a negative skew is a left skew, while a positive skew is a right skew. The absolute value gives the degree to which a sample distribution is skewed. A skewness of zero means the sample data is symmetric, so that the mean equals the median.

- `Kurtosis` gives us a numerical estimate of how "peaked" our data distribution is at its mode (if the distribution is unimodal). A higher value indicates a higher peak and thinner tails, while a lower value indicates a lower peak and fatter tails.

- `Uncorrected SS` gives us an *uncorrected sum of squares*, which is the quantity

$$\text{Uncorrected Sum of Squares} = \sum_{i=1}^{N} x_i^2, \qquad x_i = i^{\text{th}} \text{ observed value}.$$

  It is simply an intermediate calculation which may be of interest.

- `Corrected SS` gives us an *uncorrected sum of squares*, which is the quantity

$$\text{Corrected Sum of Squares} = \sum_{i=1}^{N}(x_i - \bar{x})^2, \qquad x_i = i^{\text{th}} \text{ observed value}, \quad \bar{x} = \text{sample mean}.$$

  Again, it is an intermediate calculation.

- `Coeff Variation` is the *coefficient of variation*, which is simply the ratio of 100 $\times$ the sample standard deviation to the standard mean:

$$\text{Coefficient of Variation} = \frac{100 \times \texttt{Std Deviation}}{\texttt{Mean}}.$$

  It is used as a scaled version of the spread, which is sometimes useful (i.e., a spread of 100 is large if the mean is 1, but very small if the mean is 10,000).

- `Std Error Mean` is the *standard error of the mean* and is equal to

$$\text{Standard Error of the Mean} = \frac{\texttt{Std Deviation}}{\sqrt{\texttt{N}}}.$$

  This is the estimated standard deviation of the distribution for the actual value of the mean (i.e., not the estimated value). This will be explained more below, in the discussion of hypothesis tests.

All the quantities under the heading `Basic Statistical Measures`, (`Mean`, `Median`, `Mode`, `Std Deviation`, `Variance`, `Range` and `Interquartile Range`) have been explained either above or in earlier sections. Note, however, the note in our output in Figure 10 that there were 10 modes, each with a count of two. This simply means that there were 10 values that each had two observations. If there were one value with three observations, that would be the new (unique) mode.

This leaves the section entitled `Tests for Location: Mu0=0`. This will be explained in the next section.

Lastly, be aware that SAS procedures like `PROC UNIVARIATE` often give more decimal places than are needed, which can sometimes give a misleading level of accuracy. Here, `PROC UNIVARIATE` states that the mean is 7.1668%, but that number comes from only 50 data points, so this isn't actually accurate to four decimal places. Stating that the mean is 7.17% usually suffices – there is no need to add confusion by incorporating many decimal places.

Overall, we see by inspection that all the other data in our `PROC UNIVARIATE` output in Figure 10 basically matches our results from the scatterplots, histograms and box plots, as they should.

[10] The sum of squared distances is divided by the quantity $N-1$ rather than by N to account for mathematically expected bias that would otherwise result.

HYPOTHESIS TESTS AND STATISTICAL SIGNIFICANCE

All the statistical techniques done up to now have computed *sample* quantities. For instance, the sample mean is the mean of our sample of 50 observations. The assumption is that our data set of 50 observations comprises a small sample of a (possibly infinite) number of observations that we are not observing. However, we assume that our sample is representative of the population of all possible observations (i.e., we don't have a biased sample), so that our inferences from our sample can be applied to the population.

Using our example, we assume that each data set of interest rates is a sample from interest rates of all loans in a given geographical area.

A *hypothesis test* is a test on our data whether a theoretical (unobservable) quantity (such as the mean of the underlying data) is significantly different from some other value. In Figure 10, our `PROC UNIVARIATE` tells us that our sample mean (i.e., the mean from our sample of 50 observations) is 7.1668%. But what we often *really* care about is the theoretical mean, not the sample one.

For many uses, a standard question to ask is: Is the (unobservable) theoretical value significantly different from zero?

By *significant*, we mean *statistically significant*, meaning that we account for the spread of the data. Generally, the larger the spread (i.e., the more volatile the data), the less reliable our estimates will be, including our estimate of the sample mean. Therefore, determining whether our sample data point is significantly different than zero takes into account both our sample estimates of the mean (is the sample mean far from zero?) and the standard deviation (is the spread small enough to count out zero?).

This is a hypothesis test, and this particular one is so common that sample output from it is included in `PROC UNIVARIATE`.

There are many, many kinds of hypothesis tests, but `PROC UNIVARIATE` lists outcomes from three of them. Each of them calculates a *test statistic* for our hypothesis that the mean value is equal to zero:

- `Student's t` is the *student's t-test*, or simply the *t-test*. It calculates the quantity

$$\text{Test statistic} = \frac{\bar{x}}{\text{SE}(\bar{x})} = \frac{\texttt{Mean}}{\texttt{Std Error Mean}}$$

  and matches it against the student's t distribution.

- `Sign` is the *sign test* to test whether the median is significantly different from zero. The test statistic is the average of number n^+ of values greater than zero and the number n^- of values less than zero:

$$\text{Test Statistic} = \frac{n^+ + n^-}{2}.$$

- `Signed Rank` is the *Wilcoxon signed rank test*, where the test statistic is a complex calculation derived from ranks of the values.

Which of these tests to use depends on which mathematical assumptions can be considered valid from the data. For more information on any of these tests or their underlying assumptions, see the SAS help file "The `UNIVARIATE` Procedure: Tests for Location," or Kanji (1999, pp. 17,78,80). For convenience, `PROC UNIVARIATE` lists all three of them, so that the user can glance at the results without first assessing the mathematical assumptions in question.

For each of these tests, the test statistic is matched against a certain theoretical distribution, and a nonzero *p-value* is computed. This value gives the probability that the estimated quantity is equal to zero. We can reject this hypothesis (and thus conclude that the theoretical mean is nonzero) if this *p*-value is smaller than a a given number (usually 0.05).

In the `PROC UNIVARIATE` output shown in Figure 10, the `Tests for Location: Mu0=0` section gives a table of test statistics and *p*-values for each of these three tests described above. For our data, the *p*-values are all very small (less than 0.0001), and thus indicate that our mean values are significantly different from zero (which is to be expected, since every single data value was positive).

CONCLUSIONS

This paper presents some of the most commonly used tools of exploratory data analysis. These methods give us data visualization and summarization techniques which are simple, yet very effective for quickly estimating the central tendency, spread, and other characteristics of the data distribution. These methods help us detect data irregularities which might be errors, outliers, or some interesting aspect of the data (depending on the context). Using these methods can also help us compare two or more different data sets and assess their differences. Lastly, these methods give us indicators of what should be further analyzed with more complex statistical methods.

For more information about any of the statistical ideas in this paper, good references are Gonick and Smith (1993) and Siegel and Morgan (1996). For more information about using statistical techniques in SAS, see UCLA (2009).

REFERENCES

Adams, R. (2008), Box plots in SAS: UNIVARIATE, BOXPLOT, or GPLOT?, *Proceedings of the Twenty-First Northeast SAS Users Group Conference*.
http://nesug.org/proceedings/nesug08/np/np16.pdf

Gonick, L. and Smith, W. (1993), *The Cartoon Guide to Statistics*, HarperCollins Publishers, New York.

Kanji, G. K. (1999), *100 Statistical Tests*, Sage Publications, London.

Siegel, A. F. and Morgan, C. J. (1996), *Statistics and Data Analysis: An Introduction*, second edn, John Wiley and Sons, Inc., New York.

UCLA (2009), Resources to help you learn and use SAS, Academic Technology Services: Statistical Consulting Group.
http://www.ats.ucla.edu/stat/sas/

Watts, P. (2008), Using SAS software to generate textbook style histograms, *Proceedings of the Twenty-First Northeast SAS Users Group Conference*.
http://nesug.org/proceedings/nesug08/np/np03.pdf

ACKNOWLEDGMENTS

I thank Lisa Eckler for giving me the idea for this paper and inviting me to present it at the 2009 SAS Global Forum. I furthermore thank my former professors and employers for making me realize that basic statistical ideas are far from trivial. Lastly, and most importantly, I thank Charles for his patience and support.

CONTACT INFORMATION

Comments and questions are valued and encouraged. Contact the author:

Nathaniel Derby
Statis Pro Data Analytics
815 First Ave., Suite 287
Seattle, WA 98104-1404
206-973-2403
nderby@sprodata.com
http://nderby.org